

ARTICLE TYPE

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

Matthieu Aussal¹ | Robin Gueguen²

¹Centre de mathématique appliquées, École Polytechnique, 91128 Palaiseau, France

²Institut 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

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 achieve¹. In this study, we focus on the ancient theater of Orange which has a significant size (100m wide), a complex geometry (ornements, bleachers, columns, arches, etc.) and which is open-air. In a previous work, a complete mesh was designed using *Blender* CAD software², according to the most recent archeological knowledge³. To be representative, this mesh has hundreds of thousands of elements (triangular faces), with complex shape (see fig. 1). 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 prototyping⁴. This preliminary work was usefull to construct and validate ideas and algorithms. It conducts to the creation of a new toolbox, *openRay*, now added to the master branch of *Gypsilab* and freely downloadable⁵. In a second step, all algorithms

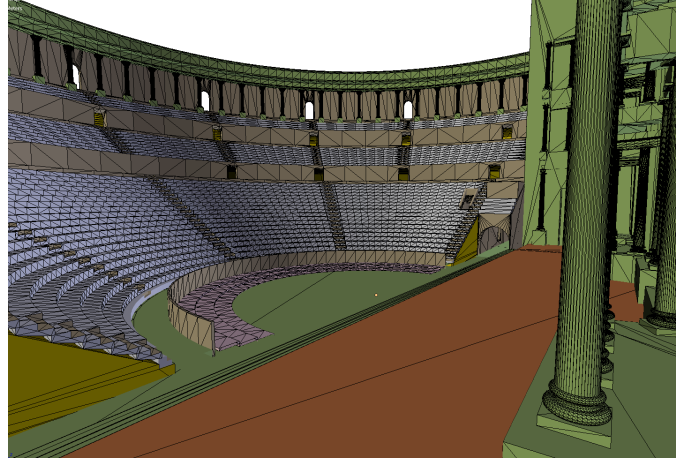


FIGURE 1 Mesh of restituted theater of Orange modeled on *Blender* (436 000 triangles).

were retranscribed in C++ using Qt Creator, leading to an autonomous library *Just4RIR*. 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.

1 | ACOUSTICAL ENERGY MODELIZATION

1.1 | Continuous domain equation

By modelling a point sound source as a localized pulse in space, the associated acoustical energy $E(t)$ propagates⁶ over time on a spherical surface $S(t)$, such that :

$$E(t) = E_0 \int_{S(t)} \vec{I}(t) \cdot \vec{ds} \quad \forall t > 0, \quad (1)$$

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 :

$$\int_{S(t)} \vec{I}(t) \cdot \vec{ds} = 1 \quad \forall t > 0. \quad (2)$$

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

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

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 :

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

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 :

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

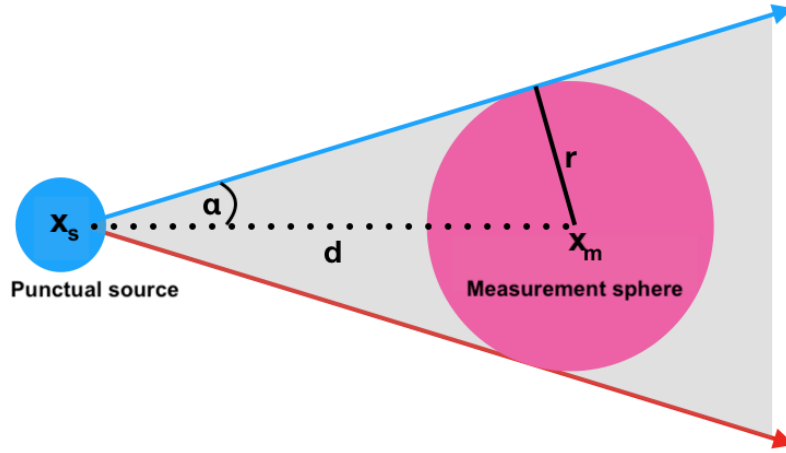


FIGURE 2 Representation of a r -radius measurement sphere centered in x_m , receiving energy from a sound source in x_s .

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 $E_0 = 1$.

1.2 | 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 \vec{u}_i ,
- An energy value E_i .

For example, with an omnidirectional source¹, *rays* are given by :

- The source coordinate ($x_i = x_s, \forall i \in [1, N]$),
- A unit sphere uniform sampling (e.g. icosahedre subdivision, Fibonacci's rule⁷, 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 a n -rays beam that intersect this sphere to calculate the acoustic energy E_m at point x_m :

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

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

$$E_m \approx \frac{n}{N}, \quad (7)$$

which means that the measured energy E_m is statistically represented by the ratio between the number of *rays* forming a beam to the total number of *rays*.

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

$$E_m = \frac{1}{4\pi} \Omega_m, \quad (8)$$

¹For a directional source, a non-uniform spatial sampling may be used.

