

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 (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 knowledge². To be representative, this mesh has hundreds of thousands of elements (triangular faces), with complex shape (insérer figure maillage pour montrer la complexité). 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 retranscripted 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.

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 $E(t)$ propagates³ 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 :

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

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 source, *rays* are given by :

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

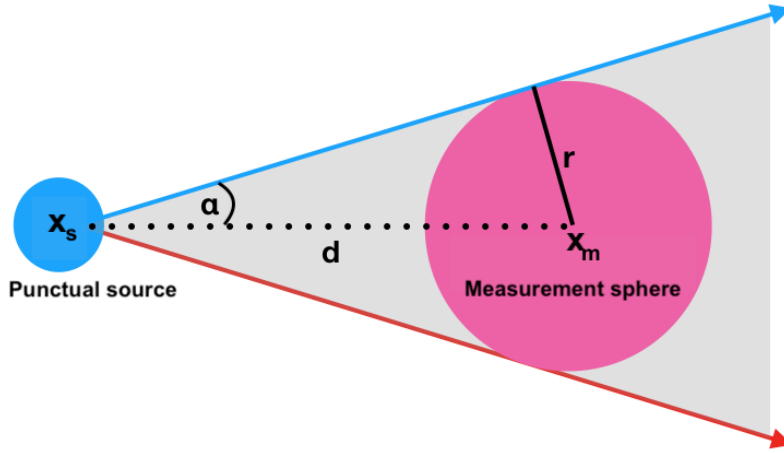


FIGURE 1 Representation of a r -radius measurement sphere centered in x_m , receiving energy from a sound source in x_s .

- A unit sphere uniform sampling (e.g. icosahedre subdivision, Fibonacci's rule⁴, 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 :

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

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

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

Following the figure 1 :

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

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 :

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

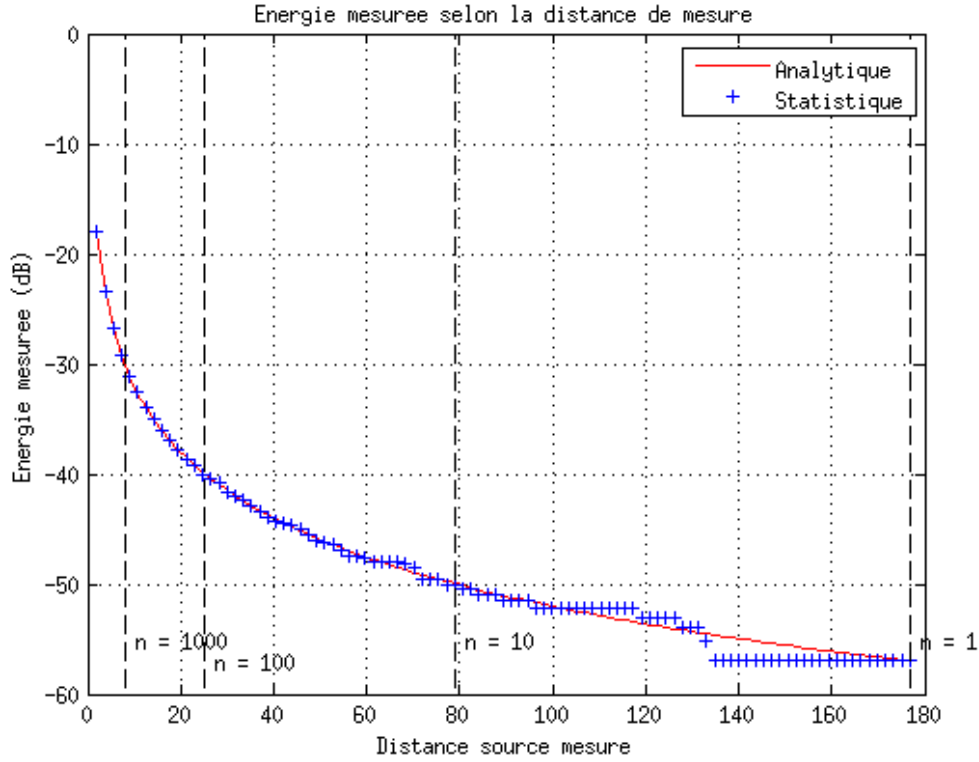


FIGURE 2 Measured energy (dB) in function of distance between x_s and x_m in meter for $r = 0.36\text{m}$ and $N = 10^6$. Blue crosses stand for the statistical measure $f(r) = \frac{n(r)}{N}$ and red ligne the analytic function $f(r) = \frac{\pi r^2}{4\pi d^2}$.

