DOI: 10.61782/fa.2025.0248

FORUM ACUSTICUM EURONOISE 2025

COMPUTER-BASED CONSTRUCTION OF SELF-SIMILAR ACOUSTIC REFLECTORS

John O'Keefe

O'Keefe Acoustics, Toronto, Canada

ABSTRACT

If one wants to optimise an acoustical reflector using computer aided methods such as Genetic Algorithms (GA), one must first develop methods of constructing those reflectors inside a computer. A number of these constructors have been developed by the author, many of them using Non-uniform Rational B-splines (Nurb) geometries that create bumpy, wave-like reflectors. The control points used to build these reflector surfaces are, by necessity, placed on a fixed grid where they are free to move but not so free as to create a completely random origami-like surface. The fixed grid, however, often limits the reflectors to slow undulating waves that are presumably more efficient at lower frequencies. Higher frequency articulation is impractical with a fixed grid because the resulting peaks and valleys of the waves will be too deep and narrow. A new method has been developed that can vary the density of the fixed grids as the GA progresses through its evolution. The result is a reflector surface with a more natural looking wave pattern. A self-similar pattern of waves within waves not unlike the surface of the sea.

Keywords: *Room Acoustics, Genetic Algorithms, Multi-objective Optimisation*

**Corresponding author*: *john@okeefeacoustics.com.*

1. INTRODUCTION

To facilitate the optimisation of acoustic reflectors using computer based multi-objective techniques such as Genetic Algorithms (GA), the computer must first be given methods to construct the reflectors. Just allowing the computer to randomly perturb a surface's control points will result in an impractical origami-like surface. The author has developed a number of methods to construct reflectors inside a computer. See, for example, ref. [1]. Traditionally, surfaces like an acoustic reflector have been perturbed inside rectilinear Bounding-Boxes. The control volumes developed by the author are non-rectilinear; created from Nurb curves. Surfaces or volumes created by Nurb curves are referred to as Boundary Representations or Breps. The new perturbation control volumes, are referred to therefore, not as Bounding Boxes but, rather, as Bounding Breps (or BBreps).

Most of the BBrep construction methods that have been developed are based on the vertebrate structure of a spine with skeletal appendages. Arguably the most successful structural concept since the dawn of visible life in the fossil record, some 500 million years ago. In these BBrep constructions, the "spine" is, more often than not, a Nurb curve, although sometimes a simple line may suffice. Attached to the spinal curve are lines and planes, perpendicular to the curve. The lines and planes are used as a guide or "track" upon which the Control Points for the reflecting surface are perturbed

.