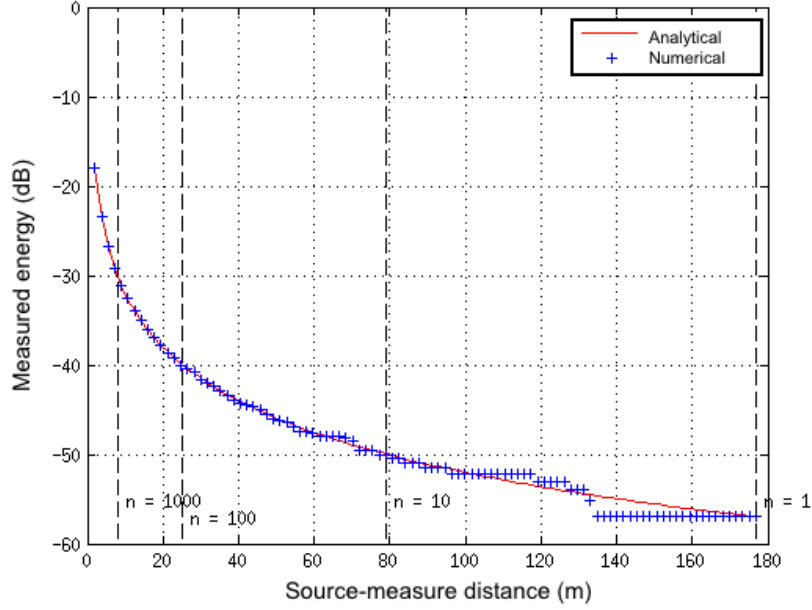


FIGURE 3 Measured energy (dB) in function of distance between x_s and x_m in meter for $r = 0.36\text{m}$ and $N = 10^6$. Blue crosses stand for the statistical measure $f(r) = \frac{n(r)}{N}$ and red ligne the analytic function $f(r) = \frac{\pi r^2}{4\pi d^2}$. (computed on *Gypsilab*)

where Ω_m is a solid angle of a cone of revolution (see fig. 2), such as :

$$\Omega_m = 2\pi(1 - \cos \alpha). \quad (9)$$

Following the figure 2 :

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

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

$$\Omega_m = \pi \frac{r^2}{d^2}. \quad (11)$$

At the end,

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

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

$$d \leq \frac{r}{2} \sqrt{\frac{N}{n}}. \quad (13)$$

In addition, figure 3 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 beam such as $n \geq 1$.

1.3 | 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 neglected⁶. For a room, this condition is reached if :

$$ka \gg 1, \quad (14)$$

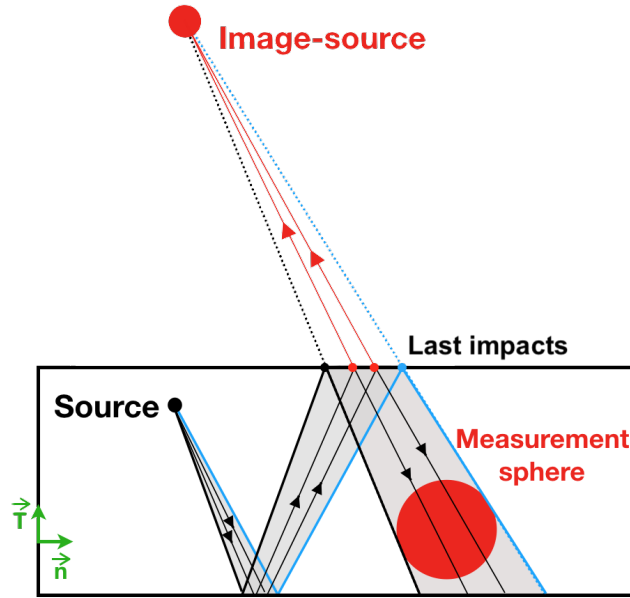


FIGURE 4 Sketch of the creation of an image-source by successive reflections of a ray on the walls of a room.

with k the wave number and a the characteristic diameter of the room⁸. This approach is currently used by room acoustic softwares (e.g. *Odeon*⁹, *Grasshopper*¹⁰, etc.) regarding to audible frequency range (62,5 to 15000Hz). In particular, as the theater of Orange has a characteristic 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 \vec{u}_i the direction vector of the incident *ray*, the reflected direction vector \vec{u}_r is defined by :

$$\vec{u}_r = (\vec{u}_i \cdot \vec{T})\vec{T} - (\vec{u}_i \cdot \vec{n})\vec{n}, \quad (15)$$

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

$$E_r(f) = E_i(f)(1 - \alpha(f)), \quad (16)$$

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 *Just4RIR*, both use the open access *Odeon* database⁹ defined on eight octave bands (see table 1).

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

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

generalizable to :

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

in the case of m reflexions.

1.4 | Image-sources

Although the extended formulation (eq. 18) may be sufficient to generate room acoustic data, we also construct images-sources from the path of *rays*. To this end, when *rays* intersect a measurement sphere and following the reverse return principle, they are retro-propagated along the last direction vector. Thus, from this measurement sphere, *rays* focus on punctual images-sources (see fig. 4). Each image-source is then located relatively to the listener and carry an energy according to equation (18). By

²Harris, 1991

³Dalenback, CATT

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 ²	0.36	0.36	0.44	0.31	0.29	0.39	0.25	0.25
3000	Hollow wooden podium ³	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 onlign *Odeon* database⁹.

noting $(x_s)_{s \in [1, N_s]}$ the relative position of the N_s image sources and $(E_s)_{s \in [1, N_s]}$ the associated energy, couples $(x_s; E_s(f))_{s \in [1, N_s]}$ contain many useful informations for room acoustic analysis and auralization.

First of all, relative distance of each source image $(d_s)_{s \in [1, N_s]}$ can be computed. This distances should be used to compute air absorption, adding a term in the equation (18) :

$$E_s(f) \approx \frac{n}{N} e^{-\beta(f)d_s} \prod_{j=1}^m (1 - \alpha_j(f)), \quad (19)$$

with $\beta(f)$ a frequency dependent coefficient (ref norme). Furthermore, fixing the sound celerity c , room impulse response can be generated, converting each distances d_s in time of arrival. Taking care to convert energy into sound pressure ($p = \sqrt{E}$), finite impulse response can be generated and analyzed using standard metrics (e.g. T_{30} , C_{80} , D_{50} , etc.). For auralization, this room impulse response is convolved with an audio signal in order to listen the acoustical rendering. In particular, this convolution can involve relative position of predominant images sources, in order to realize a spatialized auralization with multichannel or binaural renderers. Finally, to complete acoustic studies with visual analysis, images sources can be projected on the room used for computation to see where are located listened reflections (see last impact on fig. 4).

