SYNTHESIS BY MATHEMATICAL MODELS: ELLIPTIC FUNCTIONS AND LACUNARY SERIES

Vittorio Cafagna and Domenico Vicinanza

Musica Inaudita
Dept. of Mathematics and Computer Science
University of Salerno, Italy
cafagna@unisa.it dvicinanza@unisa.it

ABSTRACT

Sound synthesis methods can be interpreted, from a mathematical point of view, as a collection of techniques of selecting and conceptually organizing elements of a Hilbert space. In this sense, mathematics, being a highly structured and sophisticated system of classification, modeling and categorization, seems to be the natural tool to describe existing synthesis methods and to propose new ones. Because, from this perspective, one can think of any available (or theoretically predictable, or imaginable) synthesis method as a collection of procedures to deal with meaningful parameters, with the term ”synthesis by mathematical models” we mean an extensive use of the modeling and categorization power of mathematics applied to the world of sounds.

In this paper we give a few examples of sound synthesis techniques, based on mathematical models. After reviewing shortly FM synthesis and synthesis by nonlinear distortion, and suggesting some, to our advice, interesting open problems, we propose two different new methods: synthesis by means of elliptic functions and synthesis by means of nowhere (or almost-nowhere) differentiable functions and lacunary series.

The resulting waveforms have been produced using CSound as an audio engine, driven by Python scripts.

1. INTRODUCTION

A sound synthesis method is mainly an organization technique: each method selects a parametrized collection of sounds among all the elements of a Hilbert space. To every choice of a frequency ν it is naturally associated the Hilbert space of functions of period $T = \frac{1}{\nu}$, interpreted as the collection of all possible synthesizable sounds with that fundamental frequency. From this point of view, a synthesis technique is a selection strategy. FM synthesis, for example, selects sounds by means of three parameters: the carrier frequency ω_c, the modulator frequency ω_m and the modulation index I. The celebrated formula

$$\sin(\omega_c t + I\sin(\omega_m t)) = \sum_{k\in\mathbb{Z}} J_k(I)\sin(\omega_c t + k\omega_m t)$$

which is at the basis of the FM *miracle*, allows us to think of an FM sound as an element of a Hilbert space H of functions of a certain period determined by ω_c, ω_m, I. When ω_m is a multiple of ω_c, for every value of I the sound produced stays in the same Hilbert space of functions of period $T = \frac{1}{\omega_c}$. The mathematical model is a curve, for which we propose the name of *Bessel curve*, $\beta : \mathbb{R}_+ \to H$ defined by $\beta(I) = \{J_k(I)\}_{k\in\mathbb{Z}}$. Leaving ω_c fixed, changing ω_m causes your curve wandering around different Hilbert spaces, whose period is determined by ω_m and I. Therefore the complete model of FM synthesis can be interpreted geometrically as a surface (the *Bessel surface*) in a mathematically complicated object, defined as the parametrized family of all Hilbert spaces of functions of positive periods. Adding variability in ω_c generates a three-dimensional manifold (the *Bessel 3D manifold*.)

An obvious question to ask is *how many* sounds you get employing this procedure. Fourier’s theorem essentially says that you can produce *every possible* sound (with a given fundamental frequency) by additive synthesis. It is unlikely to expect to obtain with just two oscillators (as opposed to infinitely many) the same result. In geometrical terms, this would mean that the Bessel curve is a space-filling curve, what for many good reasons we are tempted not to believe. Still, the question remains. Experience suggests that the number of FM sounds is *huge*, which is akin to say that the Bessel curve goes around a lot. The mathematical question is what does *a lot* mean. Maybe, even if it does not pass through every point of the Hilbert space, the Bessel curve might eventually pass near each. Then the question becomes what does *near* mean. Analogous questions can be asked for the Bessel surface and the Bessel 3D manifolds. We think of these as interesting open problems.