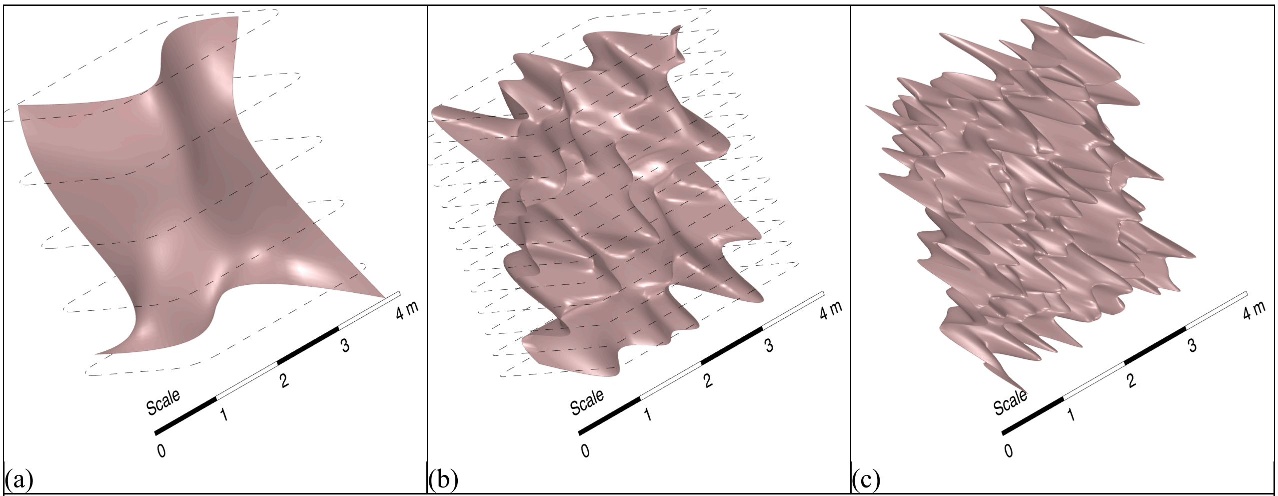

Figure 1 Reflectors created with the Spinal BBrep Constructor [1], all inside the same BBrep but using different Control Point grid densities. The Boundary Nurb (BN) Curves are shown as dashed lines in (a) and (b) but omitted in (c) for visual clarity. The grid densities are: (a) 5 x 5, (b) 13 x 13, (c) 25 x 25.

What will be referred to here as the Self-Similar BBrep Constructor, is an extension of some of the previously developed methods [1], [2], [3]. An example of a reflector built using the Spinal BBrep Constructor from ref. [1] is shown in Figure 1(a). A geometry that typically incorporates slow, undulating curves and is, presumably, less efficient at scattering high frequencies. If a design requirement calls for better high frequency scattering, the only way to do it with the previously developed constructors would be to increase the density of the Control Points. This however will result in geometries that are very difficult to build and are of dubious acoustical value

This is demonstrated in Figure 1, which shows three versions of a reflector built using the Spinal BBrep Constructor with varying densities. In this example, the BBrep is a rounded rectangular box, approximately 3.0 x 3.0 x 1.0 metres in size and with Control Point densities ranging from 5 x 5 to 25 x 25. For a BBrep of these dimensions, and with a Control Point density of 25 x 25, the resulting reflector could – and sometimes does – have valleys as deep as 1000 mm between peaks that are only 40 mm apart. Figure 1(c) shows an example of this extreme situation.

2. SELF-SIMILAR BBREP CONSTRUCTOR

The Self-Similar Constructor addresses this problem by varying the density of the Control Point grids as the Genetic Algorithm (GA) progresses through its evolution. The result is a reflector surface with a more natural looking wave pattern. A self-similar pattern of waves within waves not unlike the surface of the sea.

2.1 Background – The Control Point Gene

Before introducing the new surface constructor, however, the manipulation of the Control Points used to build them must be explained. In the parlance of Genetic Algorithms, the Control Points are the genes. The reflector surface and the reflection fields it creates are the genome.

As the name suggests, the Control Point Gene is a computer object that controls the shape of the reflector surface. The essential property being, of course, the point's position in 3 dimensional space. There are, however, several other properties of the gene that are used to control its perturbation. These are illustrated for a single gene in Figure 2 (a). Figure 2(b) shows the arrangement for a typical group of genes that share the same spinal plane. Each Control Point Gene contains the following properties:

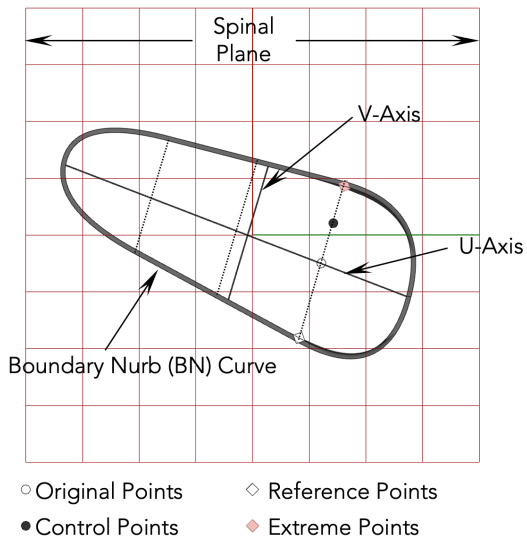

Figure 2(a) Properties of a single reflector Control Point Gene, located to the right of the V-Axis.