2 | IMPLEMENTATION

2.1 | Standard algorithm

As standard principles are introduced, we focus now on the numerical implementation of an acoustic renderer by ray-tracing. Before any acoustic computation, a numerical room has to be modeled with surfaces and materials. In our case, we use simplex representation with meshes composed of flat triangles. From it, geometrical intersection are computed between *rays* (represented by oriented line (L)) and mesh elements (represented by piece of plan (P)), using parametric equations :

$$(L) : a + \delta \vec{u}, \quad \delta \in \mathbb{R}, \quad (20)$$

$$(P) : b + \lambda \vec{v} + \mu \vec{w}, \quad \lambda, \mu \in \mathbb{R}. \quad (21)$$

The following conditions determine pairs (*rays*;elements) with unicity :

- $0 < \lambda \leq 1$ and $0 < \mu \leq 1$ to ensure that *ray* is inside the triangle,
- $\delta > 0$ to respect the propagation direction,
- δ minimum not to go throught the mesh.

Practically, to find these pairs, we can solve directly underlying linear system with or use the Moller-Trumber algorithm¹¹. For N *rays* and M triangular elements, this process has a numerical cost quadratic proportional to NM , which is critical if both N and M are large (see section 2.2). Once all pairs found, energy measurement has to be done in order to build images-sources (see section 1.4). To this end, *rays* are intersected to the measurement sphere $S(x_m, r)$ using its cartesian representation :

$$(x - x_m)^2 - r^2 = 0, \quad \forall x \in \mathbb{R}^3, \quad (22)$$

which leads to a linear numerical cost proportional to N .

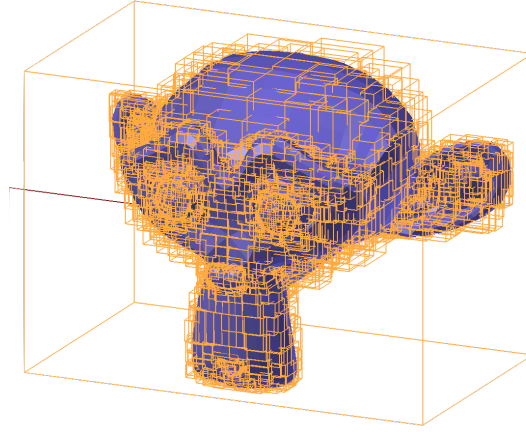


FIGURE 5 Binary tree leaves of an arbitrary mesh from *Blender*. (computed on *Just4RIR*)

Finally, a *ray* is reflected according to equation (15) and propagated while its distance travelled verify condition (13). This iterative strategy ensure the energy propagation by the elimination of all *rays* that would be in non-measurable beams. In the particular case of an open-air room, *rays* which don't encountered surface of the mesh are also eliminated. Once all *rays* are eliminated, images-sources can be built and post-treated (room impulse response, auralization, etc.).

2.2 | Tree-base acceleration

The most critical stage of the standard algorithm is the research of intersections between *rays* and triangular elements, leading to a quadratic complexity $O(N \times M)$. Indeed, each *ray* has to be tested with each face, for each iteration of the ray-tracing algorithm. For a large number of mesh elements (e.g. $M > 10^5$ for the Orange theater) and a lot of *rays* to ensure reasonable accuracy (typically $N > 10^6$), the calculation time may be prohibitive. To alleviate this problem, a "Divide and Conquer" approach using binary trees is performed (ref cooley tuckey fft, hierarchical matrix hackbush). The general principle consists in creating a mother-box, containing all the mesh elements. This mother-box is then subdivided along the main dimension to create two daughter-boxes, each with the same number of elements (median spatial subdivision). This process is then applied recursively, until a stopping criterion is reached. In our case, we stop when leaves contain only one element (see fig. 5). This hierarchical tree is completely mesh dependent, computed in $O(M \log M)$ operations, and gives a structure to fastly navigate inside the mesh. Thus, at each ray-tracing iteration, rather than face off all *rays* against all elements, this structure is used to accelerate this process.

More precisely, we first initialize the ray-sorting considering a bounding box, containing all *rays* and all elements. After the first median subdivision, we have to distribute *rays* inside the two new boxes. This stage is done in $O(N)$ operations using W.Amy *et al.* algorithm¹². Each box has N_1 and N_2 *rays*, such as $N = N_1 + N_2$. Assuming this subdivision performed recursively to the p -level, each box contains (N_i) *rays* such as:

$$N = \sum_{i=1}^{2^p} N_i. \quad (23)$$

Then, ray sorting at the $(p+1)$ -level also conducts to $O(N)$ operations. To reach the leaves-level, we have to perform $O(N \log M)$ operations, where $\log M$ is close to the depth of the binary tree. At the end, as we have only one element per box, the ray-element intersection only needs $O(N)$ operations. Finally, instead of $O(NM)$ operations, we compute ray-tracing algorithm in :

$$O(M \log M) + O(N \log M) + O(N), \quad (24)$$

with is a quasi-linear complexity. More over, if binary tree is precomputed, each iteration of ray-tracing just stands for :

$$O(N \log M) + O(N). \quad (25)$$

To evaluate numerically complexities with or without binary tree acceleration, we measure the computation time of one iteration by increasing the number of *rays* and the number of faces in the mesh ($N = M$). As we can see in figure 6 , the

