Received 01 June 2018; | Revised 01 August 2018; | Accepted 01 August 2018

DOI: xxx/xxxx

ARTICLE TYPE

Open-source platforms for fast room acoustic simulations in complex structures

Matthieu Aussal[1] | Robin Gueguen[2]

[1]Centre de mathématique appliquées, École Polytechnique, 91128 Palaiseau, France
[2]Institut des Sciences du Calcul et des Données, Sorbonne Université, Campus Pierre et Marie Curie - 4 place Jussieu, 75252 Paris Cedex 05 France

Correspondence
Email: matthieu.aussal@polytechnique.edu
Email: gueguen.robin@gmail.com

Summary

This article presents new numerical simulation tools, both for Matlab and Blender CAD software. Available in open-source under GPL 3.0 license, it uses a ray tracing / image-sources hybrid method to calculate room acoustics for large meshes. Performances are optimized to solve significant size numerical problems (typically more than 100,000 surface elements and about a million of rays). For this purpose, a *Divide and Conquer* approach with a recursive octree structure has been implemented to reduce the quadratic complexity of the ray/element interactions to near-linear. Thus, execution times are less sensitive to mesh density, which allows complex geometry simulations. After ray propagation, a hybrid method leads to image-sources format, which can be visually analyzed to localize sound map. Finally, impulse responses are constructed from the image-sources and FIR filters are proposed natively over 8 octave bands, taking into account material absorption properties and propagation medium. This algorithm is validated by various comparison with theoretical test cases. Furthermore, an exemple on a complex case with the ancient theater of Orange is presented.

KEYWORDS:
room acoustic, ray-tracing, image-sources, octree, Matlab, Blender, archeology, open-source

1 | INTRODUCTION

Today, digital technologies allow research to explore previously inaccessible areas, as virtual reality for archaeology. In this domain, many works focus on the visual restitution, but acoustic studies can reinforce researches to improve the understanding of the ancient world. For example, during the Roman Empire, architects have designed buildings also based on the sound propagation they wanted to achieve[1]. In this study, we focus on the ancient theater of Orange which has a significant size (100m wide), a complex geometry (decoration, gradins, colonnes, arcades ...) and which is open air. In a previous work, a complete mesh was designed using Blender CAD software (citer Blender), according to the most recent archeological knowledge[2]. To be representative, this mesh has hundreds of thousands of elements (triangular faces), with complex shape (inserer figure maillage pour montrer la complexite). As this numerical model is now used by researchers to perform archeological hypothesis, we developed our own room acoustic software, directly integrated in the archeologists workflow. As the mesh size makes difficult the use of precise methods (FEM, BEM, ...), this software is based on a ray-tracing approximation, in order to compute fastly the full-band room impulse response (50 to 15000Hz). This tool has been developed following two steps. We first build a prototype using Gypsilab, an open source Matlab framework for fast prototyping (citer le papier gypsilab). This preliminary work was usefull to construct and validate ideas and algorithms. It conducts to the creation of a new toolbox, now added to the master

branch of Gypsilab, and freely downloadable (citer adresse github). In a second step, all algorithms was retranscrypted in C++ using Qt Creator, leading to an autonomous library. A python interface was added, in order to use this library as a Blender plug'in. At the end, this plug'in allow archeologists to only work on Blender, modifying easily meshes and materials, run acoustic simulation and visualize results.
After reminders on acoustical energy propagation represented by ray-tracing, this paper gives implementation details for fast computation for large meshes. At the end, validation test cases are given and application on our modelization of the ancient theater of Orange is performed.

2 | ACOUSTICAL ENERGY MODELIZATION

2.1 | Continuous domain equation

By modelling a point sound source as a localized pulse in space, the associated acoustical energy E(t) propagates[3] over time on a spherical surface $S(t)$, such that :

$$E(t) = E_0 \int_{S(t)} \vec{I}(t).\overrightarrow{ds} \qquad \forall t > 0, \tag{1}$$

with E_0 the initial energy and $\vec{I}(t)$ the acoustical intensity. According to the first principle of thermodynamics and by neglecting the effects of losses related to the absorption of the propagation medium, the acoustic energy is preserved over time. Thus, for a normalized punctual source :

$$\int_{S(t)} \vec{I}(t).\overrightarrow{ds} = 1 \qquad \forall t > 0. \tag{2}$$

After integration on the spherical surface $S(t)$, the infinitesimal acoustic intensity is written :

$$||\vec{I}(t)|| = \frac{1}{4\pi d(t)^2} \qquad \forall t > 0, \tag{3}$$

reflecting that intensity decreases as the square of the distance to the source d(t). Integrating on a portion $\sigma(t)$ of the complete sphere $S(t)$, the energy is then carried by the solid angle Ω_σ :

$$E_\sigma(t) = E_0 \int_{\sigma(t)} \frac{1}{4\pi d(t)^2} ds = \frac{E_0}{4\pi}\Omega_\sigma. \tag{4}$$

This last equation shows that the energy of a solid angle is constant over time and corresponds to a portion of the initial energy E_0. Thus, subdividing $S(t)$ in N portions $\sigma_i(t)$, the total energy can be decomposed as a sum of elementary energies, carried by solid angles Ω_i, such as :

$$E(t) = \sum_{i=1}^{N} E_i(t) = \frac{E_0}{4\pi} \sum_{i=1}^{N} \Omega_i \qquad \forall t > 0. \tag{5}$$

Ones can notice that $(\Omega_i)_{i\in[1,N]}$ is a directional basis, representing energy propagation by piecewise constant elements. Furthermore, for greater clarity, we definitively set in the following $E_0 = 1$.

2.2 | Discrete model

To numerically represent the energy propagation, we have to discretize basis $(\Omega_i)_{i\in[1,N]}$ in (eq. 5). For this purpose, we define a *ray* object composed by :

- An origin coordinate x_i,
- A direction vector $\overrightarrow{u_i}$,
- An energy value E_i.

