Geometric Algorithms for Machine Based Optimisation of Acoustic Reflectors

John O'Keefe
O'Keefe Acoustics
Toronto, Canada
john@okeefeacoustics.com

***Abstract*— One of the challenges in optimizing acoustic reflector design in a performing arts venue is the potential interference with other elements in the room, notably sightlines for audience members or stage lighting. Conventional machine based optimization of geometric shapes like an acoustic reflector randomly augment the shape of the reflector inside rectilinear Bounding Boxes. The space available for acoustic reflectors in a theatre or concert hall rarely comes in the shape of a six sided rectilinear box. A collection of algorithms has been developed to address this issue, creating what might be called Bounding Breps. To test the geometries created within these B-Breps, a key component of any machine based optimization is the Objective or Fitness Function that the algorithm is working towards. An Objective/Fitness Function has been developed that optimizes reflected sound to arrive on a surface from lateral directions, at points evenly distributed on that surface. The Bounding Breps and the Objective/Fitness Functions were applied using both Simulated Annealing and Genetic Algorithms. The latter was found more appropriate for acoustic reflector design.**

Keywords—Room Acoustics, Spatial Sound, Multi-Objective Optimization (MOO), Simulated Annealing (SA),Genetic Algorithms(GA).

I. Introduction

The instigation for this study was a desire to optimize spatial sound in a "surround" shaped concert hall. It was inspired by an observation, commenting that "Most surround concert halls don't have sound that surrounds you". [1] The author developed the idea of a room taking the form of a hyperboloid [2]. Trying to optimize laterally reflected sound in this room and other more traditional rooms has led to a series of routines that might be used to create geometrical objects within Machine Learning algorithms.

In each case, the geometry of the reflector is governed by a matrix of control points (P_M). These are then used to construct meshes for a ray tracing computation of the acoustical performance. Unlike meshing from a point cloud, for example one generated from an image by a Delaunay mesh [3] the points, P_M, must remain ordered throughout the iteration process. Seven routines will be presented, followed by a discussion of a proposed Objective Function that aims to encourage lateral reflections, thus improving the spatial nature of the sound.

II. Bounding B-rep Routines

A. *Free Cartesian Grid B-Brep (The Hyperboloid Experiment)*

A simple hyperboloid will generate mostly overhead frontal reflections. A hyperboloid with an undulated inner shell might deliver more laterally orientated reflections. The concept of two nested hyperboloids acting as spatial boundaries was developed. The optimized surface generated between the two boundaries will become the undulated inner shell, i.e. the interior walls of the concert hall. The following describes how that surface is created.

A curve is drawn on a (world) X-Z plane. In [2] it was a Non-uniform Rational B-Spline (Nurb) curve that approximated a hyperbola but it could be any kind of curve that might be desired. This was then extruded by rotating the curve around an ellipse in the X-Y plane generating a surface with the approximate geometry of a hyperboloid. A volume such as this, created by Nurb-based curves, is often referred to as a Boundary Representation or Brep.

To create the Bounding Brep for the hyperboloid, (referred to in the following as a B-Brep), generate a grid of points in (relative) U-V space at mid-points between the two nested hyperboloid surfaces. Please see Fig 1. Join these points to create the U-V gridlines. Then, from each of the grid points, draw infinite lines. Each line should include the grid point and the point on the vertical centroid axis of the hyperboloids that is closest to the given grid point. Each of these lines is cropped

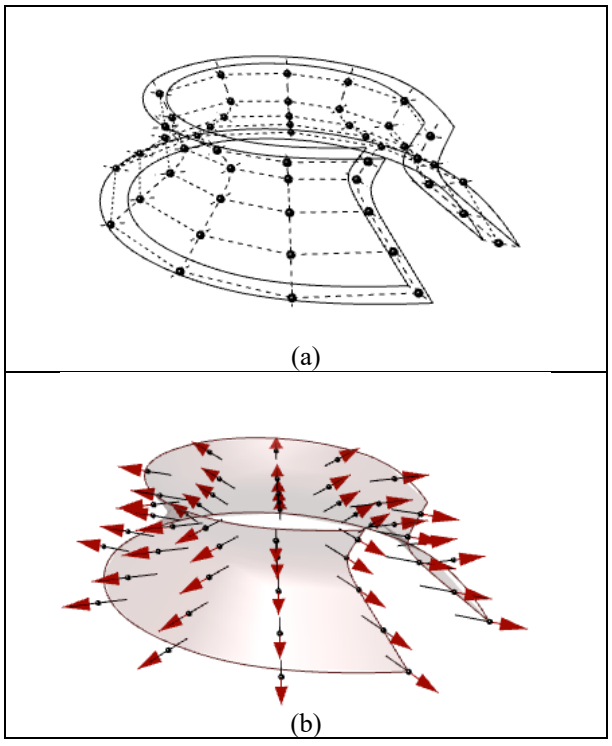

Fig. 1. Free Cartesian Grid B-Brep. Create two geometrical similar objects (in this case, two hyperboloids), one nested inside the other. Generate a grid of points mid-way between the two objects. Join these horizontally and vertically to create U-V gridlines, as shown in (a). For the third dimension (W), draw infinite lines that include one of the grid points and its closest point on the object's vertical centroid axis. Crop the infinite lines between the nested objects to create the W gridlines. Normalize all three sets of gridlines.