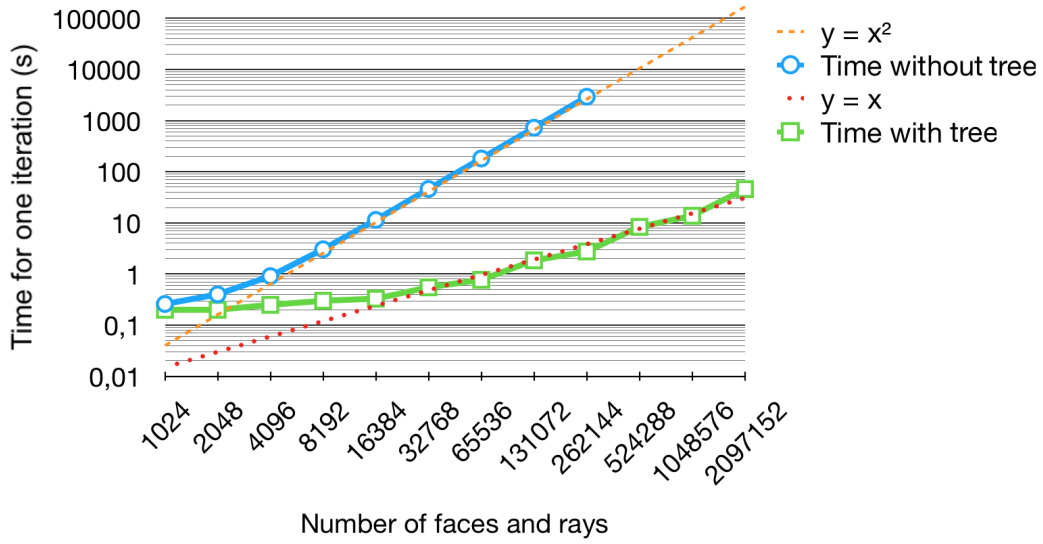


FIGURE 6 Computation time for one iteration of ray-tracing in function of the number of face and rays, such as $N = M$ (log scale). An omnidirectional source is located at the center of a mesh of a unitary tetrahedral. (computed on *Just4RIR*)

Number of faces and <i>rays</i>	Time without tree (s)	Time with tree (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 of fig. 6 on a standard laptop (2.7 GHz core and 8 Go ram).

complexity of the algorithm is therefore quite linear by using tree-based method. This allows to treat large meshes with millions of *rays*, by maintaining a weak computation time. In particular, we can see in the table 2 that for 250 000 *rays* and faces the computation time is divided by a factor of a thousand. It's important to precise that no parallelization strategy has been employed, all results are obtained in single core computation on a standard laptop (2.7 GHz core and 8 Go ram).

3 | NUMERICAL VALIDATION

To evaluate and validate methods and algorithms, several non-regression tests has been implemented. In this study, we focus only on two examples, that we hope will be the most significant. For more details, readers can refers to R. Gueguen PhD manuscript³ or *Gypsilab* source code⁵.

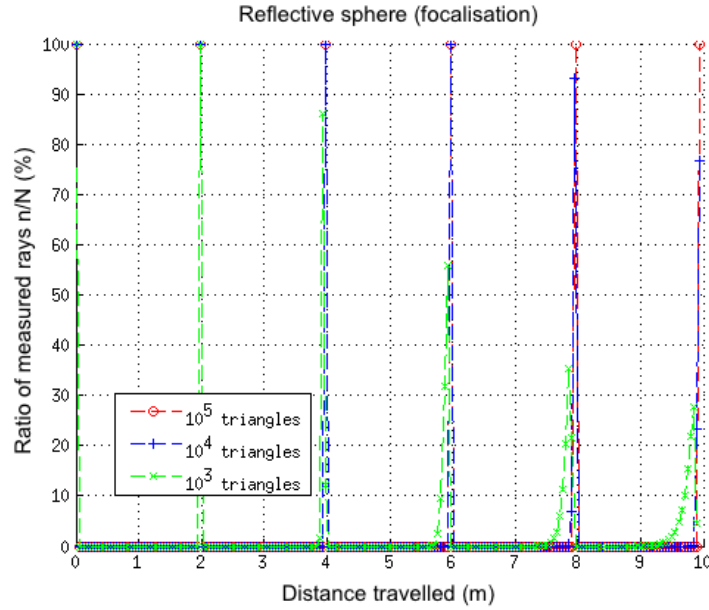


FIGURE 7 Energy measured according to equation 7 for a 100% reflecting unit sphere, with punctual source and measure located at the center (in percent). This result is given after six iterations of the tree-based algorithm, and we note as expected a two meters distance between pulses. 10^4 rays have been used. (computed on *Gypsilab*)

3.1 | Energy conservation by reflecting sphere

Firstly, we want to prove that statistical approach combined to fast algorithm is able to conserve acoustical energy (eq. 2). To do this, we consider an academic problem, where the room is a 100% reflecting unit sphere, containing punctual source and measure located at the center. If we consider only specular reflections with no air-absorption, it is expected that energy measurement takes out a Dirac comb, two meters spaced. Numerically, as we discretize the sphere with flat triangles, this focusing property clearly depend of the mesh refinement. As shown on figure 7, numerical diffusion appears in function of the distance. At the end, only the 10^5 elements mesh is able to maintain a pulsed energy in the range given, and this computation needs fast methods. Practically, this problem is representative of curved surfaces in rooms, which need high refinement to ensure a good propagation of the acoustic energy.