More generally. FM synthesis can be seen as an example of nonlinear distortion. You take a sound (the carrier) and you modulate it with the help of a function (the modulator). The abstract framework is to take a function $f : \mathbb{R} \to \mathbb{R}$, and define a nonlinear operator $\tilde{f} : H \to H$ by the obvious formula $[\tilde{f}(s)](t) = f(s(t)),\ \forall s \in H$. Taking into account some technical hypotheses on the function f, which we are not going to discuss here, this provides you with a good (in some suitable sense) nonlinear operator on the Hilbert space H (mathematicians call it a Nemytski operator.) FM modulators are an example, others are Chebishev polynomials and operators defined in terms of finite summation formulas. Here again, there is a vast amount of pure-mathematical questions to be answered, which are likely to be meaningful for the world of synthesis. Are there other examples of Nemitsky operators with a predictable behaviour from a spectral point of view? In mathematical terms, can you describe the action of some Nemytski operators *via* the action on some bases of the Hilbert space? Maybe bases others than Fourier’s, like Bessel’s or Gabor’s or wavelets, or whatever. Or: what happens if you perturb a nice-behaved operator (like, for example, a Chebyshev polynomial?) And so on. We think of these as a sample of many interesting (very) open problems.

In this paper we propose two examples of sound synthesis methods based on well-known and highly developed pieces of mathematical knowledge, the theory of elliptic functions and the theory of lacunary series, with its *cornucopia* of nowhere or almost-nowhere differentiable functions.

Elliptic functions are complex functions of a complex variable, doubly-periodic and meromorphic. Double-periodicity is the main reason of interest for the synthesis of sounds. The class of doubly-periodic functions contains as a particular case the class of simply-periodic ones. But, as is frequently the case in mathematics, the passage from real to complex world allows the discovery of deep relations within objects, invisible from a one-dimensional perspective. Think, as a perspicuous metaphor, of the relations between trigonometric functions and the exponential, which demand the introduction of complex numbers to become apparent. One of the most important results of the theory is that any two elliptic functions are connected by an algebraic relation. This is patently false for simply-periodic functions: think of e^z and e^{e^z}. The implications are that you can, in principle, transform any sound into any other by manipulating a few coefficients of two polynomials. Also, as a consequence, you can express any elliptic function as a rational function of some chosen *simple* ones, which play, so to say, the role of a basis, the most historically established choice being that of the Weierstrass $\wp$ and its derivative $\wp'$. Meromorphy, which is the cause of the beautiful formulas, brings in singularities. In an expressive way (see [4]), one could say that noise is the price one has to pay in order to have an algebra of sounds. But noise appears in a very controlled and structured manner. Sounds produced by elliptic functions are very rich, the richness coming from the fine structure of the spectrum produced by the existence of singularities.

The interest for synthesis of continuous nowhere and almost-nowhere differentiable functions lies in the fact that their graph is incredibly jagged. Again, this gives rise to interesting sounds. The surprising fact is that this wild behaviour can be described very simply in terms of Fourier coefficients. Here are two examples, discussed later on:

$$R(x) = \sum_{k=1}^{\infty} \frac{1}{k^2} \sin(k^2 x) \tag{1}$$

which is called the Riemann function, and

$$W(x) = \sum_{k=0}^{\infty} a^k \cos(b^k \pi x) \tag{2}$$

$0 < a < 1, \quad b > 1 \quad ab > 1 + 3\pi/2,$ b an odd integer, which is the Weierstrass function. In both cases one has gaps between the frequencies of two adjacent terms of the series (the reason these series are called *lacunary*), gaps which increase as k becomes larger. Here the situation seems to be opposite with respect to the former case. While spectra of sounds produced by elliptic functions are very rich, due to the presence of zones with a high density of spectral lines, in this case the spectral lines are sparse. Still, the philosophy is the same: to produce a large organized class of interesting sounds, by manipulating few well-understood parameters. At the end of the paper, we reproduce the image a simple interface in CSound to play around with lacunary series and to experiment the dramatical changes in timbre due to small variations in the parameters. The CSound files, plus an archive of sounds, waveforms, spectra, Python scripts and Mathematica notebooks related to the items discussed in this article can be downloaded from the site `http://www.musicainaudita.it`.