For example, with an omnidirectional source, *rays* are given by :

- The source coordinate ($x_i = x_s,\ \forall i \in [1, N]$),

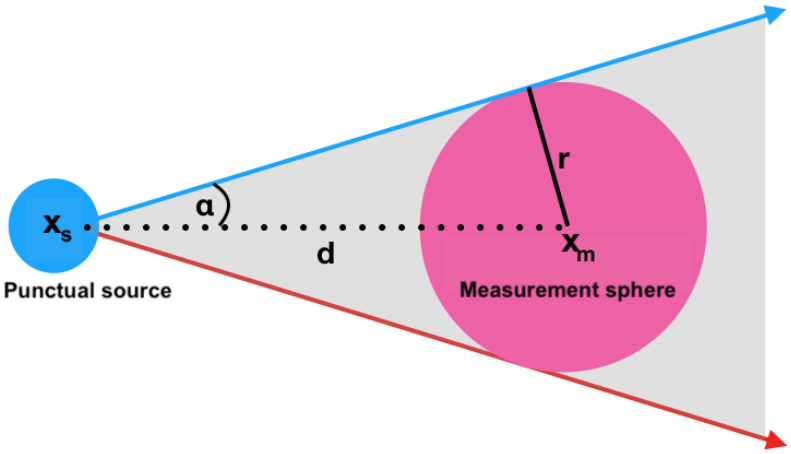


FIGURE 1 Representation of a r-radius measurement sphere centered in x_m, receiving energy from a sound source in x_s.

- A unit sphere uniform sampling (e.g. icosahedre subdivision, Fibonacci's rule[4], etc.),
- An uniform energy repartition ($E_i = \frac{4\pi}{N}$, $\forall i \in [1, N]$).

To complete this approach, we have to define a discrete measure of energy propagation. To this end, we consider a r-radius measurement sphere $S(x_m, r)$, centered on x_m. We can then add the contributions of the n rays that intersect this sphere to calculate the acoustic energy E_m at point x_m :

$$E_m \approx \frac{1}{4\pi} \sum_{i=1}^{n} E_i. \tag{6}$$

In the particular case of an omnidirectionnal source, we have :

$$E_m \approx \frac{n}{N}, \tag{7}$$

which means that the measured energy E_m is statistically represented by the ratio of the countered *rays* to the total number of *rays*.

Moreover, applying the continuous model (eq. 4), the measured energy is given by :

$$E_m = \frac{1}{4\pi} \Omega_m, \tag{8}$$

where Ω_m is a solid angle of a cone of revolution (see fig. 1), such as :

$$\Omega_m = 2\pi(1 - \cos\alpha). \tag{9}$$

Following the figure 1 :

$$\Omega_m = 2\pi \left(1 - \sqrt{1 - \frac{r^2}{d^2}}\right) \tag{10}$$

and considering $\frac{r}{d} \ll 1$,

$$\Omega_m = \pi \frac{r^2}{d^2}. \tag{11}$$

At the end,

$$E_m \approx \frac{n}{N} \approx \frac{\pi r^2}{4\pi d^2}. \tag{12}$$

To ensure the existence of this last approximation, at least one *ray* have to be measured ($n \geq 1$). Thus, fixing a measurement radius r, approximation (12) gives a maximum validity range of the discrete model :

$$d \leq \frac{r}{2} \sqrt{\frac{N}{n}}. \tag{13}$$

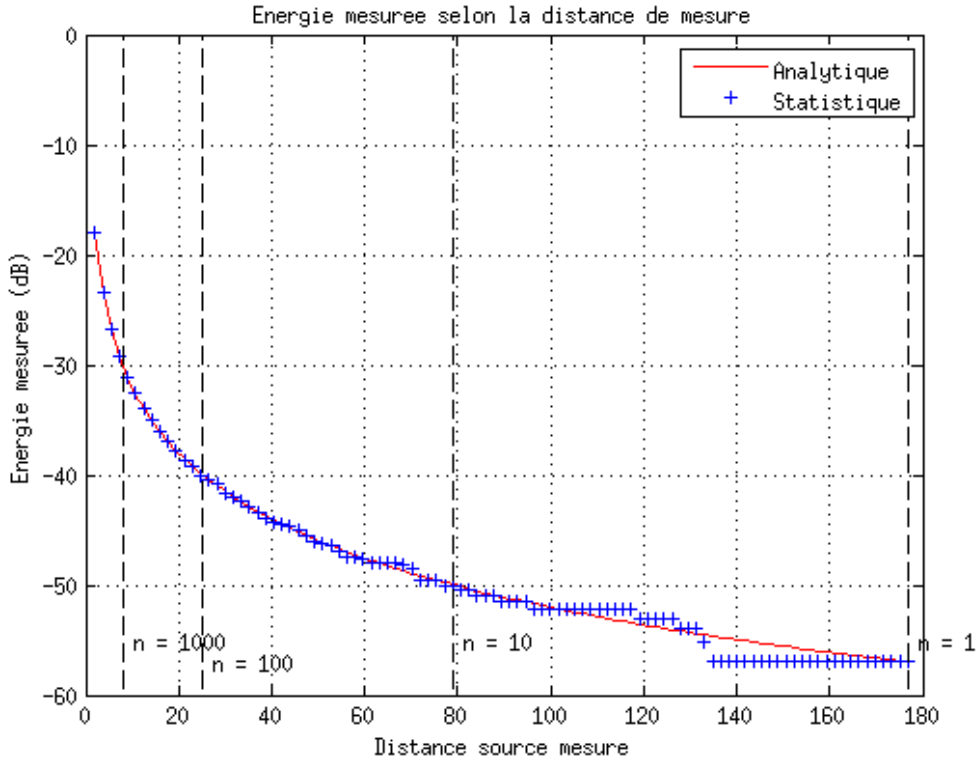


FIGURE 2 Measured energy (dB) in function of distance between x_s and x_m in meter for $r = 0.36$m and $N = 10^6$. Blue crosses stand for the statistical measure $f(r) = \frac{n(r)}{N}$ and red ligne the analytic function $f(r) = \frac{\pi r^2}{4\pi d^2}$.

In addition, figure 2 shows how this modelization fills with distance between source and measures. The accuracy of the measurement depends strongly on the number of *rays* counted, then, the more n increases, the more accurate will be the measurement. Nevertheless, in practice, values for a short distance between the source and the measurement sphere represent direct sound and first reflections, whereas long distances describe the diffuse field. Under this assumption, we can consider this model acceptable for all $n \geq 1$.

2.3 | Presence of an obstacle

For the case of acoustic propagation in presence of an obstacle, we choose to only consider specular reflections (Snell-Descartes laws). Indeed, this approximation is suitable when surfaces are large in comparaison to wavelengths, because diffraction effects can be neglected[3]. For a room, this condition is reached if :

$$ka \gg 1, \tag{14}$$

with k the wave number and a the characteristical diameter of the room[5]. This approach is currently used by room acoustic softwares (e.g. *Odeon*[6], *Grasshopper*[7], etc.) regarding to audible frequency range (62,5 to 15000Hz). In particular, as the Theater of Orange has a characteristical diameter about 50 meters, the high frequency approximation is reached.

