Author/s:
Mirra, G;Mack, M;Pugnale, A

Title:
Aeolus: a Grasshopper plugin for the interactive design and optimisation of acoustic shells

Date:
2023-07-10

Citation:
Mirra, G., Mack, M. & Pugnale, A. (2023). Aeolus: a Grasshopper plugin for the interactive design and optimisation of acoustic shells. Xie, YM (Ed.) Burry, J (Ed.) Lee, TU (Ed.) Ma, J (Ed.) Proceedings of the International Association for Shell and Spatial Structures Annual Symposium 2023:Integration of Design and Fabrication, pp.1785-1797. IASS.

Persistent Link:
https://hdl.handle.net/11343/336084

IASS 2023
July 10–14 Melbourne
Integration of Design and Fabrication

Proceedings of the IASS Annual Symposium 2023
Integration of Design and Fabrication
10–14 July 2023, Melbourne, Australia
Y.M. Xie, J. Burry, T.U. Lee and J. Ma (eds.)

Aeolus: a *Grasshopper* plugin for the interactive design and optimisation of acoustic shells

Gabriele MIRRA, Michael MACK*, Alberto PUGNALE

Faculty of Architecture, Building and Planning, The University of Melbourne
Glyn Davis Building (133), The University of Melbourne, Masson Rd, Parkville VIC 3010, Australia
michael.mack@unimelb.edu.au

Abstract

The design of music venues, such as concert halls and open-air concert stages, requires an integrated approach in which the acoustic response of the space being created is evaluated at every stage of the process to inform its formal development and associated performance. Various software exists to assess acoustic performance: *Odeon* is considered by many as the industry standard standalone software for detailed and accurate acoustic analyses, whereas *Pachyderm* is a plugin for *Grasshopper* (*Rhinoceros 3D*), which provides architects and engineers with more rapid feedback on preliminary design ideas because of its integration with a parametric modelling environment. Although an experienced acoustic designer could rapidly learn to use any of these programs, their usefulness during conceptual design or teaching activities is limited by the computational power and time required to get acoustic performance results. In these instances, testing time is more valuable than accuracy.

This paper introduces *Aeolus*, an acoustic modelling plugin for *Grasshopper*. Unlike other similar *Grasshopper* plugins, *Aeolus* allows users to control the simulation accuracy and can therefore be used to rapidly test the performance of design ideas. *Aeolus* can also easily be interfaced with various *Grasshopper* optimisation plugins. *Aeolus v0.1* was publicly released on 7 March 2023, and will be the focus of a Masterclass offered at the IASS Symposium 2023 in Melbourne, Australia.

Keywords: acoustic design, shell structures, computational morphogenesis, acoustic optimisation, conceptual design

1. Introduction

In architectural design, computational and performance-oriented approaches are no longer restricted to a niche number of large practices, engineering consultants and researchers: CAD software, particularly *Rhinoceros 3D* in recent years, has progressively facilitated the interface of architectural models with performance analysis software and optimisation algorithms. From Luigi Moretti's essays of the 40s and 50s about a future of parametric architecture and performance optimisation [1] to the Building Optimisation Procedure (BOP), developed by SOM Chicago in the 80s [2], software customisation and the integration of performance criteria in the design process have gone a long way [3-5].

For various reasons that are well beyond the scope of this paper, it can be said that *Rhinoceros 3D* has become a reference CAD environment over the past 20 years for many architects and engineers that make use of, or experiment with, computational design workflows. Whether the purpose concerned exploring and optimising complex geometries or automating design processes for fabrication, user-friendly scripting environments proved to be the answer to developing new tools and interfaces between design ideas and performance assessment. *Grasshopper* for *Rhinoceros 3D* is a visual scripting environment released between 2007 and 2008 [6]. Over the last decade, users have determined its success through numerous innovative performance-oriented design applications and by developing

plugins to enhance its native features and functionalities. For example, thanks to *Karamba 3D*, architects and engineers can now perform finite element analyses and optimise structures directly within *Rhinoceros 3D*, without interrupting their creative design workflows because of annoying operations of exporting and reimporting geometric and performance data [7, 8]. With *Kangaroo*, a plugin focused on simulating physical phenomena, designers can perform structural form finding without using physical models [9]. Thanks to *LadyBug*, integrating environmental aspects during conceptual design has become easier than ever [10].

In this panorama, plugins that allow the optimisation of key building performance criteria, such as structural and environmental, or plugins that manage the flow of data between geometric models and fabrication machines, have been developed much faster than other tools to evaluate secondary – or even invisible – performance requirements. Acoustics falls into this category as it is rarely a performance aspect that informs the design of the built environment, with the only exception of music venues. Moreover, the computational power required to perform an accurate – and therefore useful – acoustic analysis is a major hurdle for developing simple plugins that can provide architects and engineers with real-time feedback within a parametric modelling environment.

Aeolus was conceived to address this problem precisely [11]. It was released as a *Grasshopper* plugin in March 2023 to allow designers, particularly architects and students, to rapidly explore and evaluate formal solutions for the design of acoustic shells [12]. According to acoustician experts, such as Beranek [13] and Long [14], an acoustic shell constitutes a temporary or permanent physical device, a surface as a minimum, either integrated into architectural systems or as a standalone object, designed to reflect sound both towards the audience and to the musicians themselves. The integration with industry-standard software and the extreme simplicity of *Aeolus*, which is based on the Image Source Method rather than FEM, are key features to bear in mind while comparing this product with other software to understand its value fully.

After a brief introduction to acoustic modelling strategies, and a description of commonly used current software and plugins in Section 2, the parametric design workflow with *Aeolus* is explained in detailed in Section 3. Given the applicability of *Aeolus* to the conceptual design of various building typologies, Section 4 focuses on potential uses of the tool inspired by real case studies to stimulate the reader's creativity. This paper should be considered a formal introduction and presentation of Aeolus, which will be complemented by a Masterclass at the IASS Symposium 2023 in Melbourne, Australia.

2. Acoustic modelling in parametric design environments

Acoustic modelling utilises the principles of geometric acoustics to simulate the propagation of soundwaves: at any given time t, a point on the wavefront is represented by a vector – an acoustic ray – with magnitude equal to the distance travelled by the sound. The interaction of an acoustic ray with architectural surfaces is modelled using the Snell Law, which states that the angle of reflection must be equal to the angle of incidence. A sound reflection path is computed using basic principles of descriptive geometry by recursively applying the Snell Law: each time the acoustic ray intersects a surface, the reflection angle is computed, and a new ray is cast. The ray casting process aims to either visualise the reflection paths or measure the sound pressure levels (SPLs) at a specific target location.