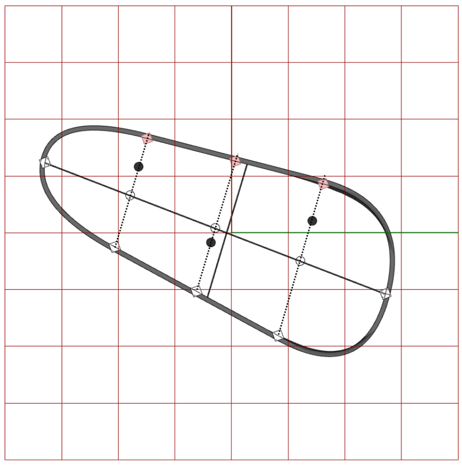

Figure 2(b) A typical collection of five Control Point Gene on a Spinal Plane.

(i) an Original location where it starts out on the U-Axis. This is shown in Figure 2(a) as an open circle to the right of the V-Axis.
(ii) a perturbed location (solid circle). Referred to as the Control Point, this is the actual point used to shape the surface.
(iii) two points indicating the furthest possible perturbations. These are referred to as the Reference Point, shown as a rotated open square, and the Extreme Point, shown as a shaded square.
(iv) each gene, as an object in computer code, has its own U and V axes.
(v) each gene has its own vector in the V-direction, derived from its V axis.
(vi) each gene has its own Spinal Plane.
(vii) each gene has its own Boundary Nurb (BN) Curve.

Properties (i) to (iii) are unique to each gene. Properties (iv) to (vii) are shared amongst the genes that share the same Spinal Plane.

The construction of Control Point Genes for the Self-similar BBrep Constructor is essentially the same as that for the Spinal BBrep Constructor [1]: points are manipulated on a grid that has been created by a central Spinal Curve with appendages that act as the axes of the grid. The difference in the methods comes from the preparation of those grids and the Control Point Genes associated with them. In the case of the Spinal BBrep Constructor, the grid is fixed throughout the Genetic Algorithm's (GA) evolution. The Self-similar Constructor is not, in a sense, a single evolution of a GA but, rather, a series of GAs, each series building on the previous one. Thus, as the Constructor moves from one series to the next, a preparation stage is required to modify the genomes and, in particular, the density of the Control Point grid and the range of Control Point perturbation allowed on that grid.

2.2 Procedure

The basic concept is to perform a complete optimisation run then take its results to create the Control Point Gene structure for the next run, only this time with a higher number of Spinal Planes and a higher number of Control Points on a given U-Axis. After much experimentation, the following procedure has been developed:

(i) The terminology of the series' progressions has been borrowed from Genetic Algorithm studies. That is, $t = 1$ describes the first series, $t = 2$ the second, etc. Thus, as one moves from one series to the next, the progression is from the t^{th} to the $(t + 1)^{th}$.

(ii) Moving from one series to the next, the density of the Spinal Planes should increase by a factor of 2. That is, similar to a musical or acoustical octave. Other factors were considered and experimented with, including the concept of applying a Fibonacci sequence. At the time of this writing however, they have proved too computationally cumbersome. They may be explored further in the future.
(iii) The U-Axes for the Control Point Genes, which in the Spinal BBrep Constructor [1] is a line, needs to be a curve in the Self-similar Constructor. This has proved to be the best way to pass on, or inherit, the optimisation results from one series to the next. The U-Axes for the $t+1$ series comes from one of the reflectors optimised in the t^{th} series. The chosen reflector is then "sliced up" by the Spinal Planes from the $t+1$ series. The curves created by the intersection of the reflector (from the t^{th} series) with the Spinal Planes (from the $t+1$ series) produces the U-Axes to be used in the $t+1$ series.
(iv) It was found, after much experimentation, that the Spinal Curve, upon which the Spinal Planes are built, should remain constant from one series to the next. It could, in principle, be changed, for example a $(t+1)$ Spinal Curve could be interpolated from a reflector in the t^{th} series. But this proved extremely difficult to control, notably when Control Points have to be matched to Spinal Planes. The less complicated procedure proved more efficient. It was computationally faster and still provided the appropriate amount of surface perturbation.