Following the discrete model, when an incident *ray* collides a flat surface, a reflected *ray* is generated from the collision point. Noting $\overrightarrow{u_i}$ the direction vector of the incident *ray*, the reflected direction vector $\overrightarrow{u_r}$ is defined by :

$$\overrightarrow{u_r} = (\overrightarrow{u_i} \cdot \overrightarrow{T})\overrightarrow{T} - (\overrightarrow{u_i} \cdot \vec{n})\vec{n}, \tag{15}$$

with $\overrightarrow{T}$ the tangent basis and $\vec{n}$ the normal vector of the surface. Moreover, the energy of the reflected *ray* is obtained by :

$$E_r(f) = E_i(f)(1 - \alpha(f)), \tag{16}$$

Reference	Material name	62,5Hz	125Hz	250Hz	500Hz	1kHz	2kHz	4kHz	8kHz
1	100% absorbent	1	1	1	1	1	1	1	1
2	100% reflecting	0	0	0	0	0	0	0	0
107	Concrete block, coarse[1]	0.36	0.36	0.44	0.31	0.29	0.39	0.25	0.25
3000	Hollow wooden podium[2]	0.4	0.4	0.3	0.2	0.17	0.15	0.1	0.1

TABLE 1 Examples of absorption coefficient given in the onligne *Odeon* database[6].

with $\alpha(f)$ the absorption coefficient of the surface, function of the frequency f. Practically, the absorption coefficients are often given per octave bands and can be found in various databases. In Gypsilab and ..., both use the open access *Odeon* database[6] defined on eight octave bands (see table 1).

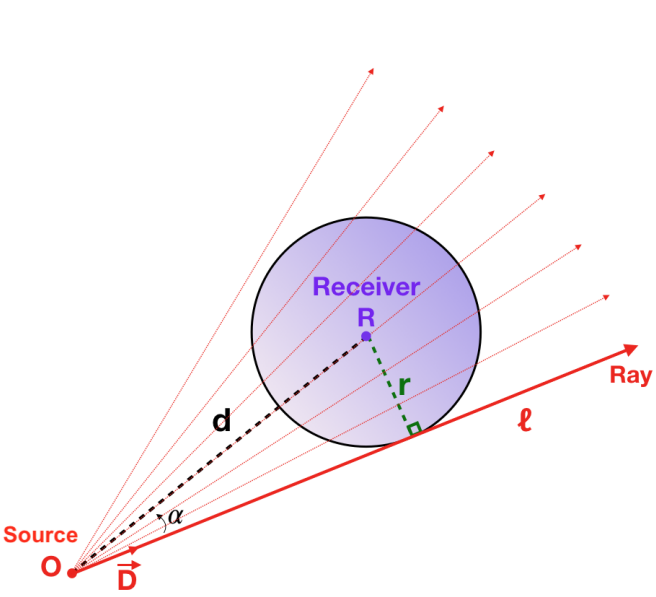
Finally, considering wall absorption, energy measured statistically (eq. 12) is extended by :

$$E_m(f) \approx \frac{n}{N}(1-\alpha(f)), \tag{17}$$

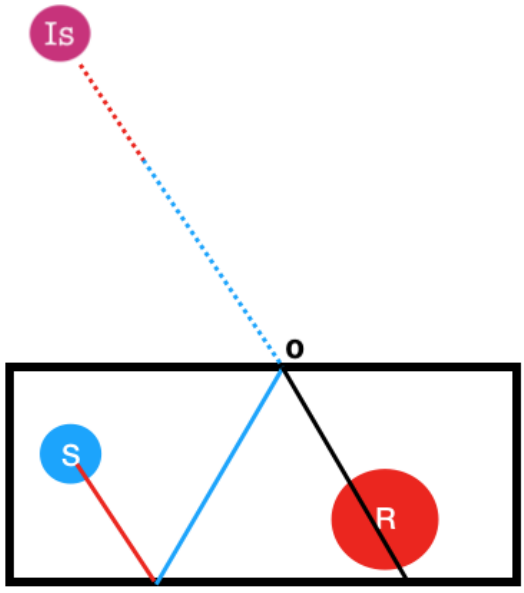
generalizable to :

$$E_m(f) \approx \frac{n}{N}\prod_{j=1}^{m}(1-\alpha_j(f)), \tag{18}$$

in the case of m reflexions.



(a) Representation of *rays* mesure by a receiver.



(b) Sketch of the creation of an image-source by successive reflections of a ray on the walls of a room

FIGURE 3 Acoustic emission from a point source.

[1]Harris, 1991
[2]Dalenback, CATT

2.4 | Image-sources

3 | IMPLEMENTATION

3.1 | Standard algorithm

3.2 | Tree-base acceleration

4 | NUMERICAL VALIDATION

4.1 | Shoes box (modele complet)

4.2 | Energy conservation (sphere reflechissante)

5 | APPLICATION TO ORANGE THEATER

6 | CONCLUSION

7 | HYBRIDE RAY-TRACING / IMAGE-SOURCE METHODE

7.1 | Ray-tracing[8]

The method developed aims to propagate rays from a point source. A ray is define with the following equation (see fig .3 a) :

$$\overrightarrow{R}(d) = O + \overrightarrow{D}.d, \tag{19}$$

where :

- O is the origine of the ray,
- $\overrightarrow{D}$ is the unitary orientation vecteur.

Once convert into cartesian coordinates and normalized we can test the intersection with the triangles of the mesh. The mesh elements have to be triangle with normales directed towards the inside of the room. By using the Moller-Trumbore fast algorithm[9], for each ray we can find the points T intersecting triangles, such as :

$$T(u,v) = (1-u-v)V_0 + uV_1 + vV_2 = O + D.d, \tag{20}$$

with (u,v) the barycentric coordinates such as :

$$\begin{cases} u \geqslant -\epsilon, \\ v \geqslant -\epsilon, \\ (u+v) \leqslant 1+\epsilon, \end{cases} \tag{21}$$

where $\epsilon = 10^{-5}$ to avoid rounding errors due to machine precision (float). This can be written :

$$\begin{bmatrix} d \\ u \\ v \end{bmatrix} = \frac{1}{(D \times E_2).E_1} \begin{bmatrix} (T \times E_1).E_2 \\ (D \times E_2).T \\ (T \times E_1).D \end{bmatrix}. \tag{22}$$

with :

$$\begin{cases} E_1 = V_1 - V_0, \\ E_2 = V_2 - V_0, \\ T = O - V_0. \end{cases} \tag{23}$$

The good intersection is the one whose the distance d between the origin of the ray and the intersection point T is the shortest. The rays can then be reflected on the face as on a mirror to find the new orientation vector :

$$\vec{r} - \vec{i} = 2 \times (-\vec{i}.\vec{n})\vec{n}. \tag{24}$$

with :

- $\vec{r}$: the reflected ray,
- $\vec{i}$: the incident ray,
- $\vec{n}$: the normal of the face.