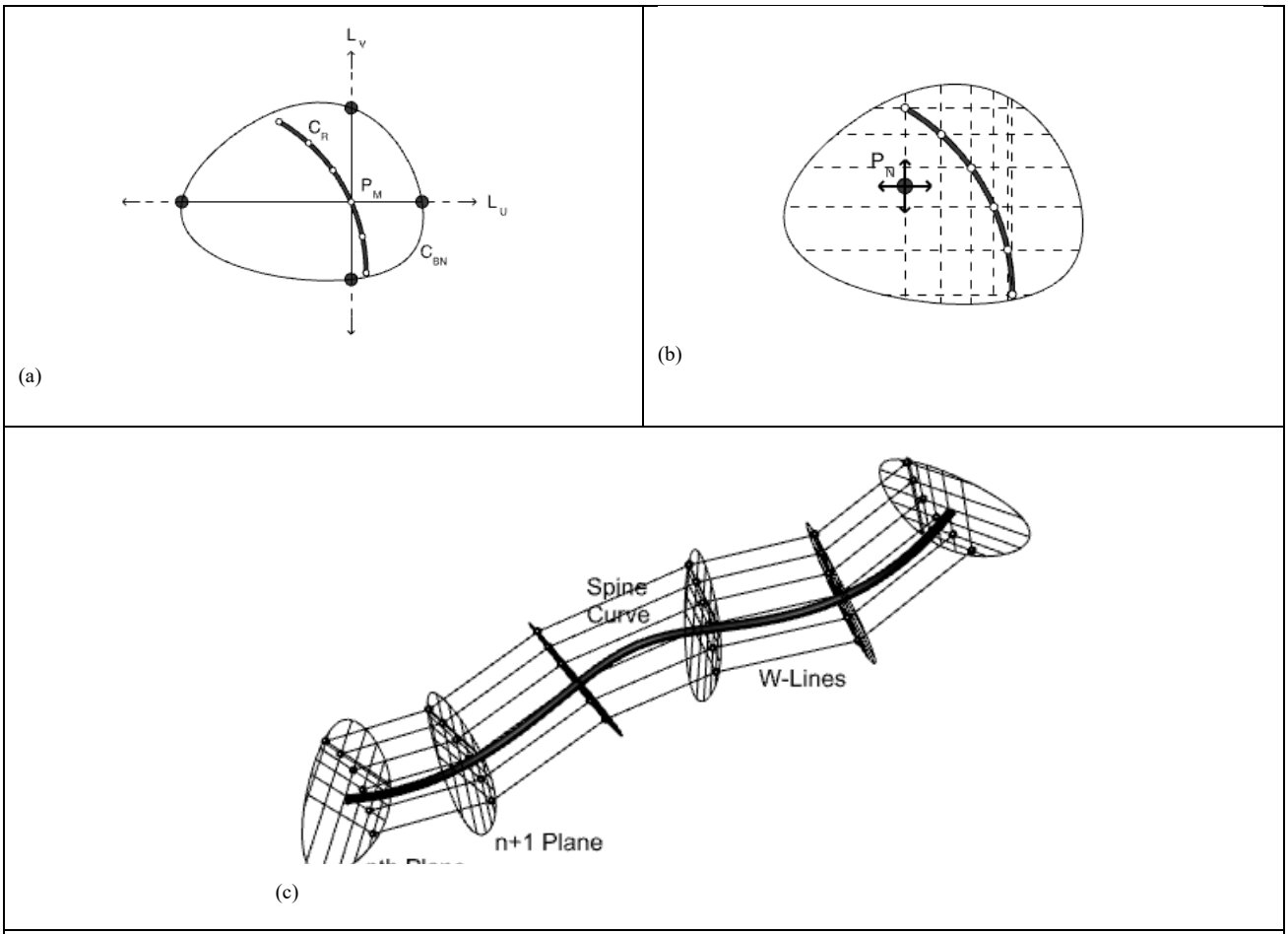

Fig. 2. Shaped Cartesian Grid B-Brep. In 2-D space (a) create a initial reflector curve, C_R, inside a Bounding Nurb curve, C_{BN}. Divide the initiating curve at control points, P_M. At each control point draw infinite lines in the horizontal and vertical direction, lines L_U and L_V. Determine the intersection points between the C_{BN} curve and the L_U & L_V lines. As shown in (b), create gridlines from these points. Normalize the U-V gridlines. Control points P_M traverse up and down, back and forth in 2-D along the gridlines shown in (b). For a 3-D perturbation, create a new curve to act as a "spine" as shown in (c). Divide the Spinal curve, C_S, at n points. Create planes that are perpendicular to the curve C_S at each of the n points. Project the gridlines created in (b) onto each of these planes. Draw gridlines between adjacent planes. These are the W-lines and they too are normalized. The control point P_M is moved at each iteration by means of a unit random number generator, first in the two U-V directions shown in (b) then in the third W-direction shown in (c)

between the two hyperboloids. These are the W gridlines. All three sets of lines are normalized (i.e. divided by their respective lengths) and new control points are randomly generated on them for each iteration of the optimization algorithm.

Using a Simulated Annealing algorithm [4], iterations of the perturbed hyperboloid were optimised then analysed. Lateral Fractions were estimated using the single reflection method proposed by Protheroe and Day [5]. Lateral Fractions were improved to the range of 0.25. Thus, a geometry that inherently generates frontal overhead reflections could be optimized to create Lateral Fractions that would be acceptable in a traditional shoe box shaped concert hall.

It was quickly found that different kinds of reflecting surfaces or objects in an auditorium will require different tools to generate them. What might be called geometric algorithms. Some of the more useful algorithms developed will be described.

B. Shaped Cartesian Grid B-Brep

For a reflector profile curve (C_R) inside a closed Bounding Nurb curve (C_{BN}), divide the reflector curve C_R into any number of control points (P_M), as shown in Fig. 2a. At each point, P_M, create two infinite lines in the U and V directions. Find the intersection points between these lines and the Bounding curve, C_{BN}. For the example curve C_{BN}, the two lines will generate two intersection points each. Create finite lines between the intersection point pairs, one in the U direction, one in the V-direction. Repeat this for the desired number of division points, m.