This process can be implemented through two different methods: "ray tracing" and "image source". The ray tracing method casts a pre-defined number of rays from the sound source in multiple directions [15]. After constructing a certain number of reflection paths, this method computes how many reflected rays intersect with a target surface representing the audience. The accuracy of this method heavily relies on the initial number of rays used, as the distance between two points on the wavefront increases with the square of the wavefront radius. In other words, the sample resolution decreases exponentially with the distance from the sound source. Conversely, the image-source method [16] constructs the reflection paths backwards from specific target points to the sound source. Having specified a reflection order N, i.e., how many times a ray bounces off surfaces before reaching the target, the method tests all possible

combinations of reflection paths, which will be equal to the number of surfaces composing the scene raised to *N*. Therefore, the image source method allows the computation of accurate SPLs at one or multiple target locations, which can be visualised as a 2D map. It is worth mentioning that both ray tracing and image source methods have pros and cons. Therefore, the acoustic analysis uses a combination of both methods to model early and late reflections, respectively.

These methods are currently integrated into plugins for *Grasshopper* to various extents. Excluding *Aeolus*, three other *Grasshopper* plugins are currently available: *Equissons, Snail,* and *Pachyderm*. *Equissons* focuses on auralisation, *Snail* on visualising ray tracing. However, they do not provide a quantitative analysis of acoustic measures. *Pachyderm* is the only plugin that allows users to visualise ray tracing but also implements the image source method to calculate early reflections. Standalone acoustic analysis software, such as *Odeon* and *Catt-Acoustic*, can simulate sound wave properties for calculating acoustic phenomena: for example, scattering and diffusion. Although the accuracy of these commercial standalone packages is higher than those of current *Grasshopper* plugins, they are primarily focused on detailed analysis: the learning curve is not necessarily practical in an architectural design setting, where simple computation and rapid feedback are required. Table 1 summarises the main features of all the plugins currently available for *Grasshopper*. *Odeon* and *CATT-Acoustic* are also included for comparison since they are commonly considered as the industry standards in acoustic engineering.

Table 1. Comparison of *Aeolus* against industry-standard acoustic analysis software and *Grasshopper* plugins.

	Odeon	**CATT-Acoustic**	**Equissions**	**Snail**	**Pachyderm**	**Aeolus**
Plugin for Grasshopper / Rhinoceros 3D			✓	✓	✓	✓
Native integration with parametric modelling			✓	✓	✓	✓
Quantitative analysis of acoustic data (e.g. SPL, RT)	✓	✓			✓	✓
Visualisation of acoustic analysis method	✓	✓	✓	✓	✓	✓
Auralisation	✓	✓	✓		✓	
Control of simulation method	✓	✓			✓	
Control over simulation accuracy	✓	✓				✓

Integrating acoustic modelling tools with parametric design software is essential to being able to design architectural spaces that are informed by acoustic performance. *Grasshopper* for *Rhinoceros 3D* allows the direct association between design parameters and acoustic outcome, letting designers receive acoustic feedback rapidly and interactively on preliminary formal ideas.

As mentioned above, *Pachyderm* is the only *Grasshopper* plugin that provides quantitative acoustic data, relatively accurate acoustic analysis, and visualisation and auralisation functionality. However, this plugin does not allow sufficient control over the acoustic simulation accuracy and speed, which makes it unsuitable for single and multi-objective optimisation algorithms.

Aeolus was developed in 2015 by Gabriele Mirra and Eduardo Pignatelli as part of the deliverables submitted with their master's thesis in Architecture (University of Naples Federico II) [11]. The purpose was to create an acoustic modelling plugin that could interface with multi-objective optimisation algorithms directly within the *Grasshopper* environment. At present, *Aeolus* only implements the image source method to accurately compute SPLs for early reflections. Even though the image source is a computationally expensive method, *Aeolus* allows users to select the level of the simulation accuracy to

produce a fast – but reliable – preliminary acoustic analysis. *Aeolus* can therefore be used in performance-oriented design workflows, where the resolution of the acoustic analysis can be reduced to prioritise the efficiency of optimisation algorithms.

It is worth underlining that *Aeolus* has been used in practice, research and teaching settings for the past eight years, even though it was only made publicly available as a free plugin for *Grasshopper* in March 2023. Regarding research, *Aeolus* was applied to the design and optimisation of a temporary acoustic shell in Acireale, Sicily. This acoustic shell, also known as "ReS", was first conceived in 2012 to be assembled and disassembled from wooden components and its design has then been improved and refined on a yearly basis [11], 17]. *Aeolus* was also used by Pugnale [18] in 2018 to study the acoustic performance of some of Frei Otto's most known tensile structures. In teaching and learning, *Aeolus* has been used for the past four years as the parametric acoustic analysis software in an architecture master's design studio at The University of Melbourne: the focus has been the acoustic design and optimisation of a music concert hall and school.

3. The design workflow with *Aeolus*

The section describes how *Aeolus* works and explains a typical acoustic analysis workflow with this plugin, from constructing the acoustic scene to visualising the simulation output as a 2D SPL map (Figure 1). The scene considered is simple to focus on the different stages of the workflow: it consists of a sound source, a curved surface acting as an acoustic reflector and an array of target/receiver locations evenly distributed over a hypothetical audience area (Figure 1, at the top).