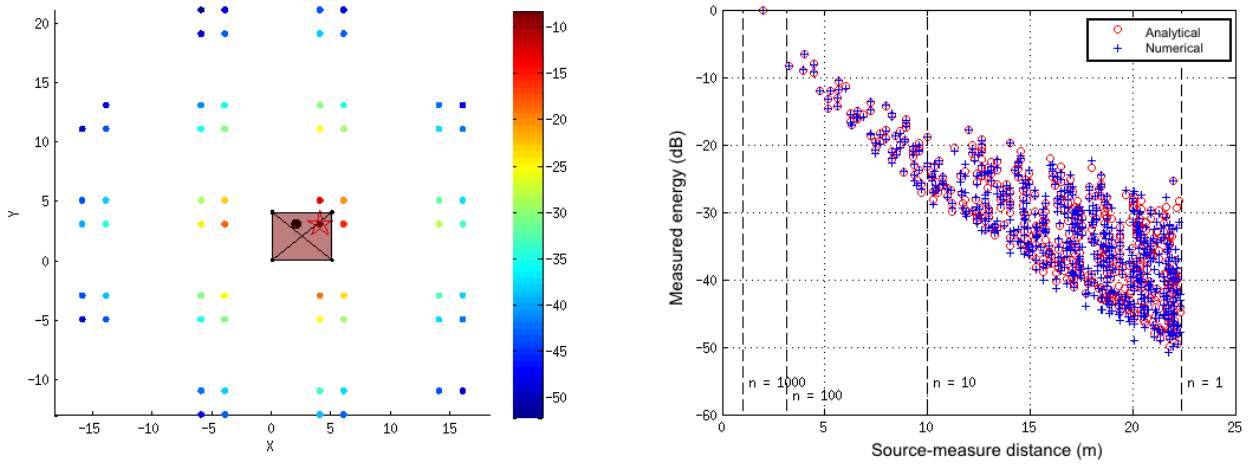
3.2 | Analytical comparison by shoes box model

Secondly, numerical computation is compared to the analytical shoes box model, introduced by S. McGovern *et al.*¹³. This model gives image-sources positions at any reflection order, in function of punctual source and measure located inside a the box. Originally, the energy associated to each image-source is given by :

- Quadratic decrease $1/d^2$, where d stands for image sources distances to measurement point,
- Absorption coefficients α for each wall, with no frequency dependance.

To use this model for our validation process, we add to McGovern solutions a frequency dependance for wall absorption, and the atmospherical impact in function of the distance. This extended model gives the energy as equation (19), but the statistical counting is there replaced by the distance quadratic decrease.

To compare analytical extended model and numerical ray-tracing computation, we use a simple triangular mesh of $M = 12$ triangular elements building a $[5, 4, 3]$ meters box. A randomly location is given for source ($x_s = [4, 2, 1.7]$ m) and an arbitrary measurement sphere is fixed ($x_s = [2, 2, 1.7]$ m and $r = 0.2$ m). Wall absorption coefficients are chosen randomly in the Odeon database⁹, and atmospherical model is used following the norm ISO-9613-1¹⁴. $N = 10^5$ rays were used and iterations were stopped according to criterion (13). As a results, figure 8 a show images sources computed which fit exactly to analytic positions (machine accuracy). As tri-dimensional representation is really confusing, we only represent images sources in the plan $z = 1.7$



(a) Image sources in the plan $z = 1.7$ m and associated energy (b) Energy impulse response in dB. Even if accuracy decreases (dB). Analytical and numerical solutions are the same, to the with the distance, far solutions remain acceptable to describe precision of the machine. the diffuse field.

FIGURE 8 Numerical solutions from ray-tracing inside a shoes box model. Comparisons with analytical solutions. (computed on *Gypsilab*)

m. Furthermore, energy value is given in decibel on the color bar, in order to visually see the distance decrease. More precisely, figure 8 b shows a comparison of the energy in decibel, in function of the distance of source images. As expected, a good matching is reached for direct sound and early reflections and diffuse field seems well approached.

4 | APPLICATION TO ORANGE THEATER

The fast ray-tracing algorithm allows acoustical computation for complex rooms as the ancient theater of Orange. Indeed, using a virtual model, archeologists can explore many architecture hypothesis. In particular, acoustical analysis allows to understand some behaviors like : the influence of the position of spectators in bleachers, the shape of the roof, the materials of the *orchestra*, etc. Moreover, it is interesting to study where reflections come from, and more generally how the theater responds to acoustical sources in various location.

First of all, a restituted version of the Orange theater has been meshed on the software *Blender* (fig. 9), according to the archeological surveys performed by l'Institut de Recherche sur l'Architecture Antique (IRAA)¹⁵. As this mesh just have to fill the geometry, there is no need of Delaunay properties, but the architecture complexity however conduct to a 436 000 faces triangulation (see fig. 1). In particular, this monument is mainly composed by :

- *Postscenium* (stage wall) partially ornamented with two basilicas on either side,
- *Pulpitum* (stage),
- *Orchestra*,
- *Cavea* (bleachers) with *Porticus* at the top (column gallery),
- Various covers (stage roof, *velum*, etc.).

Several materials are assigned to each part, in order to define specific absorption coefficients taken from Odeon database⁹ (see fig. 9). The ray-tracing solver is then used to compute the spatial impulse response. To reach the reverberation time close to -60dB (RT_{60}), we fix one million *rays* and a 2m-radius measurement sphere. The result is obtained in few minutes on a standard laptop (2.7 GHz core and 8 Go ram). At the end, all sources-images positions are generated as well as the associated multi-band acoustical energy (i.e. couple $(x_s; E_s(f))_{s \in [1, N_s]}$, see section 1.4). From it, multi-outputs are analyzed in order to study the acoustical behavior of the theater.