Normalize each line, thus giving each point, P_M, an axis in the U and V directions that has been parameterized to a length from 0.0 to 1.0 and is spatially limited by the perimeter of the Bounding Nurb curve, C_{BN}. These will be referred to the U and V axes and along them, each iteration of the optimization algorithm is free to move any control point P_M anywhere inside the 2 dimensional Bounding Box curve C_{BN}. This is facilitated by generating a random number at each iteration and is described further in Section III, below.

The third dimension, which will be referred to as the W-direction, is along the Spinal curve (C_S). Along this curve, there a number (n) of copies of the C_{BN} Bounding Nurb curve. These are created by dividing the curve C_S at n-points and, at each one of these points, a plane perpendicular to the curve is created. The Bounding Nurb curve, C_{BN}, and its associated gridlines are then projected onto each plane. The W-axes are created by joining a control point P_M on a grid line in the n^{th}

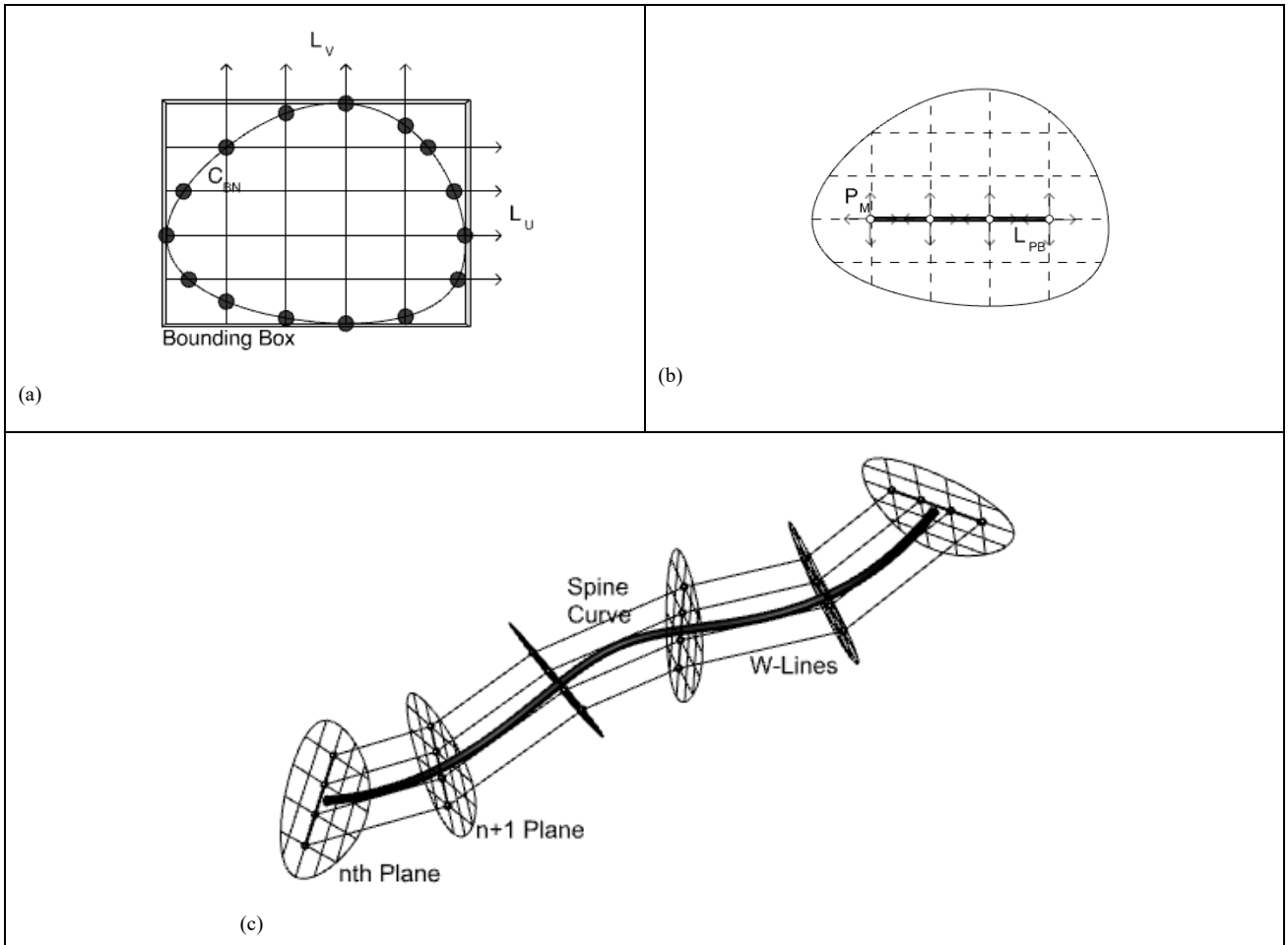

Fig. 3. Uniform Cartesian Grid B-Brep. Create a rectilinear Bounding Box around the Bounding Nurb curve C_{BN}, shown as a double line in (a). Divide the left and lower edges of the Bounding Box at n points and from each of these cast infinite lines in the vertical and horizontal directions. Do note cast them from the ends of the edge lines. Find the intersection points (solid circles) between these lines and the C_{BN} curve. Create gridlines between pairs of these points, as shown in (b) and normalize them. Draw a line to act as the initial base of the reflecting panel, shown as line L_{PB} in (b). Divide this into m = n – 2 points. These are the control points that will traverse along the U-V gridlines. Extrapolate this into 3 dimensions along the W-Lines, shown in (c).