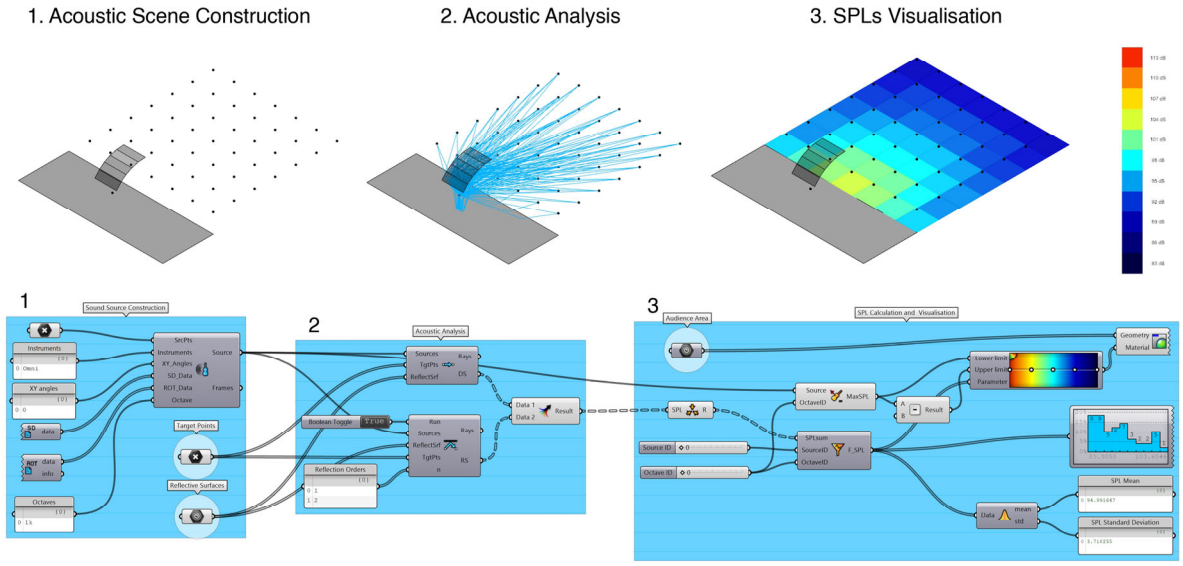

Figure 1. Overview of the design and analysis workflow with *Aeolus*, from constructing the acoustic scene to visualising the simulation output as an SPL map.

3.1. Constructing an acoustic scene

An acoustic scene configuration comprises, as a minimum, three geometric entities: a point, which represents the sound source location; a surface, which acts as an acoustic reflector; and a second point, which identifies the target or receiver location. Such a scene can be analysed with geometric acoustics to construct the sound reflection paths from a sound source to a target location. In acoustic modelling software, the three entities of a minimal acoustic scene can be assigned with physical properties to produce an accurate simulation. For instance, a sound source can be characterised by a sound directivity map, which assigns different SPLs to emission angles according to the instrument or loudspeaker

typology, whereas an acoustic reflector can have sound absorption coefficients which depend on the reflector material. *Aeolus* assumes that all surfaces are perfectly reflective, meaning they do not absorb sound. However, the plugin allows assigning a sound source with the directivity map of up to 15 instruments, which enables a more realistic analysis should the application require it. The following subsections describe how to construct the acoustic scene illustrated in Figure 1 using *Aeolus*: three steps are necessary, one for each of the entities listed above.

3.1.1. Defining the sound source

In *Aeolus*, a sound source object is constructed through a dedicated component by specifying: (1) a point location; (2) the instrument typology; (3) the sound source orientation in degrees; and (4) the octave band, which may include one or more of the seven reference values, i.e., 125, 250, 500, 1k, 4k, 8k, and 16k Hz. The sound directivity and orientation data of the supported instruments can be inputted through the components 'SD Database' and 'ROT Database', respectively. These two components import and perform the formatting of text files that are stored in the *Aeolus* installation folder. Such files reproduce information from the CATT-Acoustic sound directivity database and can be updated in future releases to include additional instruments.

Generally, a preliminary acoustic analysis would not require specifying instrument typology and the corresponding sound directivity values. An omnidirectional sound source can be used in *Aeolus* to perform an analysis that does not consider sound directivity. In this case, the specification of sound source orientation and octave band does not affect the simulation output.

3.1.2. Defining the acoustic reflectors

In *Aeolus*, acoustic reflectors are modelled as flat surfaces because the image source method requires a finite number of planes – one for each surface – to pre-compute all the possible reflection paths. Therefore, to analyse curved surfaces, these must be approximated by flat elements (Figure 1, stage 1).

Although there are no limitations to the number of surfaces that *Aeolus* can process, increasing this number can significantly affect the computational cost of the simulation. A trade-off between geometry approximation and simulation accuracy is required to keep the computational cost within manageable timeframes. It is worth mentioning that, as Section 3.4.1. will describe later, the approximation of a curved surface with flat elements does not overly affect the quality of the simulation. In some cases, a surface approximated by a few flat elements can perform better than a more refined one, thus suggesting alternative design solutions that might also be easier to build.

3.1.3. Defining the targets

In *Aeolus*, a target identifies the end point of sound reflection paths and can be considered as a listener's location in the audience area. The plugin accepts the specification of one or multiple targets to perform an acoustic simulation. If multiple targets are used, arranging them on an evenly spaced grid is preferable so that a map can be generated to visualise the distribution of SPLs.

3.2. Performing an acoustic analysis

Once the acoustic scene is defined, the three geometric entities – sound source, reflectors, and targets – are fed to the two *Aeolus* analysis components, 'Direct Sound' and 'Image Source', to construct the direct and reflected acoustic rays and calculate the SPLs.

The 'Direct Sound' component traces the direct sound paths, that is straight lines connecting the sound source with the target points. The paths that intersect any reflective surfaces in the scene are discarded. The component also computes the SPLs associated with the direct sound paths using the path length and the directivity values of the sound source object. The 'Image Source' component performs the main acoustic simulation. Given an acoustic scene and a reflection order N, this component tests all the reflected sound path combinations and keeps only those that connect the sound source with the targets

after *N* intersections with the acoustic reflectors. This component also computes the SPLs using the distance travelled by the reflected sound and the directivity values.

Upon completion of the analysis, these two components produce the direct and reflected sound paths, which can be visualised in the *Rhinoceros 3D* viewport as lines and polylines, respectively (Figure 1, stage 2). By visually inspecting the sound paths, the user can preliminarily assess the performance of the acoustic reflectors. Alternatively, the user can perform a more detailed analysis based on the SPLs computed from the direct and reflected sound paths. *Aeolus* allows users to combine and visualise the SPLs by following the procedure described below.

3.3. Visualising Sound Pressure Levels

Aeolus provides a set of utility components to combine, filter and display the SPLs obtained through the direct and reflected sound analysis. The SPLs are combined using the 'SPL sum' component, which performs a logarithmic sum of the direct and reflected SPLs for every target. The output data structure has the following indices: frequency id, sound source id, and target id. These indices can be used to filter the SPLs by using either the standard data tree functionalities provided in *Grasshopper* or the *Aeolus* 'SPL filter' component.