2. COMPUTATION AND SOUND SYNTHESIS

All the waveforms described in this article have been computed using Python and CSound. The main strategy is to perform all the computations using free software available for different platforms (to have a tight control both at computation and rendering level). The idea to couple an audio engine (the acoustic compiler CSound) with a scripting language (Python) is motivated by our main goal: high flexibility and efficiency, together with high quality audio files.

Among the possible choices of acoustic compilers and scripting languages, there were many reasons to choose Python and CSound. First of all, CSound is free and available for all platforms. Its file format (a couple of ASCII files, named Orchestra and Score) is a well known standard with thousands of users spread around the world. Finally Csound could run offline on a personal computer, on a workstation or even on a linux cluster.

About the scripting languages, our interest was addressed to Python, since it is free, cross-platform and object-oriented. Python allows in natural way the creation of efficient and easy-to-maintain software environment; moreover, it allows the integration into websites, using CGI.

Python has been used to produce the right CSound files to drive the acoustic compiler, according to the mathematical model taken into account. In some cases, thanks to the mathematical libraries available (`math` for usual trigonometry and `cmath` for complex analysis), it has been used to quickly develop numerical calculation packages to compute actual values of the audio samples. After the computations, Python scripts prepare the score and orchestra files to generate the waveform according to the sampled values obtained. In a certain sense, we can say that CSound is playing the role of digital-to-analog converter.

In other situations (for examples, when computing with the Riemann and Weierstrass nowhere differentiable functions), audio files have been created by CSound by additive synthesis. In this case Python has been used to set up the instrument, using CSound opcodes.

Coupling an acoustic engine and a scripting language which includes a complex and wide set of math libraries, opens up a wide range of possibilities, which run from algorithmic composition using CSound instruments and scores, to the complete low-level control of audio samples.

Sound synthesis by mathematical models becomes, in this way, the combination of two well distinct activities, each of them performed by an optimized system: numerical calculations (using Python scripts) and sound production (CSound). The two phases could be carried on independently, even at different times and on different platforms.

3. SYNTHESIS BY MEANS OF ELLIPTIC FUNCTIONS

3.1. Elliptic functions

Synthesis by means of two-variables functions has been investigated in many different ways (see, e.g., [1], [2]) since the pioneering article of Mitsuhashi ([3]). The usual point of view is to choose a surface, a closed curve (*orbit*) on the surface, and to produce a waveform sampling the function on the curve. Orbits play a very important role in n-variable synthesis, their shape and geometrical characteristics affecting the final waveform. Closed orbits return periodic waveforms and open orbits (such as spirals) seem to be a promising tool to explore time-evolving sounds.

With respect to the existing literature, we decided to use complex functions of a complex variable, as opposed to real functions of two real variables. The main reason for this choice, alluded to already in the Introduction, is that complex analysis is a very rich and successful mathematical theory, with a strong tendency to bring to the surface unseen relations between real objects and to organize sparse arguments in a very articulated structure.

Let us come to the definitions.