C_{BN} grid line bundle to its partner in the n+1 C_{BN} bundle. These W-lines are also normalized and, in concert with the U and V lines, will give the optimizing algorithm full access to the 3-dimensional space inside the volume of the Nurb-based Bounding Brep (B-Brep) and, importantly, not beyond.

It should be noted that this method and others like it presented here, assumes that the Bounding Nurb curve C_{BN} is internally concave. A C_{BN} curve that is both internally convex and concave will create multiple intersection points with the lines L_U and L_V. This will result a set of gridlines with discontinuities. Shapes generated inside C_{BN} can be either convex or concave, for example the C_R curve shown in Fig. 2a, but the C_{BN} curve itself must be internally concave.

This so-called Shaped Cartesian B-Brep method, shown in Fig. 2, assumes and benefits from prior knowledge of the acoustical designer as to what shape of reflector might be a good starting point. But, as can be noted by observing the gridlines in Fig. 2b, the entire space inside the Bounding curve C_{BN} is not fully explored. Thus, a more uniform exploration algorithm was developed.

C. Uniform Cartesian Grid B-Brep

On the plane containing the Bounding Nurb curve (C_{BN}), build a rectilinear Bounding-Box around the curve, as shown in Fig. 3a. A rectilinear Bounding Box, in this definition, is a 2-dimensional rectangle fitted around the extremities of the C_{BN} curve. Take two lines from the Bounding Box, the left edge and the bottom edge. Divide both lines evenly into n points. From each point project infinitely long lines, L_U and L_V. From the left edge of the box, project the infinite lines in the positive U-direction of the construction plane. For the bottom edge, project in the positive V direction. Do not project L_U or L_V lines from the ends of the two edge lines. Find the intersection points between each L_U and L_V lines and the Bounding Nurb curve C_{BN}. There will be two points of intersection for each line. Join the two points to create a set gridlines that are now limited to the confines of the Bounding Nurb (C_{BN}). Normalize each one of the gridlines.

Now, create a line inside the curve C_{BN} or, better still, sort through the gridlines that have already been created and select the longest one. This will be the base of a panel that will be extruded down the length of the Spinal curve (C_S). It is shown in Fig. 3b as line L_{PB}. Divide this line into m control points (P_M), where m = n – 2. This will result in each control point

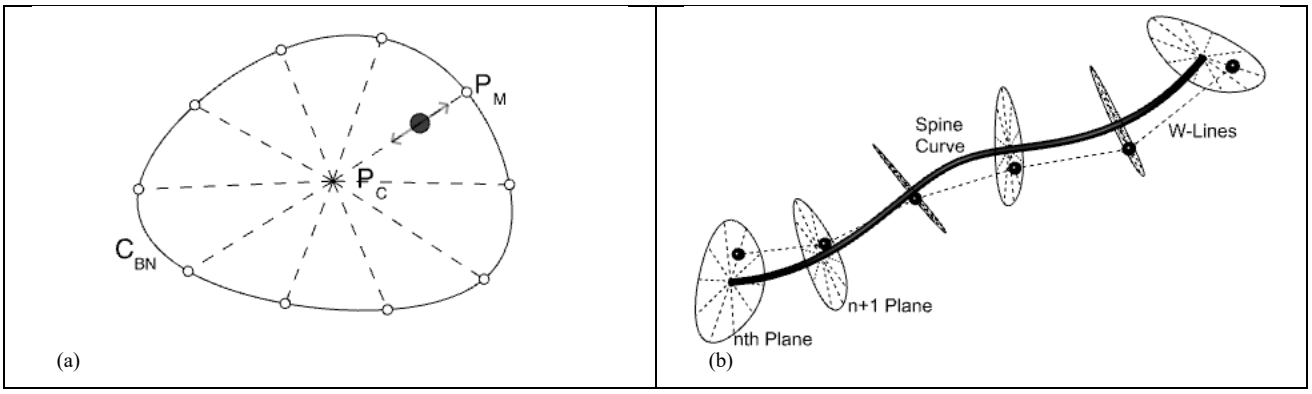

Fig. 4. Polar Grid B-Brep. Divide the Bounding Nurb curve (C_{BN}) into m control points, (P_M), as shown in Fig. 3a. Draw lines from the centroid of C_{BN} to the P_M points to create gridlines and normalize them. Extrapolate these into three dimesions along the Spine Curve, as shown in Fig. 3b.

being matched up with a U-V pair of gridlines. Referring to Figure 3b, the U-Line will be between the point P_M and its neighbour P_{M+1}, while the V-Line will be the full length vertical line associated with P_M. Expand this into three dimensions along the spinal curve (C_S) using the W-lines method described for the Shaped Cartesian Grid B-Brep (Section II-B). Perturb the control points along their respective gridlines for each iteration, as described in Section III.

D. Free Point Perturbation

The Free Point Perturbation algorithm is basically a simplification or refinement of the previous two methods and has proved to be the most flexible. Start with a single control point (P_M) on a plane, somewhere inside a Bounding Nurb curve C_{BN}. From that point, draw two infinitely long lines in the U and V directions of construction plane that contains the point P_M. As with the previous algorithms, find the intersection points between the infinite lines and the C_{BN} curve. Join the appropriate intersection points to form two lines, one in the U-direction and the other in the V-direction. Normalize these lines then move the P_M control point along the lines, as described in Section III. Extrapolate the movement of the point into 3 dimensions using the Spine curve and W-Lines method, described above and in Section III.

This method, unlike the other six methods described in this paper does not limit the movement of the control points to a pre-determined set of gridlines. In so doing, it offers a better exploration of the search space within the Bounding Nurb C_{BN}. It does however require a sorting and re-ordering of the control points on each of the spinal planes for each iteration of the optimisation algorithm.