An SPL map can be constructed from the filtered SPLs by applying a gradient of colours to the cells surrounding the targets (Figure 1, stage 3). *Aeolus* provides a utility component to extract the maximum SPL characterising a sound source object for a specific octave band, which can be used to set the upper bound of the gradient domain. Another component computes the standard deviation and the mean of the SPLs to capture their distribution over the audience area. These values can be used as objective functions for acoustic optimisation applications (see Section 3.5).

A typical use case of the visualisation workflow described above involves selecting a sound source id to extract the SPLs only for that specific sound source and comparing the results with the SPLs corresponding to the other sound sources used in the simulation. Figure 2 shows the SPL maps produced by switching the index of the sound source after completing the simulation of an acoustic scene that included two sound sources placed at the centre and one side of the acoustic reflector. The comparison reveals that the acoustic reflector evenly distributes sound only for the central sound source.

Alternatively, if the user has assigned the sound source object with an instrument and multiple octave bands at the beginning of the simulation, they can switch the frequency id to compare the SPLs for different octave bands. Figure 3 (at the bottom) illustrates the simulation outputs for an acoustic scene in which the sound source is assigned with the directivity values of a violin. The analysis focuses on the 1kHz-8kHz octave band range. By changing the frequency id, the user can visualise in real-time the SPL map associated with a specific sound directivity map (Figure 3 at the top).

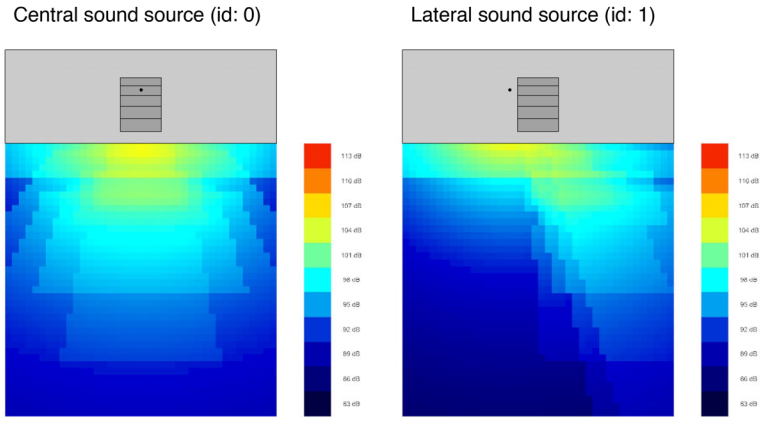

Figure 2. SPL maps produced by switching the sound source id after analysing an acoustic scene with two different sound source locations.

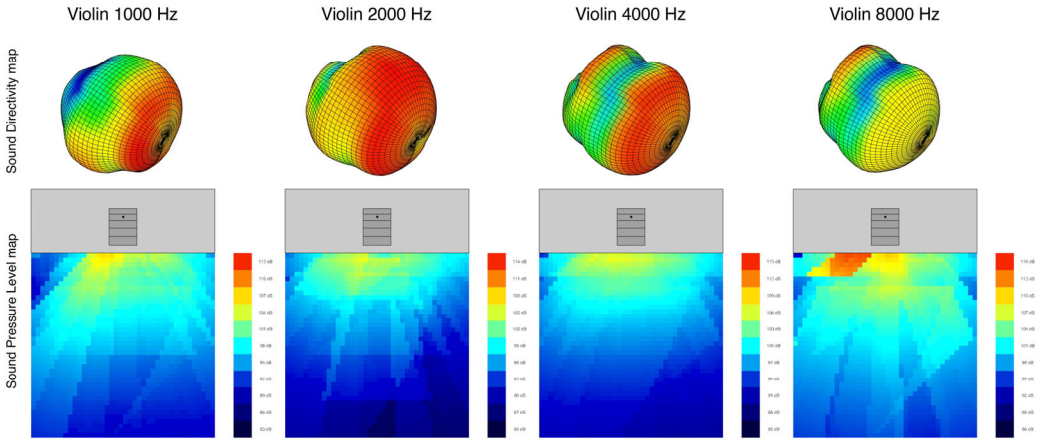

Figure 3. Simulation of an acoustic scene using sound source directivity. At the top: Sound directivity maps of a violin for four different octave bands; at the bottom: SPL maps produced by switching the frequency id after analysing an acoustic scene with a violin sound source for four different octave bands.

3.4. Adjusting the simulation settings

One of the main advantages of *Aeolus* is the possibility of fine-tuning the simulation settings to produce analyses with different degrees of accuracy. This is an essential feature of *Aeolus* because acoustic analyses are computationally expensive, limiting the interoperability of other plugins within parametric design software. This section describes the settings that reduce the computational cost of acoustic analysis in *Aeolus* and the effect these settings have on the simulation output. We do not provide a breakdown of the computational cost for each setting because this information depends on the computer hardware used. We instead compare the simulation outputs through the SPL maps.

3.4.1. Acoustic scene complexity

The complexity of the acoustic scene has a significant impact on the computational cost of the simulation. Increasing the number of acoustic reflectors, sound sources, and targets could provide very accurate information on the performance of a specific design but could also make exploring different formal configurations impractical.

Therefore, the number of acoustic reflectors is the first setting that can be adjusted to speed up the simulation. As mentioned at the beginning of this section, *Aeolus* require all geometries to be provided as flat elements. If the acoustic reflector/s are modelled as curved surfaces, these must be tessellated before running the simulation. The number of individual faces produced through the tessellation will be the actual number of acoustic reflectors used in the simulation. Figure 4 shows different subdivisions of the curved reflector considered in the previous analyses presented in the paper, and the effects these have on the SPL map. Since the reflector was described by a single-curvature surface, the tessellation only controlled the number of subdivisions along the surface V direction, which produced equally sized flat elements. The comparison of the SPL maps reveals that, for this particular geometry, a number of subdivisions of 25 is sufficient to simulate the performance of the original curved reflector accurately. In fact, a higher number of subdivisions did not considerably affect the SPL map, which is characterised by a trapezoidal area with a uniform sound distribution and almost no reflections at the final rows of the audience area. Conversely, tessellations produced from 5 and 10 subdivisions could spread sound to the entire audience area: though these do not capture the shadowing zone at the final rows that characterises the SPL maps obtained from a high number of subdivisions, and only approximate the trapezoidal shape of the strong reflections at the centre. It is worth mentioning that, although less accurate, the analysis suggests that a faceted reflector can be a better design choice since it can reflect sound more evenly.