Once we know the elements met be each ray we can update their energies. Each triangle carries an an absorption coefficient α_i for every octave band from 62,5Hz to 8kHz (so 8 bands) (see exemple on table 1). Moreover, we take into account the atmospheric attenuation. The energy of a ray is :

$$E_i = \frac{1}{N} \times \prod_{j=0}^{k} (1 - \alpha_{i,j}) \times e^{-m_i . d_{tot}}, \tag{25}$$

with :

- E_i : the energy carried by the ray on the i-th frequency band,
- E_0 : the total energy,
- k : the total number of faces encountered by the rays during its propagation,
- j : the index of the face encountered par the ray,
- m_i : the air absorption coefficient in the i-th frequency band (according to the norm ISO-9613),
- d_{tot} : the total length of the ray from the source to the center of the receiver.

7.2 | Image-sources

At each ray bounce, we check if the rays intersect the receiver-sphere. First we check if the origine point O of the ray is include in the R-center and r-radius receiver :

$$||\overrightarrow{OR}|| \leqslant r, \tag{26}$$

If not, we check the direction of the ray :

$$\cos\alpha \geqslant 0, \tag{27}$$

with α the angle between the ray $\overrightarrow{D}$ and $\overrightarrow{OR}$. Then we check if the ray is long enough to reach the receiver :

$$||\overrightarrow{OR}|| \leqslant d, \tag{28}$$

To finish, we check if the ray intersect the receiver-sphere :

$$\sin\alpha \times ||\overrightarrow{OR}|| \leqslant r \quad \Rightarrow \quad \alpha \leqslant \arcsin\frac{r}{||\overrightarrow{OR}||} \tag{29}$$

If the ray does indeed intersect the receiver, an image-source is generated (see fig. 3 b). This is the image of the sound source relative to all the walls encountered by the ray. The image-source I_S is located in space by back-propagation of the ray from its last origin point O :

$$\overrightarrow{I_S O} = \overrightarrow{D}.d_{tot} \qquad \Rightarrow \qquad I_S = O - \overrightarrow{D}.d_{tot}, \tag{30}$$

where

- $\overrightarrow{D}$ is the last unitary orientation vector of the ray,
- d_{tot} is the total distance travelled by the ray from the original source S to the point O.

The positions of the image-sources allow to know the time it takes for the signal to arrive to the receiver :

$$t_{I_S} = \frac{||\overrightarrow{I_S O}|| + ||\overrightarrow{OR}||}{v}. \tag{31}$$

with v the sound speed in the medium (air : $v = 340m/s$). The room impulse response can be generated for each octave band at a certain sampling frequency f_s such as :

$$E_i = \sum E_j, \tag{32}$$

with :

- the integer part of the product $(t_j \times f_s) = i$,
- E_i : the energy of the i^{th} sample,
- E_j and t_j : respectively the energy and travel time of the j^{th} image-source.

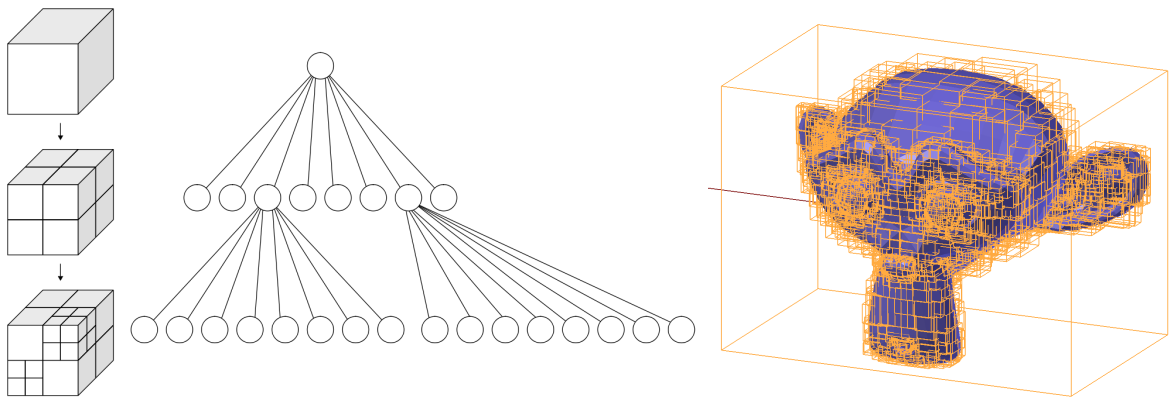
8 | ALGORITHM'S OPTIMISATION

8.1 | Presentation

The algorithm has iteration loops with different complexities. There are three main steps which depend on the number of mesh elements M and the number of rays emitted N :

- the mesh loading,
- the intersection between rays and faces,
- the image-sources creation.

The mesh loading depends only on the number of elements and is done only once at the beginning of the algorithm. The image-sources creation depends only on the number of rays and is done until the stop threshold is reached. The most critical stage is the intersection rays/faces because each ray has to be tested with each face and the loop is repeated until the stop threshold is reached. The complexity of this operation is quadratic in $O(N \times M)$. For a significant number of mesh elements ($> 100\,000$ for the Orange theater) and a lot of rays (necessary to make the measurement accurate) the calculation time is very long, which makes the tests tedious. To alleviate this problem we have developed a fast algorithm based on a "Divide and Conquer" approach based on Octree spatial cutting [10].



(a) Illustration of the Octree principle. **(b)** Octree around an arbitrary mesh

FIGURE 4 Illustration of an Octree.

The general principle consists in creating a cubic box called a "mother-box" containing all the mesh elements, i.e. all the triangular faces. This mother-box is then subdivided to create eight "daughter-boxes" of identical size which themselves will be subdivided into eight daughter-boxes, etc. (see fig. 4 a). Recursively, each mother-box element will be stored in the daughter-box that contains it. In this way, you descend into the tree structure until a stop condition is reached. Typically, the octree stops when no more daughter-boxes contain more than n items. The boxes are therefore refined in the same way as the mesh (see fig. 4 b) since empty boxes do not generate daughter-boxes.