E. Polar Grid B-Brep

The geometric algorithms described so far are appropriate for surfaces that reflect sound off a single side. Most objects inside an auditorium will reflect sound off only one side but some, for example a lighting booth located above the audience, will reflect sound off two or more sides. For this we need an algorithm that produces not a panel-like perturbed surface but, rather, a closed volumetric object. This is best done with a polar grid.

For a given Bounding Nurb curve (C_{BN}), find its centroid (P_C). Divide the curve C_{BN} into any number of control points (P_M), as shown in Fig. 4a. Draw lines between each point P_M and the centroid P_C. Normalize each of these lines. These lines, might be compared visually to the spokes in a wheel. They will be referred to here as V lines and will generate the X and Y components of the control point in a world X-Y-Z coordinate system. The third dimension, as before, is along the spinal curve, C_S. These are the W-lines and are created by connecting the mid-point of each of the lines in n^{th} bundle of V-lines to its mid-point partner on the n+1 bundle of V-lines. These too are then normalized. Generate the Z component of the point, P_M, using the method described in Section III.

F. Nested Rotational B-Brep

Consider, for example, the vertical space beside the stage that is available for a proscenium arch acoustical reflector. It is described, in plan, by the (available space) curve C_{AS}, shown in Fig. 5. If the optimization algorithm is allowed to explore and employ the entire space inside curve C_{AS} it will very likely end up occupying parts of that space unnecessarily. The eventual design of the acoustical reflector probably only requires the zone shown inside reflector space curve (C_{RS}) provided it is pointed, or rotated in the right direction.

In Fig. 5, the zone with the gridlines indicates the space that might be required for an acoustical reflector. The shaded zone might be used for loudspeakers, cables, duct risers or perhaps an unducted ventilation plenum. The same logic applies for a balcony front. Balcony facia can, and often do, provide very useful acoustical reflections. If there is space behind

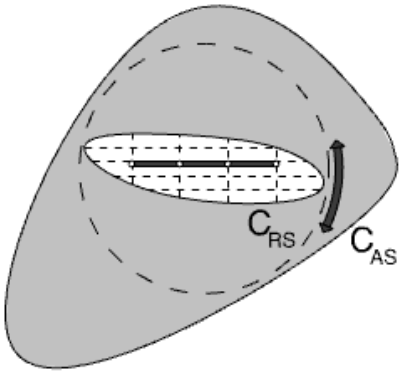

Fig. 5. Nested Rotational B-Brep. The shaded zone inside the C_{AS} curve is the space that is available for an acoustical reflector but might also be required for other components of the building. The grid line space inside the C_{RS} curve is the zone in which the reflector is optimised while be rotated randomly through the iteration process.

them, it can be used for electrical cabling, structure for lighting fixtures or even extra foot room for the patrons seated in the front row.

Recognizing the needs of disparate disciplines to occupy the same space in a theatre or concert hall, rotational versions of the methods described above have been developed. Four of the other six methods described here can be extended by adding a rotational variable. The exceptions are the Free Cartesian and the Polar Grid methods which, in themselves, occupy a closed volume and cannot be significantly optimized through rotation.

The size of the reflector search space (C_{RS}) will depend on its location inside the available space (C_{AS}). For example, in Fig. 5, if the reflector search space was located in the lower left hand corner of the C_{AS} curve, it would have to be much smaller. The rotational nature of the Nested Rotational method requires what might be called "temporary" search space. The actual search space, described by curve C_{RS}, will be allowed to rotate 180° and therefore needs enough room to perform this rotation inside the available space, C_{AS}. This, of course, forms a circle, the center of which will be the centroid of the C_{RS} curve. The radius of that circle will be half the length of the longest edge of the 2-dimensional Bounding Box surrounding the C_{RS} curve. The dashed line in Fig. 5 indicates the circle of the "temporary" search space for the C_{RS} curve.

To find the largest possible circle inside the available space (C_{AS}) curve, the following method is suggested. Generate a 2-dimensional Bounding Box around the available space curve C_{AS}. Draw two lines between the box's diagonally opposite corners, creating an "X", or cross, inside the C_{AS} Bounding Box. Iteratively traverse the parts of these two lines that are inside the C_{AS} curve with a "source" that has a rotating arm attached to it. During the traversals, search for the longest arm that can rotate 360° without touching the C_{AS} curve. The length of this arm will be the radius of the optimum "temporary" search circle and, hence, the longest possible dimension of the reflector's C_{RS} search space inside the available space curve, C_{AS}. (i.e. half the length of the longest edge of its 2-dimensional rectilinear Bounding-Box) The origin of the optimum arm will be the centroid of the reflector's Bounding Nurb curve, C_{RS}, which can be drawn in any shape, provided that it fits inside a circle with a radius no longer than the optimum arm's length.

G. Negative Space B-Brep

The Negative Space Bounding Box method grew out of the Free and Shaped Cartesian methods, described in Sections II-A and II-B. It takes its name from the term that a painter or sculptor might use to describe the parts of an artform that is free of material. So, for example, inside an auditorium, there are critical zones where acoustic reflectors cannot be placed. Most notably zones required for sightlines, although there will be others.