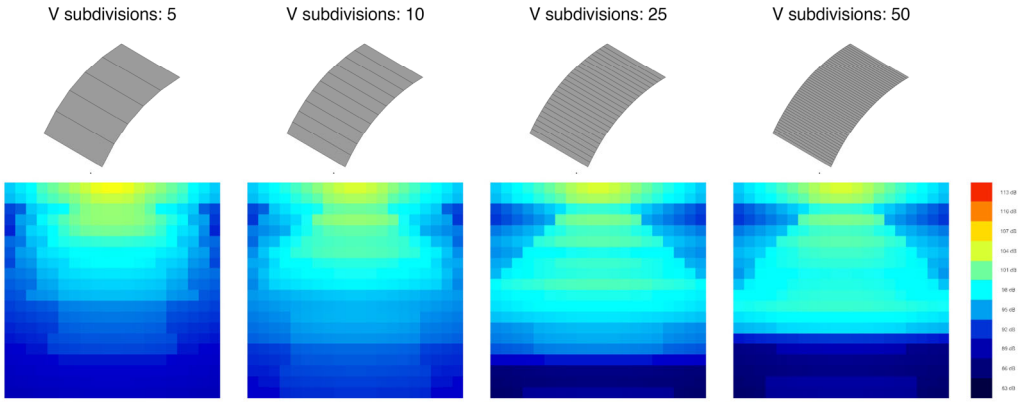

Figure 4. SPL maps for a single-curved reflector tessellated with an increasing number of flat elements.

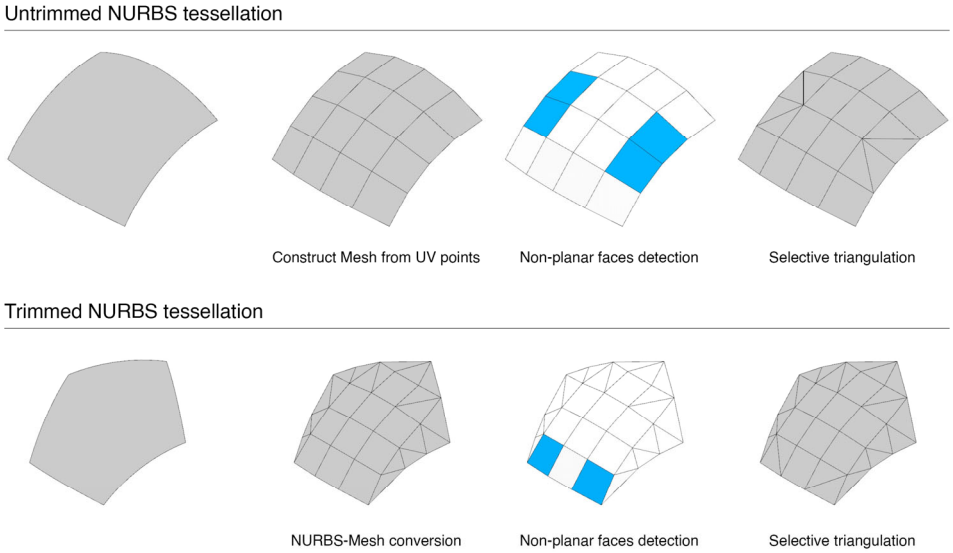

Figure 5. Stages for tessellating untrimmed and trimmed NURBS into a set of planar elements.

A tessellation based on UV subdivisions can also be applied for untrimmed doubly curved surfaces (Figure 5, at the top) by specifying the number of subdivisions for both U and V directions. However, if the surface is not ruled or extracted from a rotational surface, such as a sphere, the tessellation will not produce flat elements, and a planarity analysis is required to detect and triangulate non-planar faces.

Conversely, if the reflector is described by a trimmed surface, e.g., a NURBS with more than four edges, the tessellation cannot be performed through UV subdivisions. The user can convert the surface into a mesh object and control its resolution – i.e., the number of faces – by specifying the min and max length of the mesh edges. A planarity analysis and selective triangulation are also required to guarantee that all the faces are planar and suitable for the acoustic simulation (Figure 5, at the bottom).

Regardless of the input geometry and tessellation strategy, the selection of an appropriate number of faces should be guided by the purpose of the simulation. A low-resolution tessellation might be sufficient in the early exploratory stages of the design process, whereas a more detailed one can be useful to evaluate the performance of the final design accurately.

Another setting that affects the computational cost of the simulation is the number of targets. This parameter influences the resolution of the SPL map and can be controlled through the specification of the grid cell size. Figure 6 compares the SPL maps produced by analysing the same acoustic scene with a different number of targets. The maps are constructed with increasingly smaller grid cells, from 3 m to 0.5 m, in which each cell roughly corresponds to the footprint of a sitting area. The comparison shows

that as the map resolution increases, geometric patterns appear. These patterns are due to the reflections of the 5 panels composing the acoustic reflector and provide information about the contribution of each element to the SPL distribution.

In many cases, measuring the SPL for every point of the audience area might be unnecessary. If the objective of the simulation is assessing the overall distribution of SPLs, a low-resolution grid might be sufficient (see Section 3.5). Moreover, since a target identifies the listener position, the user may select only the actual seats as input for the simulation and discard irrelevant points from the audience area.

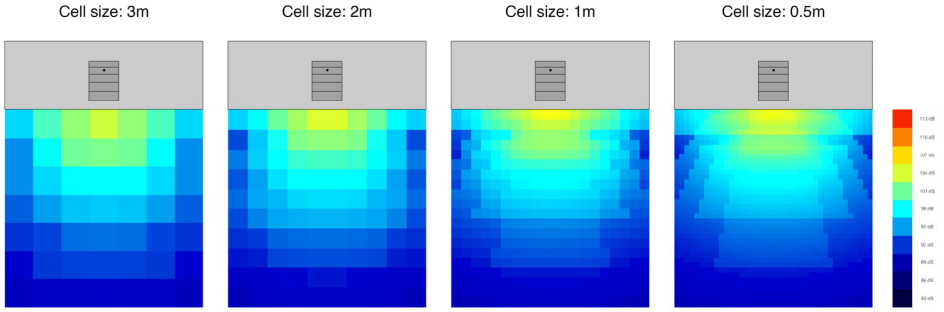

Figure 6. SPL maps at increasingly higher resolutions obtained by specifying a different cell size.