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 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 neglected³. For a room, this condition is reached if :

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

with k the wave number and a the characteristical diameter of the room⁵. 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 \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 :

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

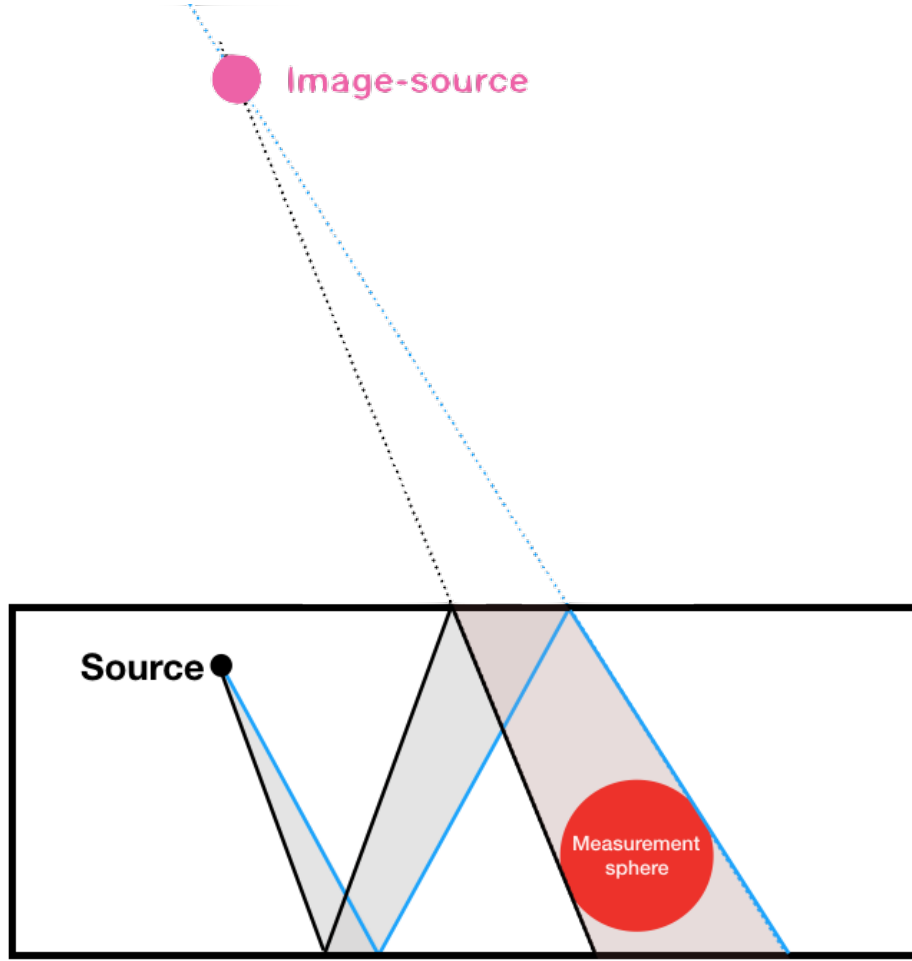


FIGURE 3 Sketch of the creation of an image-source by successive reflections of a ray on the walls of a room.

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* database⁶ 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. 3). Each image-source is then located relatively to the listener and carry an energy according to equation (18). By

¹Harris, 1991

²Dalenback, 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 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 onlign *Odeon* database⁶.

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. 3).

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 algorithm⁹. 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 .

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. 4). 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.