Fig. 6a shows a half section of a thrust stage theatre, based loosely on the Festival Theatre in Stratford, Canada. We shall refer to the inner shell of the room (i.e. walls, floors, etc.) as $Brep_{SHELL}$. Inside this, there are two zones (or Breps) emanating from the semi-circular lighting rows in the ceiling. These are zones required to project light onto the stage and they cannot, of course, be obstructed with acoustic reflectors. These will referred to as $Brep_{POS}$. We determine the negative space available for acoustic reflectors by applying a Boolean geometric subtraction:

$$B\text{-}Brep_{NEG} = Brep_{SHELL} - Brep_{POS} \quad (1)$$

There will often be more than one $B\text{-}Brep_{NEG}$ zone resulting from this subtraction, as indeed there are in the example

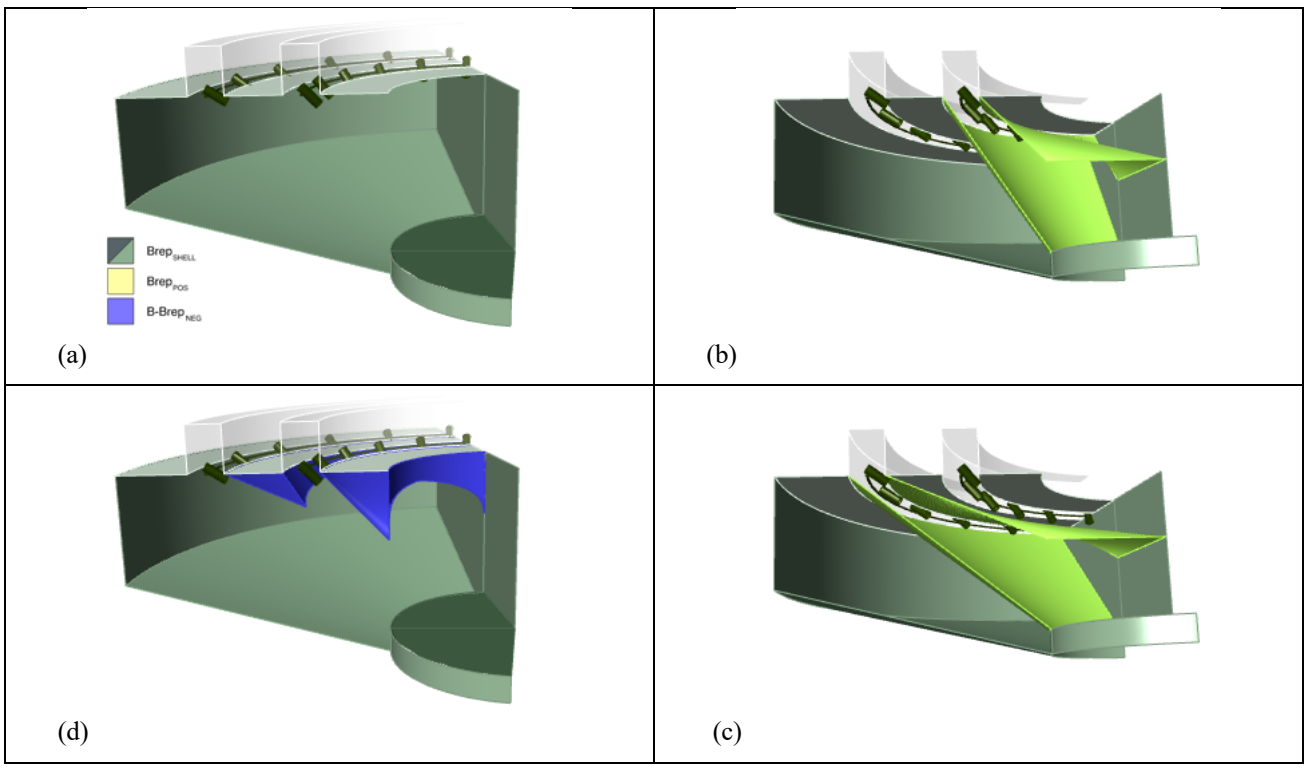

Fig. 6. Images of a semi-circular thrust stage theatre with two rows of lighting positions in the ceiling. The green zones in (b) and (c) indicate the space that cannot be interupted by an acoustical reflector. The blue zones in (d) indicate the remaining space, or B-Breps, in which acoustical refectors can be posittioned and their geometry optimized.

shown in Fig. 5d. We should note that this demonstration considers only two Brep$_{POS}$ lighting zones. There will, of course, be more Brep$_{POS}$ zones. Most importantly, sight-lines for the audience members. We should also point out that the wall above the front of the stage is not shown in the Fig. 6 for the sake of visual clarity. It explains the semi-circular termination of the negative B-Brep at the right end of Fig. 6d.

III. Iterative Creation Of New Control Points

The Bounding Breps (B-Breps) are defined by a matrix of control points (P_M) which, in turn, are moved along their respective gridlines. For each iteration of the optimization algorithm, a control point is perturbed as follows. Each grid line has been normalized and therefore has a parameterized length of 1.0. The incoming point is evaluated on its associated U and V lines on the n^{th} plane to determine its parameterized position (a, b) between 0.0. and 1.0. Two random numbers (dU and dV) are generated between -0.5 and +0.5 and then added to the original position:

$$(a',b') = (a + dU, b + dV) \quad (2)$$

The new value (a',b') is then limited to a range between 0.0 and 1.0. A parameterized version of the original reflecting curve or surface is then evaluated at parameters a and b to produce a point on the plane, P(a',b'). A new parameter c' is created in the same way along the W grid line. The point P(a',b') is then translated a distance of c' along the W-Line, between the n^{th} and the n+1 planes, resulting in the new point P(a',b',c'), described in (world space) X-Y-Z coordinates.

For Genetic Algorithms, the same principles were applied in concert with a Simulated Binary Cross-over (SBX) [7].

IV. Objective Function

In machine based optimization, perhaps the most important component is the question you ask the computer to solve. In essence, the solution or answer that the computer returns will only ever be as good as the question asked in the first place. In an optimization algorithm, the Objective Function is the question being asked.

For a given reflector casting reflections on a receiver surface, we want to shape the geometry of the reflector such that it casts as many reflections as possible onto the receiver surface, distributes these as evenly as possible and directs those reflections to arrive at the listeners from the sides – the more lateral, the better. Thus, we have what is known as a Multi-Objective Optimization (or MOO) problem.