3.4.1. Acoustic analysis accuracy

In *Aeolus*, the accuracy of the simulation mainly depends on the reflection order, which affects the extension of the time domain in which sound waves propagate. A reflection order of 1 will only consider sound paths that arrive at the target location after one intersection with the acoustic reflectors. These paths are generally shorter than those reaching the target after multiple intersections with the reflective surfaces, and therefore do not provide much information about the SPLs associated with late reflections. Although a high reflection order N is required to produce a good estimate of the SPL in the time domain, and accurately compute important acoustic parameters such as sound clarity, definition and reverberation time (RT), the computational cost of the simulation grows exponentially with N. *Aeolus* can compute any number of reflection orders, but due to this limitation, a number higher than 4 is generally unmanageable even for simple acoustic scenes. Despite that, the information provided with an SPL map generated from up to 4 reflections can still be considered sufficient to evaluate a design option at the early design stages.

Figure 7 shows the effects of the reflection order N on the SPL map for two settings: a semi-reverberant field in which the acoustic scene comprises a curved reflector tessellated by 5 flat elements (Figure 7, at the top), and a reverberant field in which the acoustic reflector is surrounded by a rectangular box (Figure 7, at the bottom). Regarding SPL distribution, a reflection order of 2 returned a sufficiently detailed analysis for both settings. In the semi-reverberant field settings, *Aeolus* found no valid reflections for orders 3 and 4. Due to the absence of reflectors at the sides of the sound source, none of the acoustic rays could bounce more than twice before reaching the targets. Conversely, in the reverberant field settings, the reflectors constructed from the faces of the rectangular box always produced valid reflections. Nevertheless, the SPL distribution is almost identical for reflection orders 3 and 4, suggesting that, for this setting, a more appropriate analysis should have considered other acoustic parameters, such as RT, to determine the acoustic performance of the architectural volume. Although *Aeolus* does not currently support RT calculation, the analysis of the early reflections may still prove useful to determine the overall performance of an acoustic scene in a reverberant-field setting. In fact, strong early reflections correlate with good sound clarity and definition [11], and thus can effectively support a performance-oriented approach at the conceptual stages of the design process.

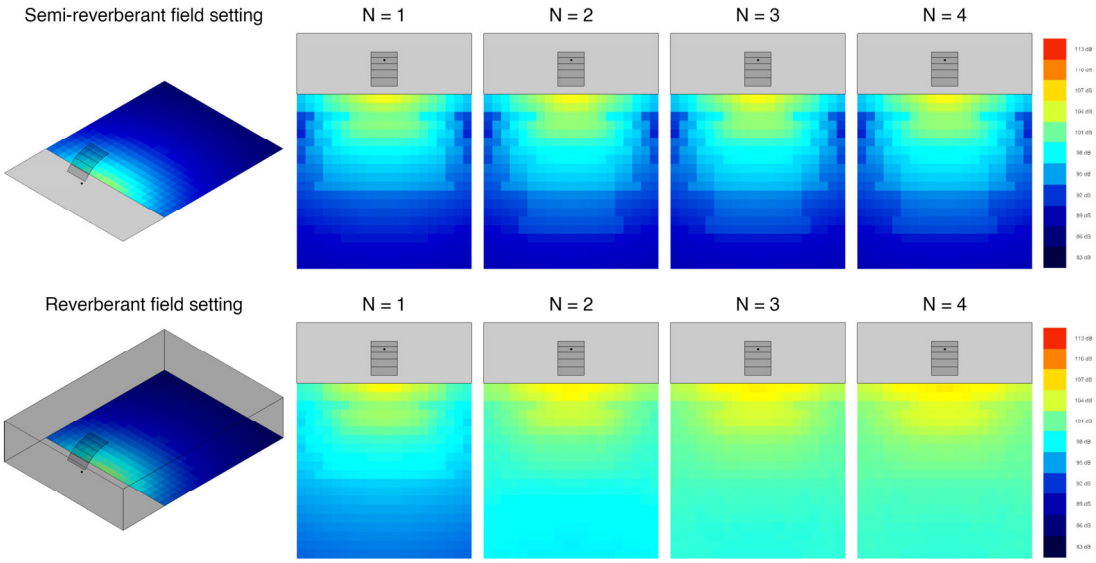

Figure 7. SPL maps for different reflection orders N. At the top, analysis of a single curved reflector (semi-reverberant field); at the bottom, analysis of a curved reflector inside a rectangular box (reverberant field).

3.5. Integration with optimisation plugins

As described in the previous sections, *Aeolus* allows performing more or less accurate analyses by fine-tuning the simulation settings, providing full control over the calculation time. With the appropriate settings, a user can rapidly test the acoustic performance of various design options within the *Grasshopper* environment. Alternatively, they can perform an optimisation process in *Grasshopper* using either the pre-built Genetic Algorithm (GA) component *Galapagos* or third-party plugins, such as *Octopus*.

Figure 8 illustrates a basic acoustic optimisation performed with *Galapagos*. The objective function was minimising the standard deviation of the SPLs distribution on the audience area. Three design variables were selected to control the z coordinate of the three control points defining the base profile of the curved reflector used for all the analyses described in this section (Figure 8, on the left). The acoustic reflector was tessellated with 5 flat elements, while a cell size of 2 m was used for the audience grid. The simulation considered reflections up to the second order. This setting kept the computational cost of each simulation below 100 ms.

The convergence graph in Figure 8 demonstrates that *Galapagos* could retrieve design options producing SPL maps with an increasingly lower standard deviation, which reduced from 3.67 (original design) to 2.36 (after 24 generations). The solution extracted from the 24th generation can be considered an improvement of the original design as the SPL is more uniform across the entire audience, meaning that the optimised design can reflect sound – for a specific sound source location – up to the last row of targets (Figure 8, at the bottom). Figure 8, at the right, compares the SPL maps of the original and optimal design obtained by increasing the resolution of surface tessellation and audience grid after the optimisation process. The optimal design still outperforms the original one in terms of SPL distribution, demonstrating that performing an analysis with *Aeolus* at a coarse resolution is a valid strategy for implementing an effective acoustic optimisation process.