A function $f : \mathbb{C} \to \mathbb{C} \cup \{\infty\}$ is doubly periodic with periods ω_1 and ω_2 if ω_1 and ω_2 are linearly independent over $\mathbb{R}$ and if $f(z + \omega_1) = f(z) = f(z + \omega_2)$ for all $z \in \mathbb{C}$. An elliptic function is a meromorphic doubly periodic function. For ω_1 and ω_2 fixed, the corresponding elliptic functions form a field. The parallelogram with vertices $0, \omega_1, \omega_2, \omega_1 + \omega_2$ containing the sides adjacent to the origin, but not containing the other two, is called the fundamental parallelogram Π. Being periodic with respect to the lattice L in the complex plane defined by the periods, one can think of elliptic functions as functions on the torus $\mathbb{C}/L$. This has not only the structure of a two-dimensional smooth manifold, but also an intrinsic structure of a Riemann surface, determined by the periods ω_1 and ω_2. Varying the periods you get a different complex structure on the same (on a topologically equivalent) underlying real torus. You associate sounds to elliptic functions with the same method as in every n-dimensional synthesis: you choose a closed curve on the torus and restrict the function to the curve to obtain a couple of periodic waveforms, the real and imaginary parts. It is tempting to think of the couple as a stereophonic sound, and try to create auditory images of complex analysis. Anyway you can enjoy the freedom to experiment with any algebraic (or even non-algebraic, for that matter) combination of the two. It is worthy to point out that, as opposed to the real case, you do not have a single sound associated to a function and a closed curve, but an infinite family. Of course, the shape of the chosen closed curve plays also a dominant role in the sonic results. To begin with, one can distinguish between two big families: shrinkable curves and non-shrinkable (curves which wind around the *holes* of the torus.) Within the former class, the meaningful distinction seems to be between curves which go around a pole (and how many times), and curves which do not. In the latter class the meaningful parameter is probably the number of tours the curve makes around a hole (winding number). The simplest family of curves of the second class are helices, obtained by projecting on the torus $\mathbb{C}/L$ straight lines in the complex planes. Extensive experimentations have been carried on, using the slope (which affects the winding number in a readable way) as a varying parameter. We refer again to the quoted site of *Musica Inaudita* for the records.

There is an entire zoology of famous elliptic functions, connected to each other by a variety of well-known (and less well-known) formulas. We refer to [5] and [6] for an extensive treatment of the subject. In this paper we give just one example, the Weierstrass $\wp$, defined as follows:

$$\wp(z) = \frac{1}{z^2} + \sum_{\omega \in L'} \left[\frac{1}{(z-\omega)^2} - \frac{1}{\omega^2} \right] \tag{3}$$

where the sum is taken over the set of all non-zero periods, denoted by L'. This series converges uniformly on compact sets not including the lattice points. What makes elliptic functions so interesting for audio synthesis is the fact, already recorded in the Introduction, that you can describe any of them in terms of some chosen collection of *special* functions. Electing, as we did, the Weierstrass function as our main character, the significant theorem becomes:

> **Theorem.** *Any even elliptic function is a rational function of $\wp$. Any elliptic function f can be written uniquely in the form*
>
> $$f(z) = g(\wp(z)) + \wp' h(\wp(z))$$
>
> *where g and h are rational functions.*

The existence of such a result permits a systematic exploration of the sound quality of different elliptic functions, exploiting the different possible formulas. To fix the ideas, one can study the *additive synthesis* formulas offered by the theorem, starting with $\wp$ and $\wp'$, and producing rational functions of increasing degree in $\wp$ and $\wp'$:

$$\begin{array}{lcl}
\wp(z) & + & \wp'(z)\wp(z) \\
\wp^2(z) & + & \wp'(z)\wp(z) \\
\wp^2(z) & + & \wp'(z)\wp^2(z) \\
\wp^2(z) & + & \wp'(z)(\wp^2(z) + \wp(z)) \\
\vdots & & \\
\wp^n(z) & + & \wp'(z)\frac{\wp^r(z)+\wp^{r-1}(z)+\cdots+\wp(z)+1}{\wp^s(z)+\wp^{s-1}(z)+\cdots+\wp(z)+1} \\
\vdots & &
\end{array}$$

3.2. Sound synthesis

Audio samples obtained using elliptic functions have been generated by CSound using the sampled values calculated by a Python script. For the computations, we used the following formula which expresses Weierstrass's $\wp$ as the sum of an infinite series

$$\begin{array}{rcl}
\wp(z) & = & \frac{\pi^2}{4\omega_1^2}\Big[-\frac{1}{3} + \sum_{k=-\infty}^{\infty} \csc^2 \frac{(z-2k\omega_2)}{2\omega_1}\pi \\
& & - \sum_{k=-\infty,k\neq 0}^{\infty} \csc^2 \frac{(k\omega_2)}{\omega_1}\pi\Big]
\end{array}$$

Thanks to the complex analysis Python libraries, all the computations have been performed without any need for external programs or resources. The same script has been used to first compute the values of the elliptic functions, at regular steps, on the given orbit, and then to produce a couple of score and orchestra files, processed by CSound and converted into an audio file.