While experimenting with the "slicing" of the t^{th} reflector with the $(t+1)$ Spinal Planes, (Item iii above) it was found that the intersections of the plane and the reflector weren't always clear at the ends. Depending on the profile of the optimised (t^{th}) reflector, a Spinal Plane at the end of the Spinal Curve might only intersect with part of the reflector. Resulting in a foreshortened U-Axis at the end of a Spinal Curve – or, indeed, a foreshortened U-Axis at both ends of the Spinal Curve. The solution to this problem was to implement an additional sub-routine to randomly move the Spinal Planes small distances along the Spinal Curve until a longer U-Axis was found.

The concept of applying Fibonacci sequences to increase the density of the Spinal Planes, as noted above, was considered and – at least for the time being – abandoned. Genetic Algorithms (GA), however, are a bio-inspired process and it was thought prudent to make the connection from one series of GAs to the next with a number or sequence that is so prevalent in nature. Although this proved impractical for Spinal Plane densities it is fairly straightforward exercise when it comes to Control Point displacements. The following method was developed.

Moving from Series t to Series $t+1$, the procedure starts with one of the reflectors from Series t. Please see Figure 3(a). Intersections are found between this reflector and the Spinal Planes of the $t+1$ series. These are seen as the dashed curves in Figure 3(a) and they will become the U-Axes for the $t+1$ series. Figure 3(b) shows an isolated view of one of the Boundary Nurb (BN) Curves and one of the U-Axes. (In this example there are 10 sets of BN Curves, U-Axes, etc. Displaying them all in the same image is not practical.) Also seen in this image are the Control, Reference and Extreme Points in the positions developed from the t^{th} series. Lines have been drawn between the Reference and Extreme Points and these will be used in the re-sizing exercise.

The RhinoCommon method $Scale()$ [4] allows for the re-sizing of many objects, including lines. The speed at which the Self-similar Constructor converges from long waves to short waves is chosen by the user and is entered into the algorithm. This number is then multiplied by the inverse of a Golden Ratio $(1/\varphi = 0.61803)$ in the hope that the re-sizing might follow a more natural pattern. The re-sizing of each line is centred around its associated Control Point, shown in Figure 3(b) with the small black spheres. The result is then seen in Figure 3(c) for a single Spinal Plane and in Figure 3(d) for the entire reflector. A transparent copy of the t^{th} series' reflector has been included in Figure 3(d) for reference.

In the final step in the transition between series, it was found that a perturbation of the Control Point Genes had to be performed. Without some sort of displacement of the Control Point Genes, no matter how small it might be, the reflectors of the $t+1$ series will look the same as those in the t^{th} series. This is because, during the Recombination process, if the two "parent" reflectors have an exact copy of a t^{th} series profile, they will always breed a "child" reflector with a similar t^{th} series profile. The perturbation is performed by means of a mutation procedure [5]. The mutation method is used, as opposed to a simple random perturbation of the points, because the latter would merely scramble all of the optimisation achieved so far in the previous series.