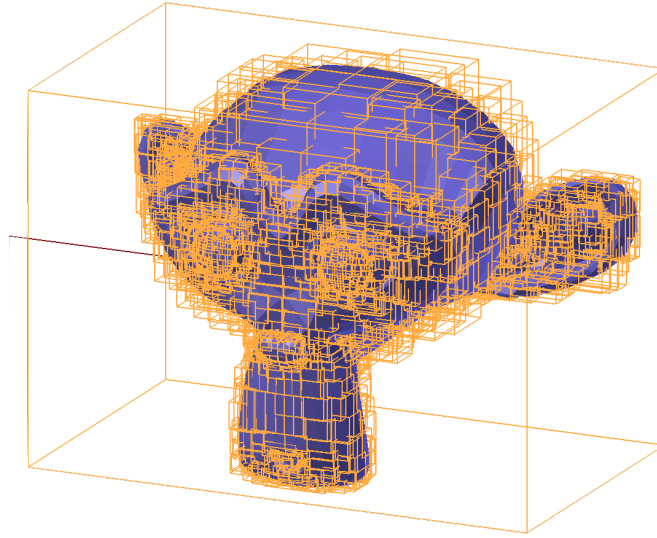


FIGURE 4 Binary tree leaves of an arbitrary mesh from *Blender*.

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.* algorithm¹¹. 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 :

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

which is quasi-linear and computationally faster (see table 2 and fig. 5). Moreover, if binary tree is precomputed, each iteration of ray-tracing stands for :

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

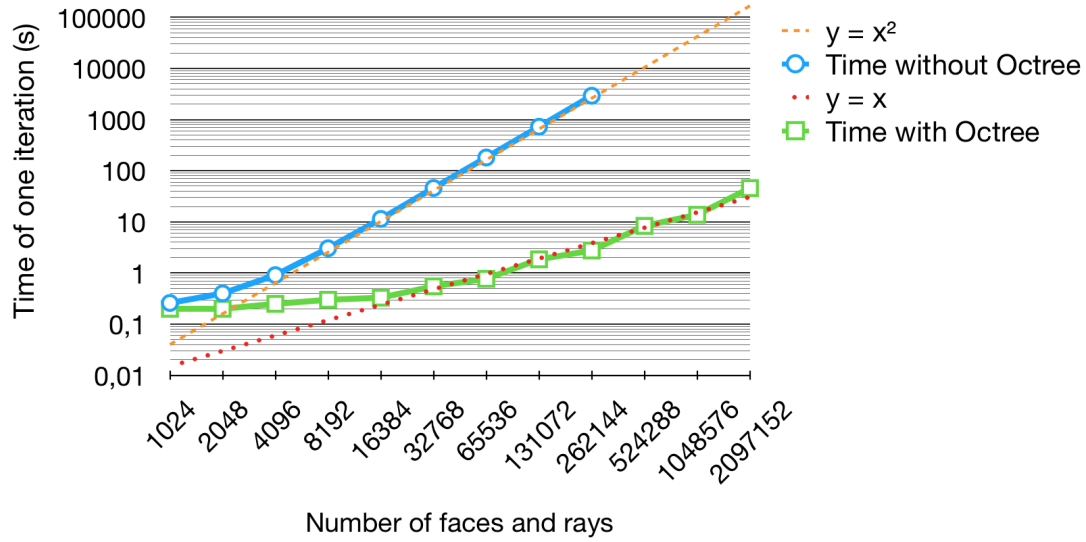


FIGURE 5 Computation time for one iteration of ray-tracing in function of the number of face and rays, such as $N = M$ (log scale). An omnidirectional source is located at the center of a mesh of a unitary tetrahedral.

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 of fig. 5 .

3 | NUMERICAL VALIDATION

3.1 | Shoes box (modele complet)

3.2 | Energy conservation (sphere reflechissante)

4 | APPLICATION TO ORANGE THEATER

5 | CONCLUSION

5.1 | 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-Box¹¹. 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.

6 | VALIDATION

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

6.1 | Quadratic decrease

First, to validate the quadratic decrease of the energy (see section 1) 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