The *revolution frequency* of the orbit has been chosen in the audio range, namely 440 *Hz*, to have a perceivable audio signal:

$$x = \cos(440\ 2\pi t), \quad y = \sin(440\ 2\pi t) \tag{4}$$

In Figures 1-5 we list some waveforms obtained from Weierstrass's $\wp$, using different closed orbits.

4. SYNTHESIS BY MEANS OF CONTINUOUS NOWHERE DIFFERENTIABLE FUNCTIONS AND LACUNARY SERIES

4.1. Introduction

In the previous Section we have presented a sound synthesis technique articulated in two phases: sample computing and sound realization (or better, conversion, since CSound works in this case as

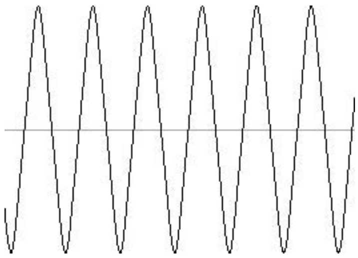

Figure 1: *Waveform of* $\Re\wp$ *calculated on the circular orbit* $x = \cos(440\ 2\pi t)$, $y = \sin(440\ 2\pi t)$

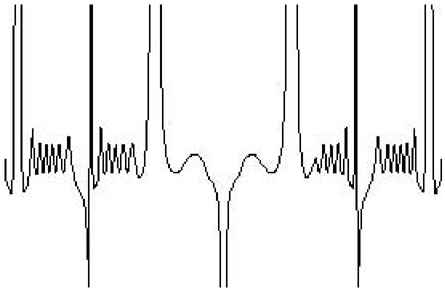

Figure 2: *Waveform from* $\Re\wp$ *calculated on the open orbit* $x = \sin(440\ 2\pi t)$, $y = t(1 + \cos(4402\pi t))$

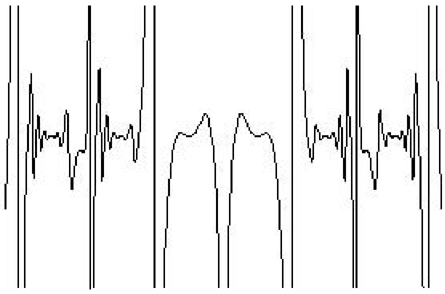

Figure 3: *Waveform from* $\Im\wp$ *calculated on the open orbit* $x = \sin(440\ 2\pi t)$, $y = t(1 + \cos(440\ 2\pi t))$

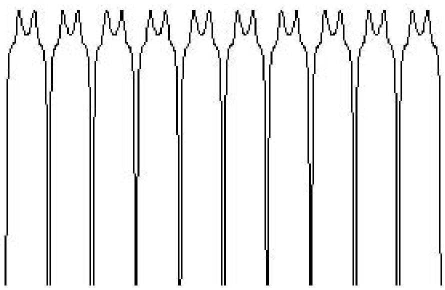

Figure 4: *Waveform from* $\Re\wp$ *calculated on the open orbit* $x = t$, $y = 3t$ *(helix)*

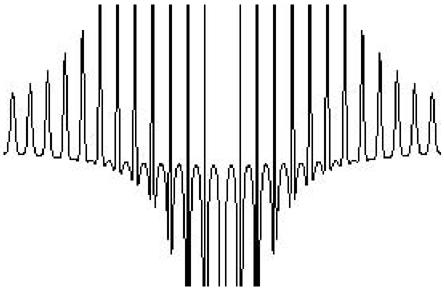

Figure 5: *Waveform from* $\Re\wp$ *calculated on the open orbit* $x = t$, $y = 50t$ *(helix)*

a digital-to-analog converter). For the sound generation procedure described in this Section we take full advantage of the capacities of CSound as an additive synthesizer. We again used Python scripts to prepare the Csound instruments, adding the right number of elementary oscillators (described by the `oscil` opcode) taking into account the frequency and amplitude relations. Once more, the main advantage in using this kind of organization is a high flexibility and the possibility to easily create a collection of CSound orchestras and scores, according to a certain principle (performing a systematic exploration of a parameter) and then run all the sound processing together afterwards.