If we name *item* a ray or a triangular face of the mesh and *operation* the storage in a box, we can calculate the number of *operations* to have only one *item* per box. We place ourselves in the case where the *items* are distributed in a uniform way in space. At level-0 all the N *items* are in the root-box. At level-1 we test each daughter-box with each *item* and we have $8N$ *operations*. At the next level each daughter-box become a mother-box and we can apply the same calculation to obtain :

$$8 \times 8 \times \frac{N}{8} = 8N \ operations.$$

because each mother-box only count $\frac{N}{8}$ *items*. Then, at level-p we will need :

$$8^p \frac{N}{8^{p-1}} = 8N \ operations. \tag{33}$$

and the total number of *operations* is $p \times 8N$.

Moreover we said that their is only one *item* per box, so there are as many boxes as there are *items*. We can write :

$$\begin{aligned} 8^p &= N, \\ p.\ln 8 &= \ln N, \\ p &= \frac{1}{\ln 8} \ln N. \end{aligned} \tag{34}$$

The total number of operation is then :

$$C = p.8N = \frac{8N}{\ln 8} \ln N. \tag{35}$$

Within the algorithm, instead of testing all rays with all faces, we test N times one ray with one face. The number of linear operations is :

$$C_{tot} = \frac{8N}{\ln 8} \ln N + N. \tag{36}$$

which is faster than N^2 for $N >> 1$.

In practice we will be able to stop the octree before arriving at only one element per box. Typically, if we stop the octree at n elements per box the total number of operation become :

$$\begin{aligned} C_{tot} &= \frac{8N}{\ln 8} \ln N - \frac{\ln n}{\ln 8} 8N + n^2, \\ &= aN \ln N + bN + c. \end{aligned} \tag{37}$$

with (a, b, c) some constants. So n needs to be very small in front of N to preserve the performance.

Note that when the distribution of items is not uniform the demonstration is more complicated but the calculation times remain substantially similar.

8.2 | Implementation

To store the triangular faces in the box we test if the center of the face belongs to the box. Once each center has been store in the leaves of the octree (i.e the last boxes of the branches) we resize the leaves to embody the whole faces they contains. To test if a ray cross a box we use a pass/fail algorithm conceived for Axis-Aligned-Bounding-Box [11]. This kind of box allow to simplify a lot the computation.

We can can then measure the computation time of one iteration (i.e all rays are intersecting a triangle) by increasing the number of rays and the number of faces in the mesh. As we can see in the figure 5 the complexity of the algorithm is almost linear by using the octree method. This allows to treat large meshes with millions of rays by maintaining a low computation time. In particular, we can see in the table 2 that for 250 000 rays and faces the computation time is divided by 1000 by using the octree.

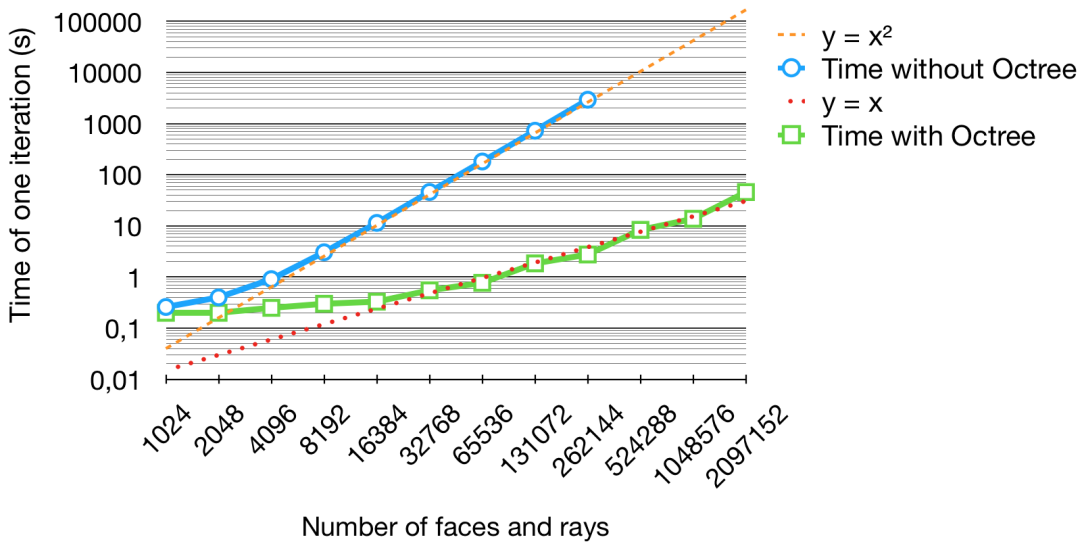


FIGURE 5 Computation time for one iteration and $N = M$ (log scale)

Number of faces and rays	Time **without** Octree (s)	Time **with** Octree (s)
2^{10} (=1 024)	0,26	0,2
2^{11} (=2 048)	0,4	0,2
2^{12} (=4 096)	0,91	0,25
2^{13} (=8 192)	3,05	0,3
2^{14} (=16 384)	11,44	0,33
2^{15} (=32 768)	46,02	0,55
2^{16} (=65 536)	181,61	0,77
2^{17} (=131 072)	725,17	1,85
2^{18} (=262 144)	2927,9	2,76
2^{19} (=524 288)	X	8,36
2^{20} (=1 048 576)	X	13,78

TABLE 2 Computation time for one iteration and $N = M$

9 | VALIDATION

In order to validate our approach and our method we compare the experimental results with theoretical results.