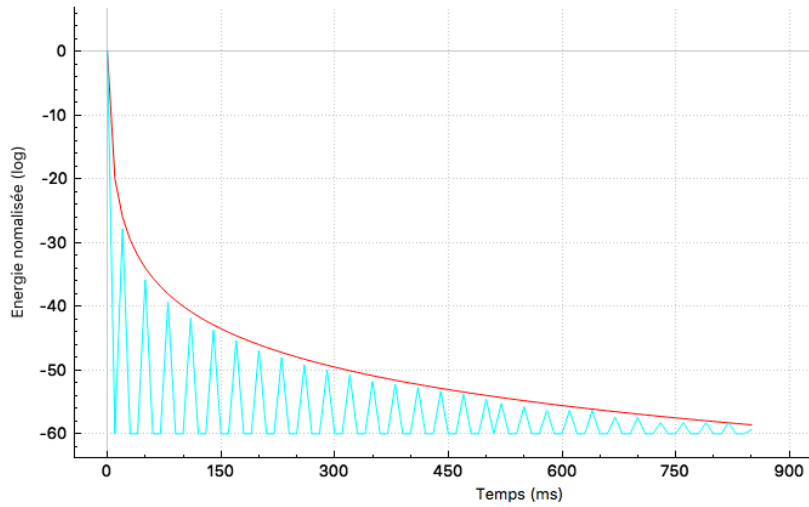


FIGURE 6 RIR in a Fabry-Pérot-like configuration for 3 millions of *rays* (blue) sampled at 100Hz and $f(x) = \frac{2}{x^2}$ (red)

space, the number of *rays* collected stands for the surfaces ratio and the quadratic decrease :

$$\frac{n}{N} = \frac{\int_s dS}{\int_\sigma dS} = \frac{\pi r^2}{4\pi d^2}, \quad (26)$$

with :

- 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).

6.2 | 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

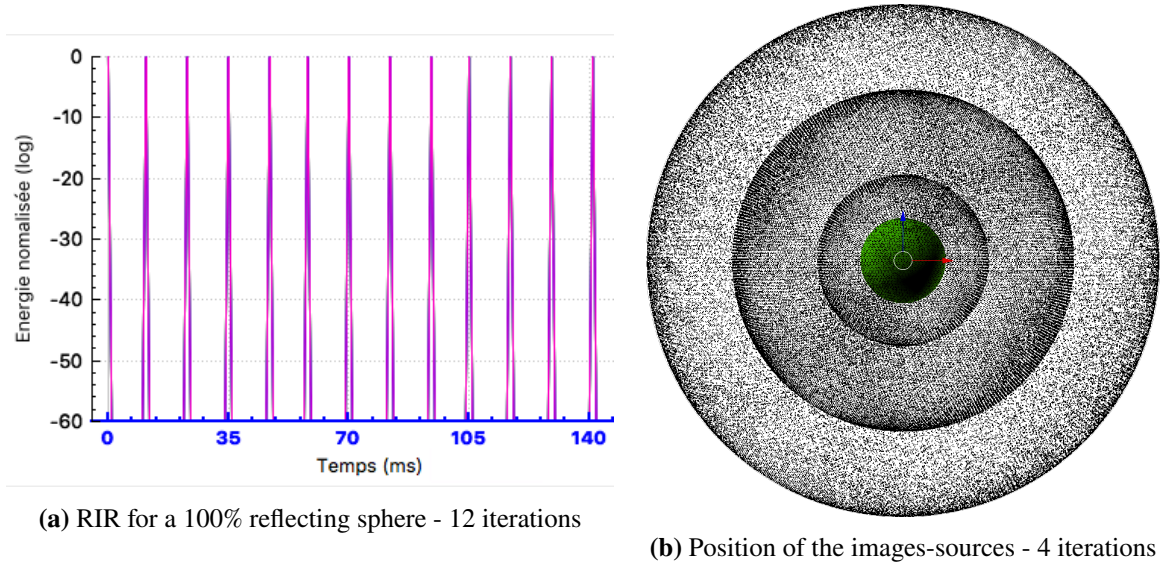


FIGURE 7 100% reflecting sphere

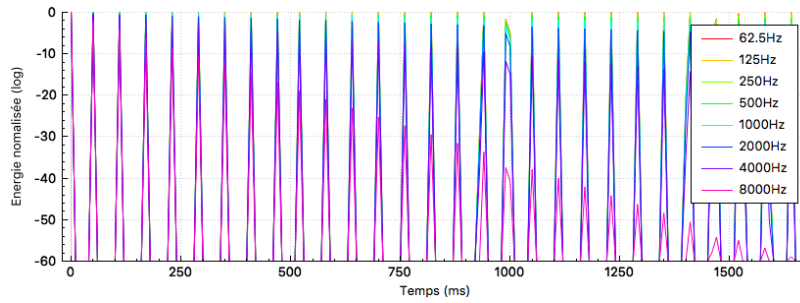


FIGURE 8 RIR for a 100% reflecting sphere with air absorption - 30 iterations

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.