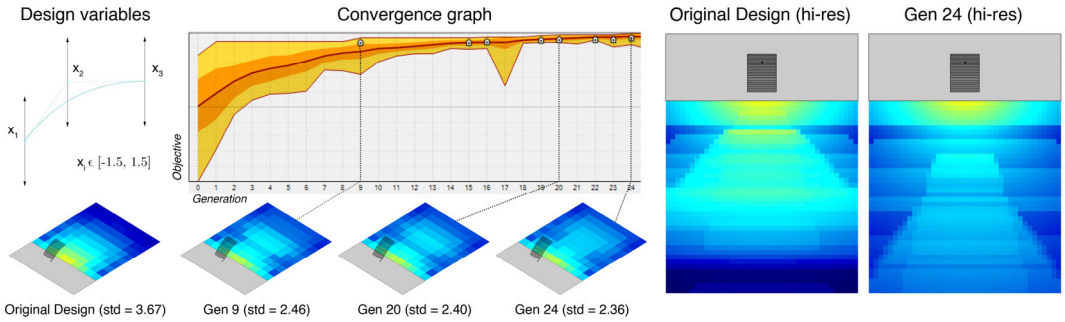

Figure 8. Example of an acoustic optimisation using *Aeolus* with the Genetic Algorithm *Galapagos* for *Grasshopper*; On the left, a representation of the design variables and convergence graph with selected design solutions; on the right, comparison between a high-resolution analysis of the original and optimised design.

4. Potential applications of *Aeolus*

The integration of *Aeolus* with architectural design and parametric modelling tools allows designers to use this plugin beyond applications that are specifically focused on acoustic performance. A brief survey of non-acoustic-oriented projects revealed that many architectural and structural forms could easily be recontextualised to perform as acoustic shells. We identified at least four different categories of architectural forms, which, even though unintendedly designed to be used as acoustic devices, impact the acoustic quality of the space they sit in. Figure 9 illustrates four example projects, and two abstracted variations of each are accompanied by an acoustic analysis to showcase the wider architectural applications of *Aeolus*. For the analyses, a single omnidirectional sound source was used at the centre of the design space, and the cell spacing for SPL analysis was set at 1m.

Facades [F] and openings in building envelopes are generally designed formally as invitational or gathering spaces providing opportunities for events. Naturally, this typology lends itself towards similarities with conventional acoustic performance shells encouraging mono-directional sound projection. In the examples, F1 mimics a canopy-like structure which can be adjusted to focus reflections intimately towards the opening or reflect sound away. [F2] proposes control over the lateral focusing of reflections, encouraging a linear approach towards the opening.

Gallery [G] spaces range from arcades to covered walkways, where sound is reflected from the ceiling treatment and perimeter walls and projected across the length of the primary circulation. The faceted example in [G1] gives a scattering-like effect at a larger scale resulting in an unfocused and lively environment. [G2] simulates a simple canopy or arched structure serving to reflect sound in the bi-lateral directions.

Canopies [C], promenades and marketplaces feature areas that are covered but open laterally for circulation. *Aeolus* can be used in these cases to understand the impact of the ceiling design and its performance in the multi-directional distribution or dispersion of sound in and away from the covered area. The example in [C1] produces similar scattering results to [G1] due to the asymmetrical ceiling treatment, and the density or regularity of the facets could be adjusted to control the level of scattering. The alternating wave canopy structure in [C2] creates clear horizontal banding of louder regions across the site. This analysis could serve to situate programs or functions across the site.

Pavilions [P] can be architecturally programmed for many purposes that vary in acoustic requirements. Using *Aeolus* to study the acoustic quality of pavilions may be useful to qualify the design for specific programs. While the faceted [P1] is larger by footprint, the active acoustic area is relatively smaller compared to the concave shell structure of [P2] which encourages a wider dispersion of acoustic reflections outside the footprint of the pavilion. Interestingly, both these examples also demonstrate a limitation of geometric acoustics, shown by the acoustic shadows generated by the pavilion elements that touch the ground.

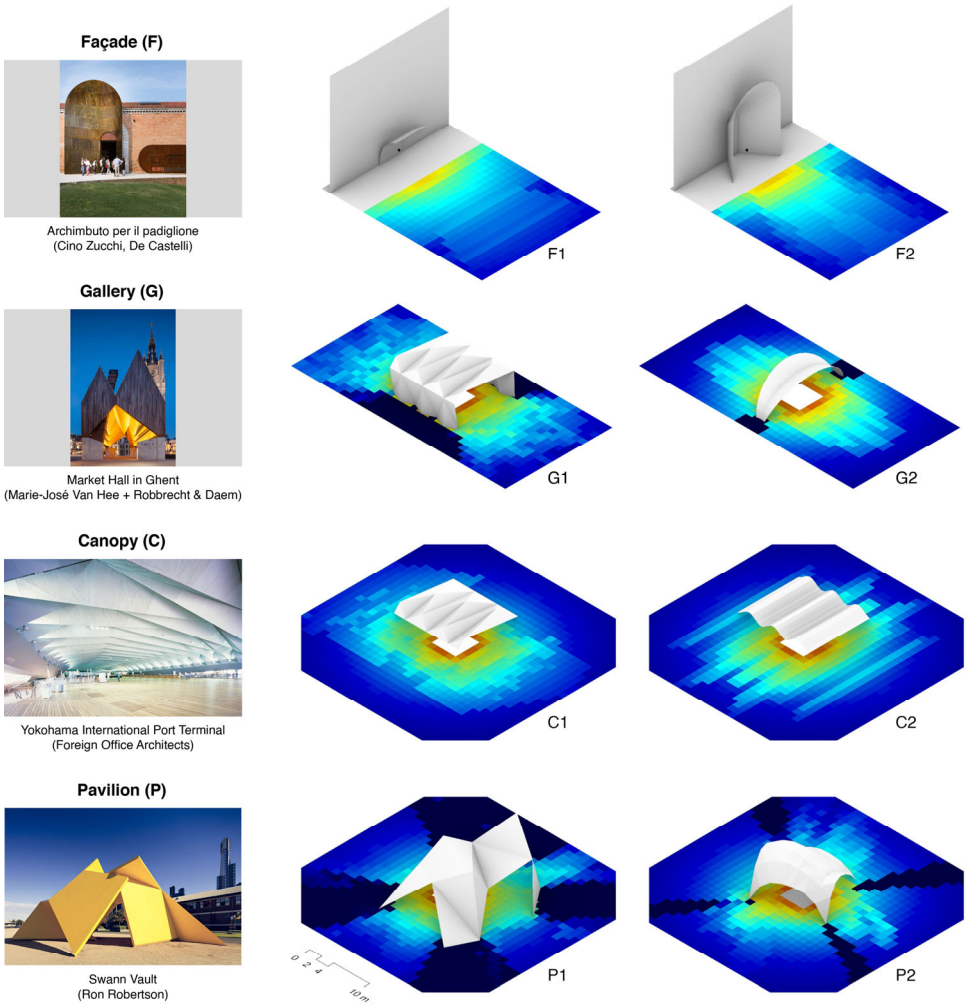

Figure 9. Example of potential applications of *Aeolus* with various architectural typologies. On the left, precedent projects illustrate each typology; on the right, comparison of SPL graphs of abstracted models.