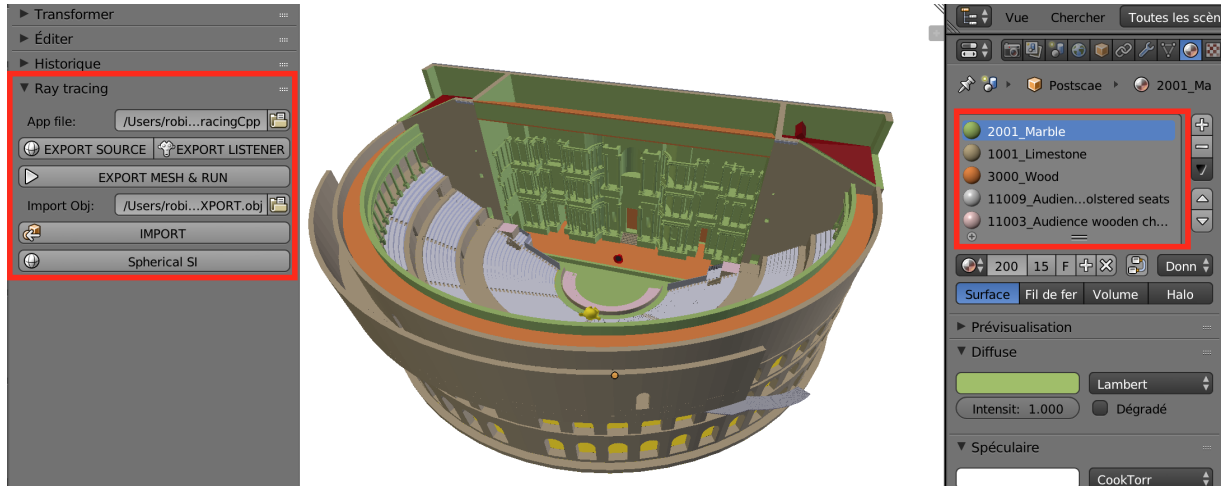
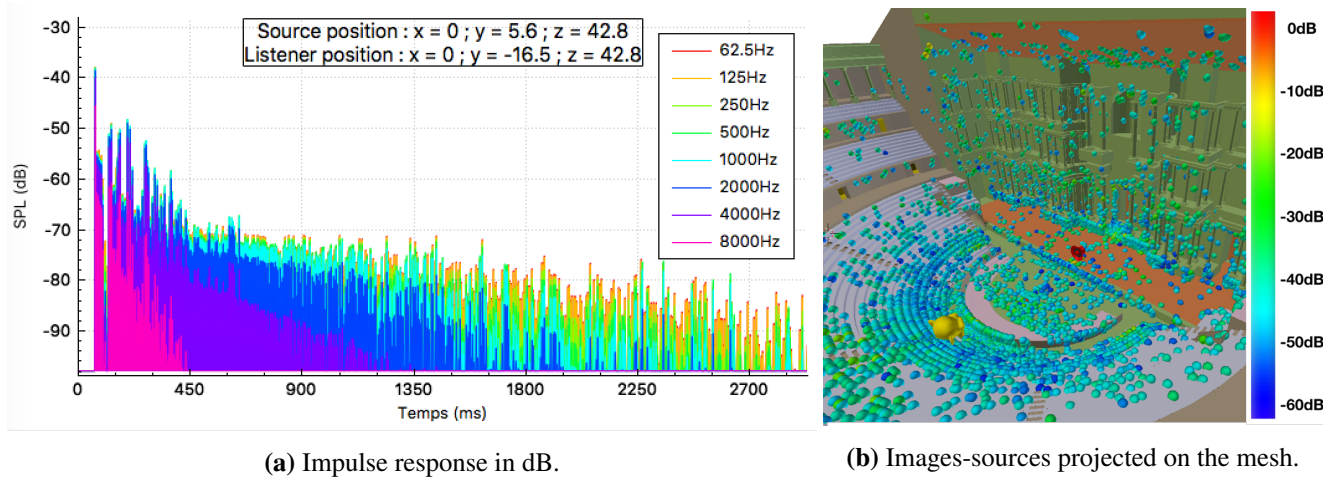


FIGURE 9 Modelization of restituted theater of Orange³ in *Blender* showing the ray-tracing add-on on the left and the material interface on the right.



(a) Impulse response in dB.

(b) Images-sources projected on the mesh.

FIGURE 10 Ray-tracing outputs for one million *rays* and a 2m-radius sphere (listener) on restituted theater of Orange. (computed on *Just4RIR*)

The following results are obtained with a source located at the front stage, at 1,60 m high (position of the mouth of an average actor). The listener is on the same axis in the bleachers. On figure 10 a, we see the multi-band impulse response of the theater until the maximum distance determined by equation 13. Primary reflections appear on the first 400 ms and diffuse field decreases according to the frequency. This difference is due to atmospheric absorption and materials properties and only frequencies above 1kHz reach RT_{60} . Furthermore, the projected images-sources represented on figure 10 b illustrate a spatial diffusion. Indeed, even if some areas carries a lot of images-sources, reflections seem to surround the listener. For a better understanding, we zoom on the top of the signal in figure 11 to display energy on a smaller range, which highlight where beams reflect on. We can notice high contribution of the *orchestra*, the wall-stage, the stage and the roof, as F. Canac demonstrated in the 60[']. To finish, perceptive factors were computed from the energy impulse responses, and given in table 3. Results correspond to a room adapted for musical playback (e.g. clarity C_{80} and reverberation time T_{30}), more than a speech transmission¹⁶. This results are confirmed by an equivalent simulation in the Odeon software, using same mesh and parameters.