6.3 | 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 formula¹² :

$$P_{is} = i \times D + P_s \times (-1)^i, \quad (27)$$

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.

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 :

$$\epsilon_{rel} = \frac{|E_{exp} - E_{theo}|}{E_{theo}}, \quad (28)$$

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.

$$\epsilon_{\infty} = \frac{|E_{exp} - E_{theo}|}{\max(E_{theo})}. \quad (29)$$

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.

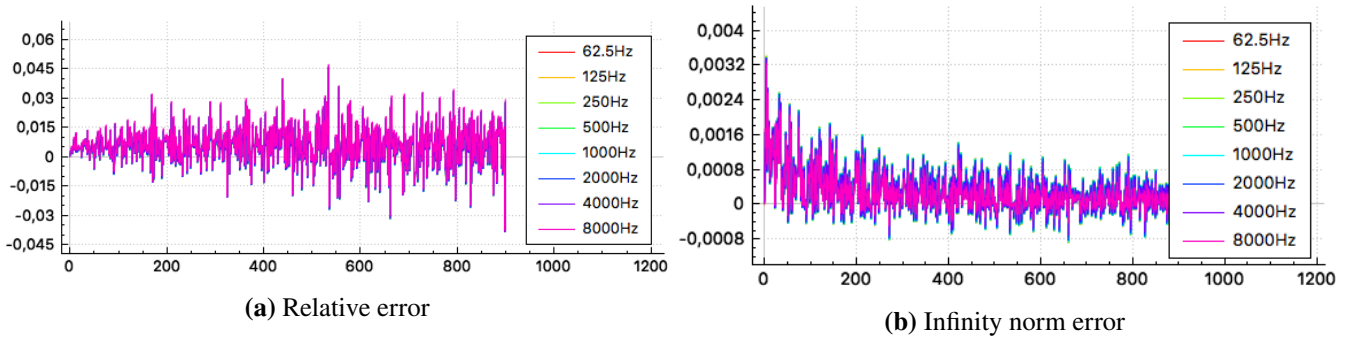


FIGURE 9 Error for each image-sources with walls and air absorption - 1 000 000 rays

7 | DEVELOPED SOFTWARES

The room acoustic tool is available in two forms.

7.1 | Matlab library

First a Matlab library¹³ ...

7.2 | 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++ compiled 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.