5. Conclusions

This paper describes the development of *Aeolus* and its functionalities as an acoustic modelling software for the design, analysis and optimisation of acoustic shells. Its integration with *Grasshopper* for *Rhinoceros 3D*, i.e. an architectural parametric modelling software and associated optimisation plugins, makes it an effective tool for and performance-oriented design workflows, particularly at the conceptual design stages. Control over the degree of simulation accuracy, number of sound sources, and directivity allows the user to adapt the plugin precision level to different stages of the design process.

Like other available acoustic analysis tools, there are several shortcomings of *Aeolus*. The image source method requires surfaces to be modelled as flat, planar reflectors. This degree of approximation will always result in a trade-off between simulation accuracy, computational cost, and time. Particularly in a reverberant field setting, the reliance on the image source method also limits the accuracy of the SPL calculation and the lack of calculation for RT, though early reflections are suitable for measures of clarity and definition. Despite this, *Aeolus* can perform with a high enough accuracy and provide reliable analysis for preliminary stages in concept design, making it a suitable tool for architectural practice, research, and teaching. Future software development will incorporate additional algorithms to compute late reflections and construct a complete impulse-response graph.

Acknowledgements

We acknowledge Eduardo Pignatelli for contributing to the creation and development of the *Aeolus* plugin for *Grasshopper*. We also acknowledge Professor Sergio Pone, who supervised Gabriele Mirra's and Eduardo Pignatelli's master's thesis in 2015, through which *Aeolus* was developed. Professor Sergio Pone also financially supported the first practical application of *Aeolus*.

References

[1] L. Moretti, "Ricerca matematica in architettura e urbanistica", *Moebius*, no. 1, pp. 30-53, 1971. Also in: F. Bucci and M. Mulazzani, *Luigi Moretti: Opera e scritti*, Electa, 2000.

[2] N. Adams, *Skidmore, Owings & Merrill. SOM since 1936*, Electa, pp. 34-36, 2007. The use of computers within SOM has also been discussed at a conference entitled "Digital Design at SOM: The Past, the Present and the Future", accessible at the web address: vimeo.com/42786059 (accessed 13 April 2023).

[3] R. Aish and R. Woodbury, "Multi-level Interaction in Parametric Design", in: A. Butz, B. Fisher, A. Krüger, P. Olivier (eds.) *Smart Graphics. SG 2005. Lecture Notes in Computer Science*, vol. 3638, 2005.

[4] K. Besserud, N. Katz, A. Beghini, "Structural Emergence: Architectural and Structural Design Collaboration at SOM", *Architectural Design*, vol. 83:2, pp. 48-55, March/April 2013.

[5] G. Celani, C.E. Verzola Vaz, "CAD Scripting and Visual Programming Languages for Implementing Computational Design Concepts: A Comparison from a Pedagogical Point of View", *International Journal of Architectural Computing*, vol. 10:1, pp. 121-137, 2012.

[6] www.grasshopper3d.com/ (accessed 13 April 2023).

[7] C. Preisinger and M. Heimrath, "Karamba—A Toolkit for Parametric Structural Design", *Structural Engineering International*, vol. 24:2, pp. 217-221, 2014.

[8] C. Preisinger, "Linking Structure and Parametric Geometry", *Architectural Design*, vol. 83:2, pp. 110-113, March/April 2013.

[9] D. Piker, "Kangaroo: Form Finding with Computational Physics", *Architectural Design*, vol. 83:2, pp. 132-135, March/April 2013.

[10] M. Sadeghipour Roudsari, M. Pak, "Ladybug: A Parametric Environmental Plugin for Grasshopper to Help Designers Create an Environmentally-Conscious Design", in *Proceedings of the 13th International conference of Building Performance Simulation Association*, Chambery, France, 25-28 August, 2013.

[11] G. Mirra, E. Pignatelli and S. Pone, "Computational Morphogenesis and construction of an acoustic shell for outdoor chamber music", in *Proceedings of the IASS Annual Symposium 2016 "Spatial Structures in the 21st Century"*, Tokyo, Japan, 26-30 September, 2016.

[12] G. Mirra, *Aeolus*, www.food4rhino.com/en/app/aeolus (accessed 13 April 2023).

[13] L. L. Beranek, *Concert Halls and Opera Houses: Music, Acoustics, and Architecture (2nd Edition)*, Springer, 2003.

[14] M. Long, *Architectural Acoustics (2nd Edition)*, Elsevier Science & Technology, 2014.

[15] E. Haines, *Essential Ray Tracing Algorithms*, in A.S. Glassner (ed.), "An Introduction to Ray Tracing", Morgan Kaufmann Publishers, 1989.

[16] V. Martin and T. Guignard, Justification of the Image Sources in Ray-tracing Method, ICSV Congress 2005.

[17] G. Mirra, E. Pignatelli and S. Di Rosario. "An automated design methodology for acoustic shells in outdoor concerts", in *Proceedings of the Euronoise 2018 conference*, Hersonissos, Crete, 27-31 May, 2018.

[18] A. Pugnale, *Integrating Form, Structure and Acoustics: A Computational Reinterpretation of Frei Otto's Design Method and Vision*, "Journal of the International Association for Shell and Spatial Structures", vol. 59: 195, pp. 75- 86, 2018.

[19] A. van der Harten, "Pachyderm Acoustical Simulation: Towards Open-Source Sound Analysis" *Architectural Design*, vol. 83, pp. 138–139, 2013.

[20] A. van der Harten, Pachyderm Acoustic Simulation, www.food4rhino.com/en/app/pachyderm-acoustical-simulation (accessed 13 April 2023).