The performance of a given reflector design iteration is determined as follows. The reflector is irradiated from a single source and reflections are calculated according to a simple ray tracing method [7]. The "lateralness" of each reflection intersecting the surface is calculated according to the method developed by Protheroe and Day [5], which we shall refer to here as the Single Reflection Lateral Fraction (sLF). Ref. [5] found a reasonable correlation between sLF and the 80 ms Early Lateral Fraction (LF), as defined in [8].

We have chosen to optimize LFs because, in the design of auditoria, if a reflector can improve the LF, it will also improve Clarity (C80). In addition, directing a reflection towards the audience or receiver surface to improve the LF, as opposed to letting it propagate towards a ceiling or wall, will increase the early Acoustic Strength (G80). If the room is proportionally tall with respect to its width this will, in all likelihood, lead to higher total Acoustic Strength (G) in the room [9]. To be clear, we have not chosen to optimize LF just for the subjective perception of Spatial Impression that it is usually associated with but, rather, for Spatial Impression, Clarity and Loudness combined. Likewise, we should point out that the decision to distribute the reflections as evenly as possible on the receiver surface is not driven by a desire for what might be described by some – including the author – as a bland one-size-fits-all sound that doesn't vary from seat to seat but, rather, to prevent the algorithm from optimizing the reflections into a small, confined zone, for example along an edge or in one corner of the receiving surface.

The kernel of the proposed Objective Function concept is to think of the sLFs created by the reflections as a bumpy field of small mounds or hills on top of the receiver surface. The centroid at the base of each mound is the point where the reflected ray has intersected the receiver surface. The height of the mound or hill is the sLF associated with that reflection. The goal of the Objective Function is to provide as many mounds as possible, distribute them evenly and make sure that the field is as flat as possible. What this means, respectively, is that the reflector will cast as many successful reflections as possible, will distribute them on the receiving surface uniformly in space and ensure that the level of the Lateral Fraction is also uniform. The field should be leveled to a pre-determined height. In this case, that height will be referred to as sLF_{goal}. In these studies, sLF_{goal} was typically set to 0.35 or 0.40.

There are many ways to level the sLF field to a uniform height with the maximum number of mounds, evenly distributed. Many have been experimented with in this study. We shall present the simplest solution developed so far.

In Multi-Objective Optimization, it is often preferrable to limit the number dimensions in the Objective Function. In what has been described so far, there are three dimensions. Sometimes an objective can be reduced to a constraint. In this study, it was decided to do this with the reflectors' efficiency (i.e. the number of successful reflections that are cast). Thus, if an iterated version of the reflector design does not cast as many reflections as the original design, it is rejected and does not move forward into the next iteration. In many cases, however, the threshold of rejection can be – and indeed was – set higher than the original design reflection count. The Objective Function now consists of a two part calculation, which will be described as Fitnesses A and B. Fitness A addresses spatial distribution on the receiver surface and Fitness B, the sLF uniformity.

A given reflector design iteration will cast a number of reflections that will intersect with the receiving surface. Think of each reflection intersection point of the surface as an object as one would in object-oriented computer code. (This is our sLF hill or mound.) The centroid at the base of the object is the intersection point and the height is the corresponding sLF for the given reflection. The object also has a radius associated with it. This forms a circle describing the base of the mound. The radius of the circle is half the distance to another receiver point that is closest to it. That is to say, the receiver point's Nearest Neighbor (NN). Each reflector optimization iteration will generate a different number of successful reflection intersections of the receiving surface. For a given number of circles on a surface, the mean distance between their centroids should be:

$$d_{mean} = \sqrt{\frac{Area\ of\ Receiving\ Surface}{\#\ of\ Receiving\ Points}} \quad (3)$$

This is the optimum packing distance between a receiver point and its NN. The radii of the circles at the base of the sLF mounds will be half of this. To evenly distribute receiver points on the surface, we create a grid of points on the surface spaced at on-center distances of d_{mean}. The goal of the Fitness A part of the algorithm is to minimize the distance between the grid points and their NN. The goal of the Fitness B part of the algorithm is to minimize the difference between the sLF at the NN and the target value of sLF_{goal}. We shall call this ΔsLF_{NN}.

The reasonably safe assumption is made that, over the course of the iteration process, the distances measured between the grid points and their NN will form a normal distribution. Likewise, we will assume that the differences between the sLFs at the NN and sLF_{goal} (ΔsLF) will form a normal distribution. Their contributions to the overall Objective Function are:

$$Fitness\ A = \frac{1}{N_{grid}} \sum_{i=0}^{N_{grid}} 1 - e^{\frac{d_{NN}}{\tau_{dist}}} \quad (4)$$

where: N_{grid} is the number of grid points
d_{NN} is distance from the i^{th} grid point to its NN
τ_{dist} is the convergence coefficient associated with distance

and:

$$Fitness\ B = \frac{1}{N_{grid}} \sum_{i=0}^{N_{grid}} 1 - e^{\frac{\Delta sLF}{\tau_{sLF}}} \quad (5)$$

where: ΔsLF is the difference between the sLF at the NN and sLF_{goal}
τ_{sLF} is the convergence coefficient associated with associated with sLF

As is often the case in optimization algorithms, the goal will be to minimize the values of these two functions.

The choice of how fast the optimization algorithms converge (τ) is arbitrary. We have found however that it helps to keep the operands of the two exponents in the same order of magnitude. In this study, distance was measured in mm. In an auditorium, that is typically on the order of 10^5 mm or more. The sLF will always have a value less than 10^1. This implies a convergence coefficient ratio of 1:10000. In this study, τ_{sLF} was typically set in the range of 0.05, which implies a τ_{dist} of 500.