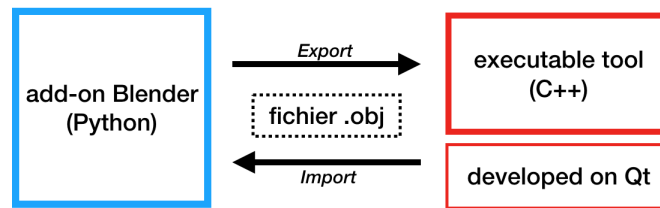


FIGURE 10 Software architecture

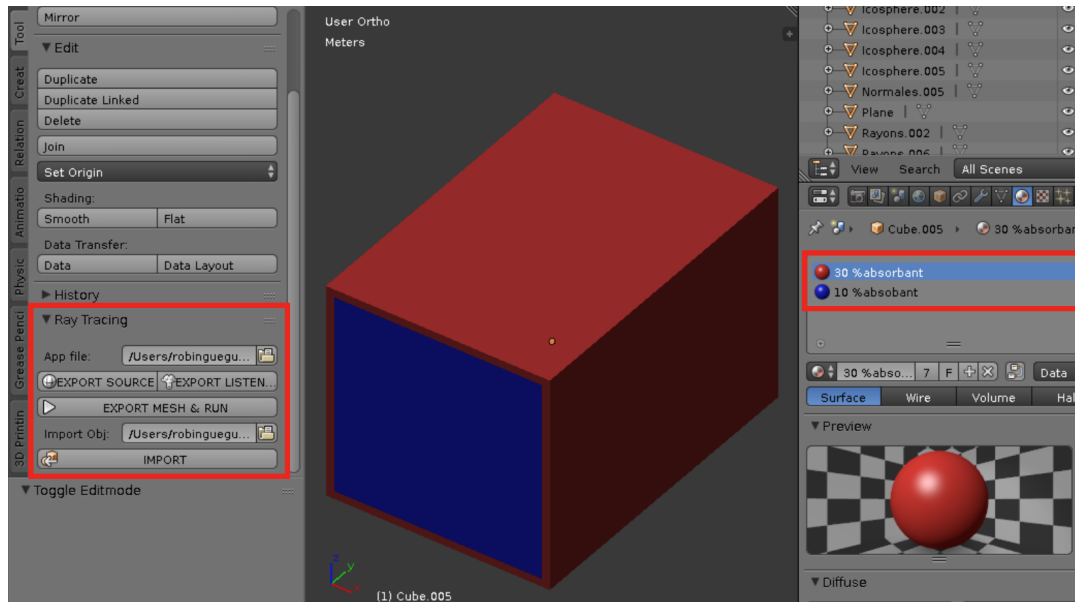


FIGURE 11 Add-on Blender for exporting/importing meshes and running computation tool

8 | APPLICATION TO THE ANTIC THEATRE OF ORANGE

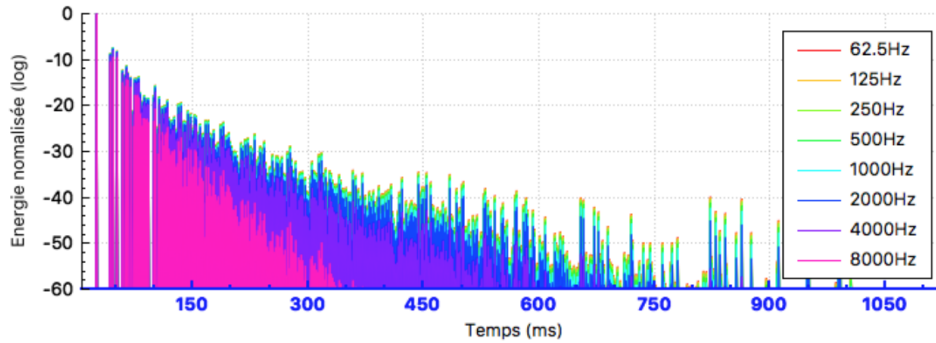


FIGURE 12 Room impulse response of the antic theater of Orange

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 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).

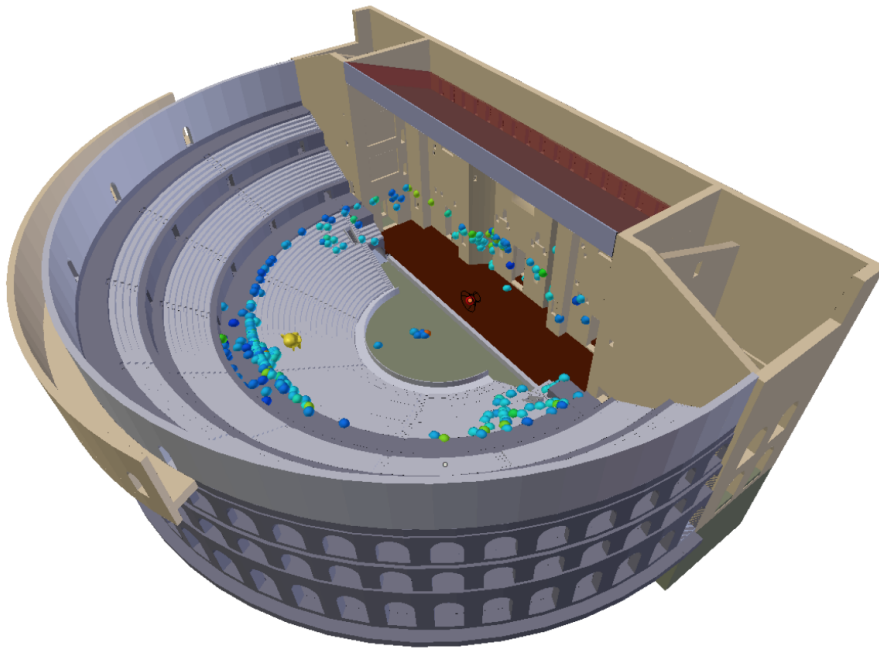


FIGURE 13 Images-sources projected on the theatre of Orange

9 | 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 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 thesis Sorbonne Université 2018.
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-Debain E., Yanga M., Nennig 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. Grennigard 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 Havre)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.