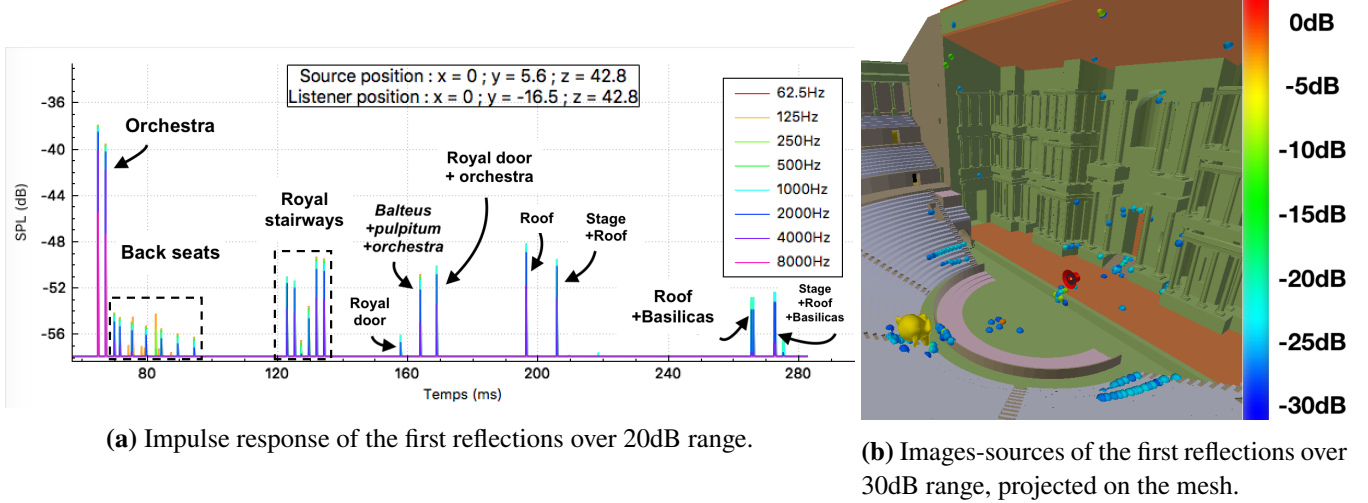


FIGURE 11 Zooming on the first reflections of fig. 10 . (computed on *Just4RIR*)

Perceptive factors	<i>Just4RIR</i>	<i>Odeon</i>
EDT (s)	1.85	2.1
T_{30} (s)	3.59	2.73
SPL (dB)	-32	-32.4
C_{80} (dB)	1.12	1.5
D_{50} (%)	47	52
T_s (ms)	117	105

TABLE 3 Perceptive factors obtained for the Orange theater, comparison between *Just4RIR* and *Odeon* software⁹. The measure is done on the 500-1000Hz band.

5 | CONCLUSIONS

In this paper, a full-chain engineering process is given, leading by an acoustical study of an imposing ancient monument. From archaeological needs, a fine mesh of the monument was constructed, associated to a complete room acoustic application suite, both for Matlab and *Blender*. The high complexity provided by this type of architecture and its ornaments leads to approximate calculation methods. Indeed, by only simulating specular reflections and wall absorption, energy propagation and measurement can be simulated by beams, carried by ray-tracing. From this representation basis, it is possible to generate a multi-band impulse response, while respecting the laws of high-frequency acoustics. Moreover, a fast algorithm with a quasi-linear complexity has been implemented, allowing users to quickly evaluate architectural assumptions, modifying their meshes regardless of the number of elements. At the end, various post-treatment has been added, as the Room Impulse Response generation, the source-image visualization, the classical perceptive factors and an auralization process.

Even if current versions of proposed softwares (*Gypsilab*⁵ and *Just4RIR*) are complete enough to be used for various studies, there are many opportunities of improvement. First of all, as image-sources positions are known, a spatial audio renderer should be added to improve current auralization tool (e.g. binaural or multichannel, eventually with trackers). Secondly, as 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. Finally, as ray-tracing modelization is an high-frequency approximation of waves phenomena, diffraction effects should be added in order to get a better fit with the physics equations.

ACKNOWLEDGMENTS

Authors particularly thanks François Alouges, Titien Bartette, 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 advices on architectural acoustics and Martin Lesellier for various contributions. This work is part of R. Gueguen PhD thesis founded by Sorbonne Université.

References

1. Maufras M. Ch.-L.. *Vitruve : De l'architecture*. Panckoucke; 1847.
2. Blender . *Manuel Blender*. 2017.
3. Gueguen Robin. Virtualisation visuelle et auditive du théâtre antique d'Orange. PhD thesis Sorbonne Université 2018.
4. GypsiLab, un nouvel outil de calcul FEM/BEM pour l'acoustique numérique.(Le Havre) Actes du Congrès Français d'Acoustique 2018.
5. Aussal Matthieu. Gypsilab .
6. Jouhaneau Jacques. *Acoustique des salles et sonorisation* Acoustique appliquée, vol. 3: . Conservatoire national des arts et métiers; 1997.
7. Keinert Benjamin, Innmann Matthias, Sanger Michael, Stamminger Marc. Spherical Fibonacci Mapping. *ACM Transactions on Graphics*. 2015;34.
8. Perrey-Debaina E., Yanga M., Nennig B., Chazota J.-D.. Approximation par ondes planes et son utilisation pour la méthode des éléments finis. In: ; 2016.
9. Odeon . Odeon webpage .
10. Robinson Philip. Acoustic Raytracer - Buzz Plug-in .
11. Möller Tomas, Trumbore Ben. Fast, minimum storage ray-triangle intersection. *Journal of Graphics Tools*. 1997;2(1):21-28.
12. Williams Amy, Barrus Steve, Morley R.Keith, Shirley Peter. An Efficient and Robust Ray-Box Intersection Algorithm. *ACM SIGGRAPH*. 2005;(9).
13. McGovern Stephen. Fast image method for impulse response calculations of box-shaped rooms. *Applied Acoustics*. 2009;70(1).
14. ISO-9613-1 . *Acoustics - Attenuation of sound during propagation outdoors*. 1993.
15. Badie Alain, Fincker Myriam, Moretti Jean-Charles, Rabatel Liliane, Rosso Emmanuelle, Tardy Dominique. *LE THÉÂTRE D'ORANGE - Rapport final d'opération - Texte*. : ; 2013. PACA Vaucluse, Orange Théâtre antique 84 087 0031 - Patriarche 9827 n2012-203.
16. Krauss Gérard, Yezou René, Kuznik Frédéric. Acoustique du bâtiment Cours-Institut national des sciences appliquées de Lyon 2009.