9.1 | Quadratic decrease

First, to validate the quadratic decrease of the energy (see section 2) we observe the room impulse response of a cubic room whose all walls are 100% absorbant except the walls on the x-axis (this may remind a Fabry-Pérot interferometer). This particular room stands for a free space simulation where the distance between the source and the receiver is regularly increased. Furthermore, since the rays spread on the x-axis and -x-axis and return in phase, the number of rays received is twice the number of rays in free space. The room impulse response (see fig. 6) follows the function $f(x) = \frac{2}{x^2}$ which is the expected behavior. In free space, the number of rays collected stands for the surfaces ratio and the quadratic decrease :

$$\frac{n}{N} = \frac{\int_s dS}{\int_\sigma dS} = \frac{\pi r^2}{4\pi d^2}, \tag{38}$$

with :

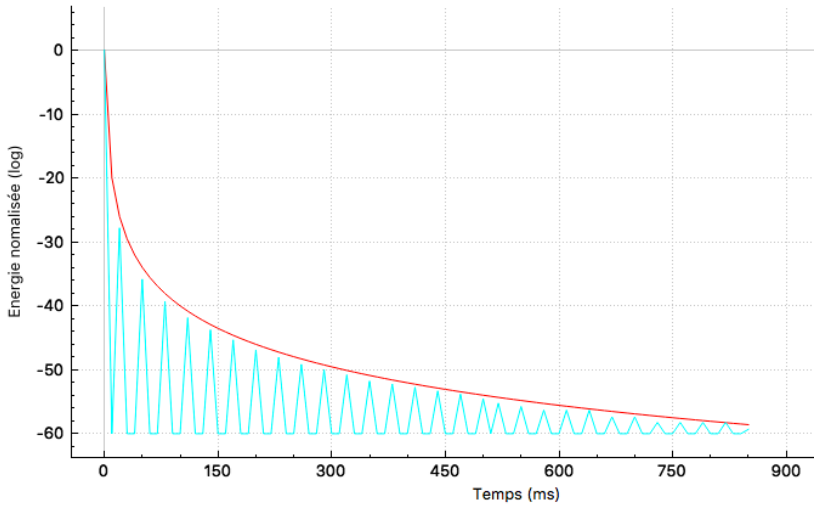


FIGURE 6 RIR in a Fabry-Pérot-like configuration for 3 millions of rays (blue) sampled at 100Hz and $f(x) = \frac{2}{x^2}$ (red)

- n : the number of rays collected,
- N : the total number of rays,
- s : the constant surface of the receiver (disk) collecting rays,
- σ : the emission sphere surface,
- r : the constant radius of the receiver,
- d : the emission sphere radius (i.e the distance between the source and the receiver).

9.2 | Energy conservation

The second test allows to simulate the conservation of the energy. We use a 100% reflecting sphere (2m radius) pretty well refined (300 000 faces) and position the source and the receiver in the center. At each iteration all the rays refocus in the center of the sphere and then are captured by the receiver. We observe the expected result as the room impulse response is a Dirac comb (see fig. 7 a) and the image-sources are positioned on spheres whose radius doubles at each iteration (see fig. 7 b). By adding the air absorption (see fig. 8) we can observe that the highest frequencies decrease faster than the lowest which reflect the nature behavior.

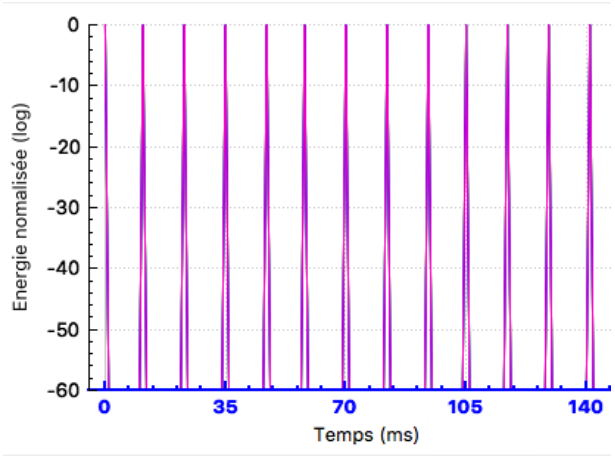
9.3 | Shoe box case

To finalize the validation algorithm we compare the results of a shoe box type room with an analytic computation. The image-sources can be positioned in space with the following formula[12] :

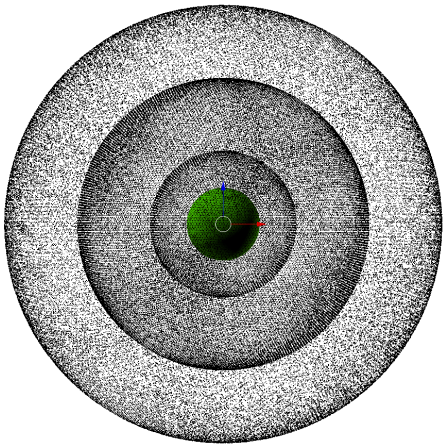
$$P_{is} = i \times D + P_s \times (-1)^i, \tag{39}$$

with :

- $i \in (-n, n)$ and $n \in \mathbb{N}$,
- P_{is} : the image-source position coordinate on X, Y or Z,
- P_s : the source position coordinate on X, Y or Z,
- D : the room dimension on X, Y or Z.



(a) RIR for a 100% reflecting sphere - 12 iterations



(b) Position of the images-sources - 4 iterations

FIGURE 7 100% reflecting sphere

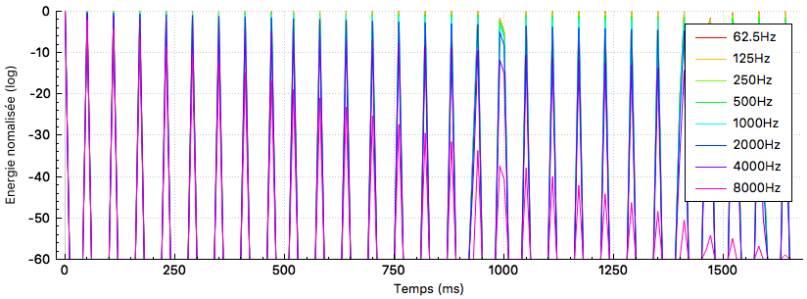


FIGURE 8 RIR for a 100% reflecting sphere with air absorption - 30 iterations

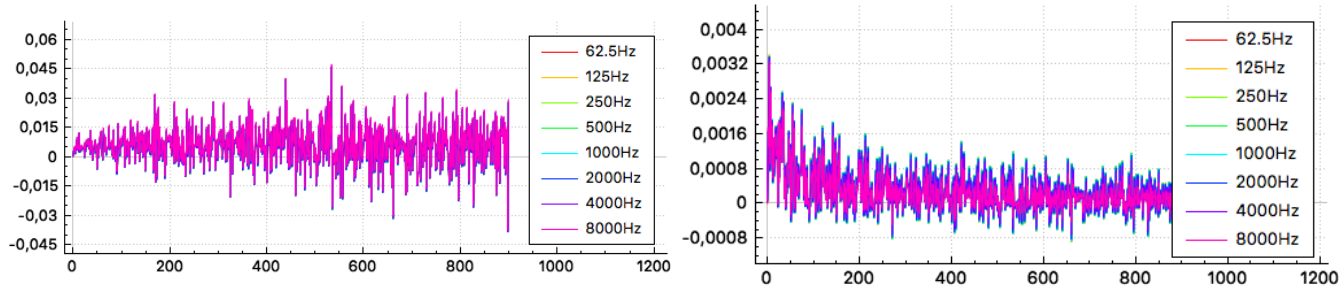
The energy of each image-source is $\frac{1}{d^2}$ where d is the distance between the image-source avec the receiver. If we compare the position of these theoretical image-sources with the image-sources obtained with the algorithm, we get the exact same result to the float precision ($10^{-6}m$). Concerning the energy, we compare two kind of errors : the relative error such as :

$$\epsilon_{rel} = \frac{|E_{exp} - E_{theo}|}{E_{theo}}, \tag{40}$$

and the infinity norm error which express the fact that the further away the image-source is from the receiver, the less important the error will be for the final result.