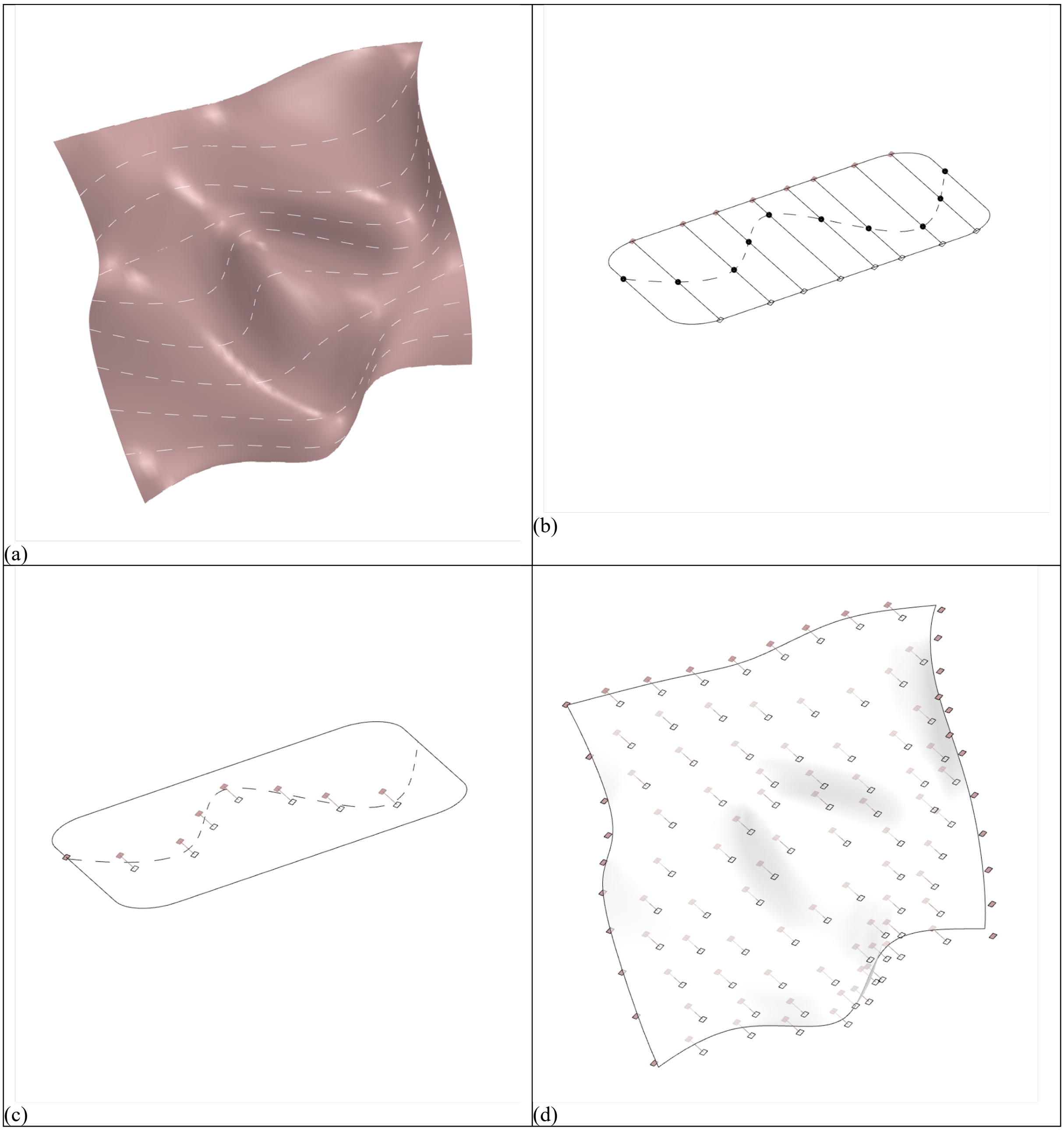

Figure 3 Creating U-Axes and resizing extremity points as the Self-similar BBrep Constructor moves from the t^{th} to the $t+1$ series. The U-Axes are shown as dashed lines. The Reference and Extreme Points are shown open and shaded squares, respectively. In panel (b), the Control Points are shown as the small black spheres.