4.2. The Riemann function

The Riemann function $R : \mathbb{R} \to \mathbb{R}$ is defined as follows (we refer to [7] for an accurate description of the world of nowhere differentiable functions):

$$R(x) = \sum_{k=1}^{\infty} \frac{1}{k^2} \sin(k^2 x) \tag{5}$$

The Riemann function R is continuous on all of $\mathbb{R}$ but differentiable only on a set of points of measure zero (*i.e.* R is non differentiable on a dense subset of $\mathbb{R}$.)

It has been proved, by Gerver [8] [9], Hardy [10] and Smith [11], that, despite the fact that R is almost nowhere differentiable, it has a finite derivative at points of the form:

$$x_0 = \pi\frac{2p+1}{2q+1}, \quad p, q \in \mathbb{Z} \quad R(x_0) = -\frac{1}{2} \tag{6}$$

It might be interesting to observe that the values of R can be explicitly computed at the numbers $x = p/q,\ p, q \in \mathbb{Z}$:

$$R\left(\frac{p}{q}\right) = \frac{\pi}{4q^2} \sum_{k=1}^{q-1} \frac{\sin\left(\frac{k^2 p\pi}{q}\right)}{\sin^2\left(\frac{k\pi}{2q}\right)} \tag{7}$$

4.3. The Weierstrass function

The Weierstrass function was the first published example of a continuous nowhere differentiable function (1875):

$$W(x) = \sum_{k=0}^{\infty} a^k \ \cos(b^k \pi x) \tag{8}$$

$0 < a < 1, \quad b > 1 \quad ab > 1 + 3\pi/2, \quad b$ an odd integer.

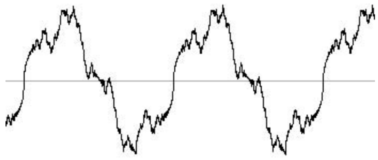

Figure 6: *Waveform obtained from Riemann's R function.*

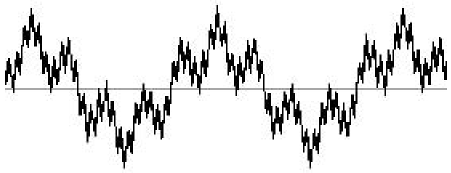

Figure 7: *Waveform obtained from Weierstrass's W ($a = 0.5$, $b = 5$).*

4.4. A graphical interface for lacunary series

Both Riemann's and Weierstrass's functions are examples of the large class of functions defined by convergent lacunary series. Let $\{n_k\}$ ($k \in \mathbb{N}$) be a strictly increasing sequence of natural numbers such that $n_{k+1}/n_k > q > 1$. The series

$$\sum_{k=1}^{\infty} a_{n_k} e^{i n_k t} \tag{9}$$

is called *lacunary*.

Under suitable conditions on coefficients and frequencies (Hardy conditions) the lacunary series converge rapidly, offering the possibility to easily obtain interesting sonic results even using a limited number of oscillators. In this Section we discuss an implementation of an interactive interface for sound synthesis by means of lacunary series using eight oscillators.