$$\epsilon_{\infty} = \frac{|E_{exp} - E_{theo}|}{\max(E_{theo})}. \tag{41}$$

We can also add some absorption coefficient on the walls and take into account the air absorption. We can see in the figure 9 a that the relative error of the energy image-source per image-source remains always below 5%. In the same way we see in the figure 9 b that the infinity norm error is below $0,3\%$ for all frequencies.



(a) Relative error

(b) Infinity norm error

FIGURE 9 Error for each image-sources with walls and air absorption - 1 000 000 rays

10 | DEVELOPED SOFTWARES

The room acoustic tool is available in two forms.

10.1 | Matlab library

First a Matlab library [13] ...

10.2 | Blender add-on

In a second hand the tool is also available as a Blender add-on. The user can work on the CAD software to model the room under test, positioning the sources and the receiver and assign materials to the walls. By clicking on the "Run" button, the mesh is exported, the materials are linked to eight absorption coefficients (extracted from a data base) and the acoustic calculation tool is launched. This is an executable C++ complied software which treats information form Blender and generates the room impulse response. The communication between the CAD and the executable is done thanks to an .obj file, so using Blender is not necessary. Different options allow to analyse the results by reimporting rays or image-sources on the CAD software. It is also possible to listen an audio file convolved to the RIR to listen the reverberate sound.

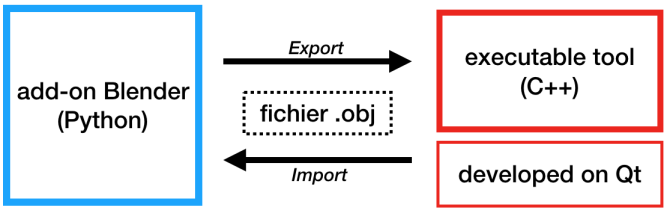


FIGURE 10 Software architecture

11 | APPLICATION TO THE ANTIC THEATRE OF ORANGE

The acoustic simulation can be done on the ancient theater of Orange. Because it is an open room the software automatically add a 100% absorbant box around the building to be sure to always count all rays. We can then calculate the image-sources and the RIR for different configurations of the theatre or different materials. Indeed, archeologists want to explore some architecture hypothesis from missing part of the theatre. An acoustical analysis can allow to understand some behaviors like : the influence

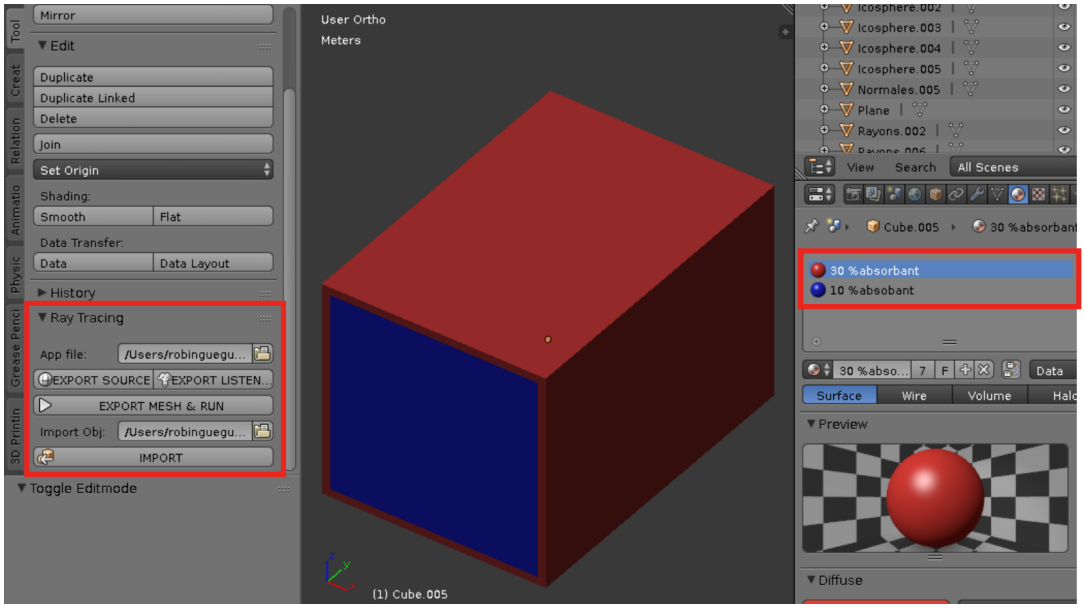


FIGURE 11 Add-on Blender for exporting/importing meshes and running computation tool

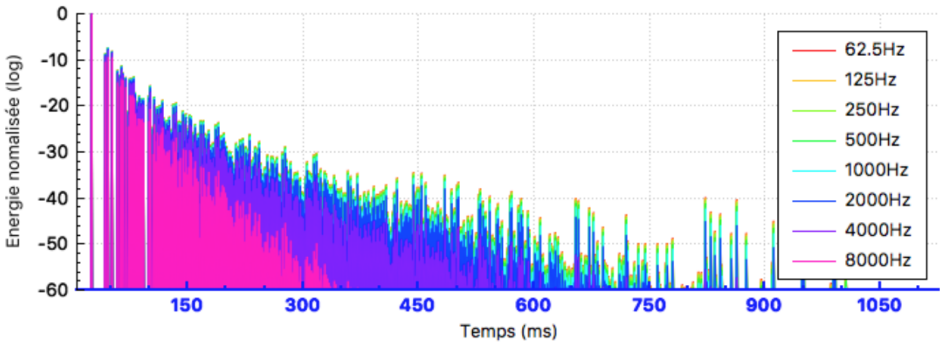


FIGURE 12 Room impulse response of the antic theater of Orange

of the position of the spectators in the bleachers, the shape of the roof, the materials of the *orchestra*, etc. With a 600 000 faces theatre (i.e including decoration elements of the stage wall), the RIR at RT_{60} is generated in 20 minutes for one million rays (see fig. 12). Each iteration is done in 25s so it really depends on the materials chosen. We can note that the more details of the mesh are refined the more we can simulate diffraction effect. Indeed, in high frequency, small detail elements will be able to reflect the rays in different directions which can resemble diffraction effects. In the theater it will be really interesting to study where the reflection come from. Thus, we project the image-sources on the wall of the theater to understand what wall is the main contributor to the energy received (see fig. 13).