An example of the reflector construction is shown in Figure 4. The three images follow a progression from series to series, similar to that shown in Figure 1. Note that the Self-Similar reflector with the highest density Control Point grid (Figure 4c), when compared to the Spinal BBrep Constructor's equivalent (Figure 1c), is a much more manageable geometry. One that is more easily manufactured and, presumably, of higher acoustical merit. Like the waves on the sea, the reflector in Figure 3(c) does, indeed, show a self-similar pattern.

3. ACKNOWLEDGMENTS

The author would like to thank Bård Støfringsdal, who posited the question in the first place and who, along with Yann Jurkiewicz and Jean-Dominique Polack, contributed to the discussions and review afterwards.

4. REFERENCES

[1] J. O'Keefe, "Zone to Zone Reflector Optimisation Using a Genetic Algorithm," in Proc. Inst. of Acoustics, Athens, Greece, 2023.

[2] J. O'Keefe, "Geometric Algorithms for Machine Based Optimisation of Acoustic Reflectors," in Immersive & 3D Audio (I3DA), Bologna, 2021.

[3] J. O'Keefe, "Applications of Machine Learning Bounding-Breps for Optimised Acoustical Reflectors", Proc. of Euronoise, 2021.

[4] R. M. &. Associates, "RhinoCommon API," 1 June 2015. [Online]. Available: https://developer.rhino3d.com/api/rhinocommon/rhino.geometry.transform/scale [Accessed 20 Mar 2025].

[5] K. Deb and S. B. Agrawal, "Simulated binary crossover for continuous search space," Complex Systems, vol. 9, no. 2, pp. 115-148, 1995.

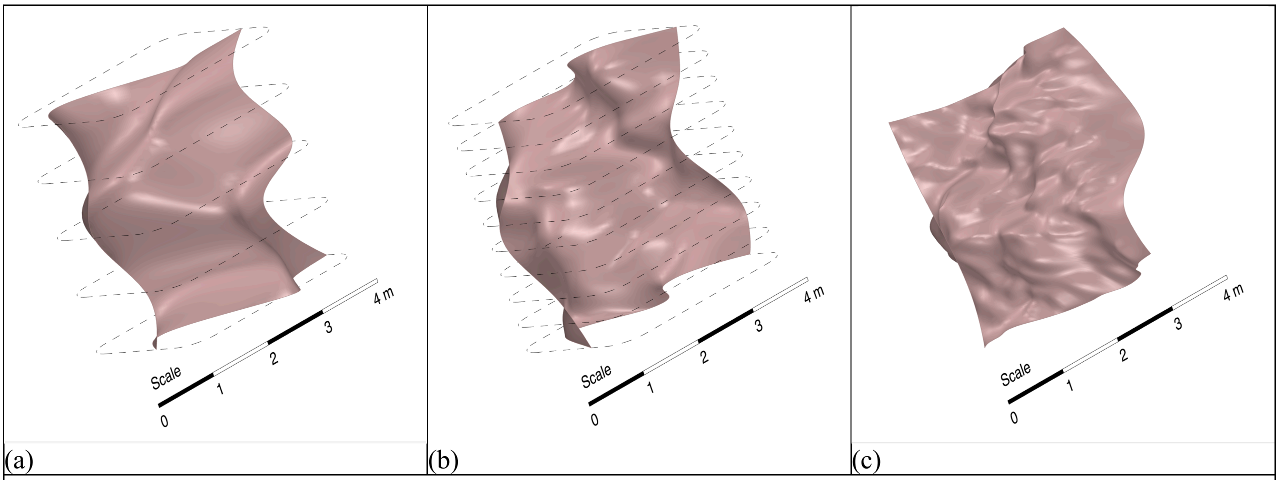

Figure 4 Reflectors created with the Self-similar BBrep Constructor, using the same dimensions and grid densities as the reflectors in Figure 1. The Boundary Nurb (BN) Curves are shown as dashed lines in (a) and (b) but omitted in (c) for visual clarity. The grid densities are: (a) 5 x 5, (b) 10 x 10, (c) 20 x 20.