ANNEXE

```

%+=====+
%|
%|          This script uses the \textit{Gypsilab} toolbox for Matlab          |
%|
%| COPYRIGHT : Matthieu Aussal (c) 2017–2018.                                |
%| PROPERTY  : Centre de Mathematiques Appliquees, Ecole polytechnique,      |
%| route de Saclay, 91128 Palaiseau, France. All rights reserved.            |
%| LICENCE   : This program is free software, distributed in the hope that   |
%| it will be useful, but WITHOUT ANY WARRANTY. Natively, you can use,      |
%| redistribute and/or modify it under the terms of the GNU General Public  |
%| License, as published by the Free Software Foundation (version 3 or      |
%| later, http://www.gnu.org/licenses). For private use, dual licencing |
%| is available, please contact us to activate a "pay for remove" option.    |
%| CONTACT   : matthieu.aussal@polytechnique.edu                             |
%| WEBSITE   : www.cmap.polytechnique.fr/~aussal/gypsilab                    |
%|
%| Please acknowledge the \textit{Gypsilab} toolbox in programs or publications in |
%| which you use it.
%|-----|
%| '&' |
%| # | FILE : nrtRayCube.m
%| # | VERSION : 0.41
%| _#_ | AUTHOR(S) : Matthieu Aussal
%| ( # ) | CREATION : 14.03.2017
%| / 0 \ | LAST MODIF : 01.04.2018
%| ( === ) | SYNOPSIS : Ray tracing with absorbing cube
%| `---' |
%+=====+

% Cleaning
clear all
close all
clc

% Library path
addpath('.../openMsh')
addpath('.../openRay')

% Parameters
L = [5 4 3]
Xsrc = [4 2 1.7];
Xmes = [2 2 1.7];
Nray = 1e5
rad = 0.2

% Read mesh
mesh = msh('cube11.mesh');

% Mesh reshape to feat dimensions in L
mesh.vtx = 0.5 .* (1 + mesh.vtx);
mesh.vtx = (ones(size(mesh.vtx,1),1)*L) .* mesh.vtx;

% Initialize ray
ray = ray(mesh, Xsrc, Nray);

% Graphical sphere
[X,Y,Z] = sphere(50);
X = Xmes(1) + rad*X; Y = Xmes(2) + rad*Y; Z = Xmes(3) + rad*Z;

% Graphical representation
figure
plot(mesh)
hold on
surf(X,Y,Z,ones(size(X)))
axis equal;
xlabel('X'); ylabel('Y'); zlabel('Z');
alpha(0.3)
colorbar

% Material properties (http://www.odeon.dk/material-manufactures)

```

```

ray.msh.col(mesh.col==1) = 10;    % X = 1
ray.msh.col(mesh.col==2) = 20;    % Y = 1
ray.msh.col(mesh.col==3) = 30;    % X = 0
ray.msh.col(mesh.col==4) = 40;    % Y = 0
ray.msh.col(mesh.col==5) = 50;    % Z = 0
ray.msh.col(mesh.col==6) = 60;    % Z = 1
% ray.msh.col(:) = 60;

% Maximum distances
r1000 = rad/2 * sqrt(Nray/1000);
r100  = rad/2 * sqrt(Nray/100);
r10   = rad/2 * sqrt(Nray/10);
rMax  = rad/2 * sqrt(Nray/2);

% Ray-tracing
tic
ray = ray.tracer(100,rMax);
toc

% Images sources
tic
[img,nrg] = ray.image(Xmes,rad,rMax);
toc

% Graphical representation for images
n = size(img,1);
tmp = ones(n,1)*Xmes + img;
tmp = msh(tmp,(1:n)');
plot(tmp,10*log10(nrg(:,1)));
axis equal
colorbar

% Analytical solution
load('odeon.mat')
num = [30 ; % X = 0
      10 ; % X = 1
      40 ; % Y = 0
      20 ; % Y = 1
      50 ; % Z = 0
      60]; % Z = 1
mat = odeon.mat(num,:);
rhm = 30;
air = 5.5 * (50/rhm) .* (odeon.frq/1000).^1.7 * 1e-4;
tic
[imgRef,nrgRef] = rayCubeAnalytic(L,mat,air,Xsrc,Xmes,rMax);
toc

% Graphical representation
plot3(imgRef(:,1)+Xmes(1),imgRef(:,2)+Xmes(2),imgRef(:,3)+Xmes(3),'ok')

% Data
r = sqrt(sum(img.^2,2));
rRef = sqrt(sum(imgRef.^2,2));
sol = mean(nrg,2);
ref = mean(nrgRef,2);

% Energy error in dB
figure
plot(rRef,10*log10(ref),'or',r,10*log10(sol),'+b')
hold on
plot([r1000 r1000],[-60,0],'k--')
text(r1000,-55,' n = 1000')
plot([r100 r100],[-60,0],'k--')
text(r100,-57,' n = 100')
plot([r10 r10],[-60,0],'k--')
text(r10,-55,' n = 10')
plot([rMax rMax],[-60,0],'k--')
text(rMax,-55,' n = 1')
xlabel('Distance source mesure')
ylabel('Energie mesurée (dB)')
title('Energie mesurée selon la distance de mesure')
legend({'Analytique','Statistique'})
grid on

```

```
% Audio file
[audio,fs] = audioread('anechoicVoice.wav');

% Fir from images
T = floor(r/340*fs) + 1;
for i = 1:size(nrg,2)
    fir8(:,i) = accumarray(T,nrg(:,i),[max(T) 1]);
end

% Bank filtering
dir8 = ray.bank(256,fs);
rir = 0;
abs(sum(sum(dir8,2)) - 1)
for i = 1:size(dir8,2)
    rir = rir + fftfilt(dir8(:,i),fir8(:,i));
end

% Graphical representation
figure
plot(rir)

% Audio rendering (5 seconds)
out = fftfilt(rir, audio(1:5*fs));
sound(out, fs)

disp('~~> Michto \textit{Gypsilab} !')
```

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.