One of the advantages of expressing Fitnesses A and B in the form of a normal distribution is that both equations (4) and (5) will always have a range between 0.0 and 1.0, despite the 1:10000 difference between the parameters that they quantify.

Over the last few decades, a number of methods have been developed to deal with a Multi-Objective Optimization problem. One of the more popular procedures has been the weighted sum method[10]. This was experimented with but the Lexicographical Order method [11] proved more reliable.

In Lexicographical Ordering of the sub-fitnesses, one decides, a priori, which of the sub-fitnesses is the most important. The overall fitness is then determined as one might order words in a dictionary. In the following example, we shall assume that Fitness A (for NN distance) is the most important. In the actual study, both Fitnesses were given priority for experiments; with similar results.

To perform Lexicographical Ordering, Fitness A is converted into an integer. With a range of 0.0 to 1.0, this will always return a value of 0.0 so the number must be multiplied by a constant: 10^k, where k is the desired number of digits for accuracy. This then becomes the whole number part of the overall Objective Function. The fractional part is taken from Fitness B, which in almost all cases, will be a fraction between 0.0 and 1.0. In the cases where it is not, Fitness B can be safely changed to the fractional value of $1.0 - 10^{-k}$, at least for the purposes of this study.

With this approach, a multi-objective problem is rendered into a single-objective decision. Which is to say that as one moves from one iteration to the next, or in the case of a Genetic Algorithm, from one generation to the next, there is only a one-dimensional test to select the "best" solution(s). One might think of it as an Objective Function bottle-neck. In addition to this, after the iteration process is complete, the end result will be a *single* reflector design that the algorithm suggests is the optimum. This, however, may not be the case. The search for the best Objective Function may have only found a local minimum, not the overall global minimum. (Remember that our goal is to minimize the Objective Function.) This we find is more likely to happen with Simulated Annealing (SA) and is, indeed, one of the SA algorithm's shortcomings when dealing with a Multi-Objective Optimization problem.

We have found that it is better to develop a so-called Pareto set of optimum reflector designs. A Pareto set is a collection of solutions, or optima, where, in the example we are using, the value of Fitness A cannot be improved without decreasing the value of Fitness B and, likewise, Fitness B cannot be improved without decreasing Fitness A. Genetic Algorithms [12] are more appropriate to solve multi-objective optimization problems, in part because they are based on a population concept. Simulated Annealing takes only a single "best" solution into the next iteration. A Genetic Algorithm takes a population of several into the next generation.

Finding a Pareto set of optimal solutions is sometimes better than finding a single optimum solution because there may be unknown characteristics among the "winning solutions" along the Pareto-optimal front that the designer might discover. Characteristics that were not previously quantified in the Objective Function. Also, in the design of a theatre or concert hall, decisions are best made by a team. Architecture, lighting design and, of course, financial concerns will all play a role in deciding the most appropriate geometry of the reflector or the room itself. Given this reality, proposing a set of solutions is often better than presenting an inflexible single solution.

For this reason, it is suggested that Genetic Algorithms might be more appropriate for acoustic reflector design than Simulated Annealing. In particular, the Non-dominated Sorted Genetic Algorithm II (NSGA-II) [13] has been applied and will be discussed in a subsequent paper [14].

ACKNOWLEDGMENTS

The author would like to acknowledge the numerous universities, institutions and private individuals currently making

their lectures and tutorials freely available on-line. Most notably the Harvard Graduate School of Design (Dr. Jose Castillo), the University of Stuttgart (Long Nguyen) and the University of Waterloo (Dr. Hamid Tizhoosh).

REFERENCES

[1] I. Hoffman, Private communication, Hamburg, 2019.

[2] J. O'Keefe, "Lateral Energy Ceiling Reflectors in Fan Shaped or Circular Rooms" ISRA Amsterdam 2019, Paper ID 2019/43 (Poster Session).

[3] L.J. Guibas,, D.E. Knuth, M. Sharir, "Randomized incremental construction of Delaunay and Voronoi diagrams", Algorithmica 7 (1–6): 381–413 (1992).

[4] S. Kirkpatrick, S., C.C. Gelatt Jr., M.P. Vecchi, "Optimization by Simulated Annealing". Science. 220 (4598): 671–680, (1983).

[5] D. Protheroe, C. Day, "Validation of lateral fraction results in room acoustic measurements", InterNoise Melbourne, 2014.

[6] A. Krokstad, S Strom, S. Sorsdal, "Calculating the acoustical room response by the use of a ray tracing technique", J. Sound Vib. 8 (l), 118-125 (1968).

[7] K. Deb, R. B. Agrawal, "Simulated Binary Crossover for Continuous Search Space", Complex Systems Vol. 9, 115-148 (1995).

[8] ISO 3382-1, 2009. "Acoustics - Measurement of room acoustic parameters - Part 1: Performance spaces", International Organization for Standardization, Geneva.

[9] J. O'Keefe, "Acoustical problems in large post-war auditoria", Proc.IOA, Vol. 24. Pt 4. (2002)

[10] P. Hajela, C.Y. Lin, "Genetic search strategies in multi-criterion design". Structural Optimization, 4(2), 99-107 (1992).

[11] P. C. Fishburn, "Lexicographic orders, utilities, and decision rules: A survey". Management Science, 20(11), 1442–1471 (1974).

[12] J. Holland, "Adaptation in natural and artificial systems", Ann Arbor, MI: University of Michigan Press, 1975

[13] K. Deb; A. Pratap, S. Agarwal, T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II". IEEE Transactions on Evolutionary Computation. 6 (2): 182 (2002).

[14] J. O'Keefe, "Applications of Machine Learning Bounding Breps for Optimised Acoustical Reflectors", EuroNoise 2021, unpublished.