12 | CONCLUSIONS

We presented the problems raised by an acoustic study of an ancient monument. The complex geometry of this type of building and their colossal size requires the use of approximate calculation methods. Thus, by simulating the reflections and absorptions of the walls, it is possible to study the reverberation of a room. Despite the inevitable approximations of the model, we have proved that the laws of physics are respected. A fast algorithm has been implemented to allow users to easily and quickly test their architectural assumptions. Thus, the calculation time becomes linear to the product number of mesh elements/number of rays. The algorithm developed allows the study of the temporal graph of reverberation of the building as well as the position in

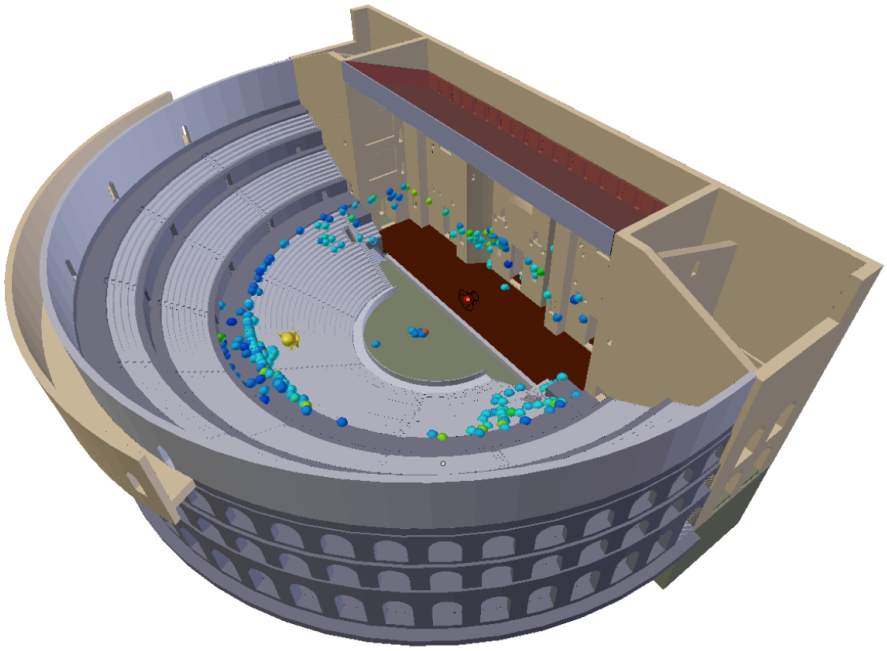


FIGURE 13 Images-sources projected on the theatre of Orange

space of the various sound reflections. Furthermore, a sound signal could be heard in three dimensions thanks to binaural filters and steering control can be done with "Head Tracker" mounted on a headset.

However, there are many opportunities for improvement that remain under consideration for this type of software tool. First, in a context where virtual reality is becoming more and more important in today's applications, we could consider moving the listener in real time and thus, allow a complete virtual tour of the building. Secondly, from the point of view of the analysis results, there are many possible improvements at the graphic level. That raises some questions. How to view acoustic calculation results? What information is essential for an archaeologist wishing to study the acoustics of a monument? Similarly, is it essential to add diffraction effects to the model? If so, what is the best method? Could certain acoustic behaviours be treated locally and then inserted into the model by ray tracing? Finally, it would also be interesting to use sources whose directivity is not uniform. This would be more representative of the real cases, and in particular, of the use made in Orange at the theatre origin. The sounds were then emitted by musical instruments or by the human voice possibly amplified by a mask.

ACKNOWLEDGMENTS

The authors would particularly like to thank François Alouges, Titien Bartet, Pascal Frey and Emmanuelle Rosso for all the help they each provided at the different stages of this project. Thanks also to Jean-Dominique Polack for his wise advice on architectural acoustics.

References

1. Maufras M. Ch.-L.. *Vitruve : De l'architecture.* Panckoucke; 1847.
2. Gueguen Robin. Virtualisation visuelle et auditive du théâtre antique d'Orange. PhD thesisSorbonne Université2018.
3. Jouhaneau Jacques. *Acoustique des salles et sonorisation* Acoustique appliquée, vol. 3: . Conservatoire national des arts et métiers; 1997.
4. Keinert Benjamin, Innmann Matthias, Sanger Michael, Stamminger Marc. Spherical Fibonacci Mapping. *ACM Transactions on Graphics.* 2015;34.

5. Perrey-Debaina E., Yanga M., Nennigb B., Chazota J.-D.. Approximation par ondes planes et son utilisation pour la méthode des éléments finis. In: ; 2016.

6. Odeon . Odeon webpage .

7.

8. Kulowski Andrzej. *Algorithmic representation of the ray tracing technique*. Applied Acoustics; 1985.

9. Möller Tomas, Trumbore Ben. Fast, minimum storage ray-triangle intersection. *Journal of Graphics Tools.* 1997;2(1):21-28.

10. Grenngard L., Rokhlin V.. A Fast Algorithm for Particle Simulations. *Journal of computational physics.* 1987;73(2):325-348.

11. Williams Amy, Barrus Steve, Morley R.Keith, Shirley Peter. An Efficient and Robust Ray–Box Intersection Algorithm. *ACM SIGGRAPH.* 2005;(9).

12. McGovern Stephen. Fast image method for impulse response calculations of box-shaped rooms. *Applied Acoustics.* 2009;70(1).

13. GypsiLab, un nouvel outil de calcul FEM/BEM pour l'acoustique numérique.(Le Havres)Actes du Congrès Français d'Acoustique2018.

How to cite this article: M. Aussal, and R. Gueguen, (2018), Room acoustic measurement tool for complex geometry, *International Journal for Numerical Methods in Engineering*, *2018;00:1–6*.