The formula which describes the waveforms obtainable with this software tool is

$$\sum_{n=1}^{8} a_n \sin(n^k 2\pi f t) \tag{10}$$

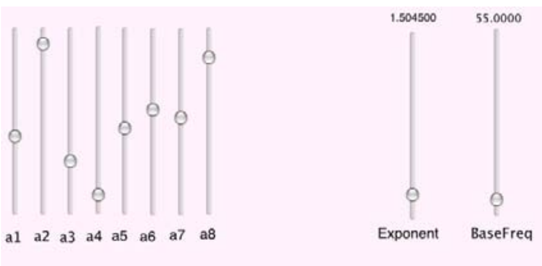

Figure 8: *CSound interface to explore the lacunary Fourier series with 8 oscillators.*

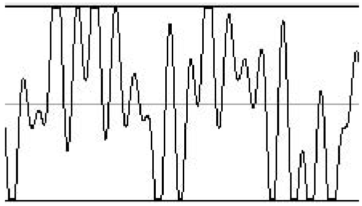

Figure 9: *Waveform corresponding to the lacunary Fourier series corresponding to the configuration of Figure 8.*

The interface has been developed using MacCsound [12] by Matt Ingalls. The eight sliders on the left act on the Fourier coefficients $a_1, \ldots, a_8$. The sliders on the right (labeled, respectively as *exponent* and *basefrequency*) act in a self-explanatory way. The choice $k = 1$ gives a full Fourier series. Choosing an integer > 1 gives rise to *lacunas* in the spectrum. One can experiment in real time the enrichment of the sound produced by the migration of the eight spectral components towards the high frequencies. Figure 8 shows a screenshot of the program while running and the Figure 9 displays the waveform obtained from the parameters corresponding to the slider configuration of Figure 8. A last word on the slider k. The slider moves continuously and therefore passes through every (ideally) real number, while the symbol k appearing in the formula stands for natural numbers. So, this little toy produces much more than sounds obtained by lacunary series. In fact the action obtained on a sound taking real exponents might be described as a nonlinear distortion, defined on a basis of the Hilbert space of functions with the chosen base frequency. This is related to the *regular transformations* defined by MacAdams [13]. To our advice, it seems a very interesting subject to investigate.

5. CONCLUSIONS

We have proposed two (as far as we know) new methods of synthesis, based on mathematical models. Our belief is that the parameters, which are significant for the synthesis are also meaningful from a pure-mathematical point of view. The hope is that, once the correspondence synthesis-mathematics established, the whole organizing power of mathematical theories can be applied to obtain an analogous conceptual frame by which interpreting sounds.

6. REFERENCES

[1] A. Borgonovo and G. Haus, "Musical sound synthesis by means of two-variable functions: experimental criteria and results," *Computer Music Journal,* vol. 10, no. 4, pp. 57-71, 1986.

[2] C. Roads, *The Computer Music Tutorial*, MIT Press, 1996.

[3] Y. Mitsuhashi, "Audio signal synthesis by functions of two variables," *Journal of the Audio Engineering Society,* vol. 30, no. 10, pp. 701-706, 1982.

[4] V. Cafagna and D. Vicinanza, "How to use musical perception in order to understand elliptic functions?" Cognitive and perceptual aspects of the computer-aided music analysis - MaMuX : Pré-séance du 7 mars 2003 - IRCAM - Paris.

[5] K. Chandrasekharan, *Elliptic Functions*, Springer-Verlag, 1985.

[6] J. Tannery and J. Molk, *Eléménts de la Théorie des Fonctions Elliptiques*, Paris, 1893-1902.

[7] J. Thim, *Continuous nowhere diffrentiable functions*, Master Thesis, Lulea University of Technology, 2003.

[8] J. Gerver, "The differentiability of the Riemann function at certain rational multiples of π," *Amer. J. Math,* vol. 92, pp. 33-55, 1970.

[9] J. Gerver, "More on the differentiability of the Riemann function," *Amer. J. Math,* vol. 93, pp. 33-41, 1971.

[10] G. H. Hardy, "Weierstrass's non-differentiable function," *Trans. Amer. Math. Soc.,* vol. 17, pp. 301-325, 1916.

[11] A. Smith , "The differentiability of the Riemann function," *Proc. Amer. Math. Soc.,* vol. 34, pp. 463-468, 1972.

[12] M. Ingalls, `http://www.csounds.com/matt`, MacCsound.

[13] S. McAdams, *Spectral Fusion, Spectral Parsing and the Formation of Auditory Images*, Ph.d Thesis, Stanford University, 1984.