This article presents the results of research on the generation of musical processes resulting from the application of Schrödinger's Equation for an atomic potential with radial symmetry, a mathematical model of dynamic systems that, like music, may change and develop in time. (For an introductory discussion of Schrödinger's Equation, see Feynman, Leighton, and Sands 1971.) I focus on the creation of musical material and its organization arising from the solutions and implications of the equation. This is applied to asynchronous granular techniques (e.g., Truax 1990b; De Poli, Piccialli, and Roads 1991; Di Scipio 1994), as these are ideally suited for stochastic processing of musical material and allow the manipulation of multidimensional timbral characteristics. Work carried out consisted of three main parts: the development of relevant theoretical principles and algorithms for the generation of music, their implementation as compositional software, and their realization as a composition that demonstrates their musical validity and viability.

Background

The relationship between scientific disciplines and music is not new in Western culture: suffice it to mention Boethius’s concept of “music of the spheres,” the medieval quadrivium, the relationship between talea and color in isorhythmic music of the 14th century, and rules of counterpoint regarding temporal proportions between voices in Fux’s Gradus ad Parnassum in the 18th century. During the 20th century, a renewed link between science and music began to emerge in composition, from the relatively straightforward principles of serialism to Schillinger’s compositional theory (Schillinger 1948), the complex constructs realized by Xenakis in works such as Achorripsis and Nomos Alpha (Xenakis 1992), the compositional strategies of Koenig (Koenig 1970a and 1970b; Laske 1981) and Hiller (Hiller 1981), and the use of stochastic processes by Truax (1982) and others (e.g., Jones 1981; Lyon 1995; Ross 1995; Hoffman 2000). Theoretical and technical advances in information technology propitiated further exploration in areas such as generative grammars (Holzman 1981; Beyls 1991), self-similar, chaotic, and more general dynamic systems (Bolognesi 1983; Dodge 1988; Pressing 1988; Truax 1990a; Di Scipio 1991; Bidlack 1992), and more general algorithmic approaches (Maia et al. 1999; McAlpine, Miranda, and Gerard 1999).

The aim of the present work is to investigate and experiment with the possibilities offered in this area by an equation that contributed to a deep change in our conception of the world, following this path to its ultimate conclusion (i.e., the musical composition). In this sense, this study presents an additional link between the thought processes that lead to the mathematical abstraction of reality (i.e., models of the physical world) and those involved in the creation of musical works. These processes mirror each other in the sense that the former construct meaning out of the observation of the physical world, and the latter transform meaning into a physical trace, namely, a piece of music.

The current project develops previous work on the derivation of organic structure and generation of material offered by Schrödinger’s Equation (Bain 1990; Fischman 2003), which preserves musical
logic and coherence. While previous research by the author was restricted to the derivation of structure in a composition for a single instrument (harp-sichord) and to the generation of pitch material, the goal of the present study was to derive a strategy for the creation of structure and musical material for a medium which allows greater complexity, includes parameters other than pitch, and provides the basis for the application of similar strategies to other media. Thus, the electroacoustic domain, with a specific focus on asynchronous granular techniques, was chosen, because it allows a significant degree of complexity, and it allows thorough investigation of the possibilities inherent in Schrödinger’s Equation for the manipulation of multidimensional timbral characteristics, with pitch included as one of these dimensions [Smalley 1986, 1997]. In particular, granular techniques are ideally suited for stochastic processing of musical material [Lorrain 1980; Xenakis 1992]. Furthermore, a work for tape can provide the immediate aural feedback necessary to evaluate the results.

The research methodology adopted during the project consisted of three main parts. First, I developed appropriate theoretical principles and algorithms for the creation of musical material arising from the solutions and implications of Schrödinger’s Equation. Next, I developed software that implements these algorithms. The software includes a graphical interface that enables the manipulation of large amounts of data and musical streams, is reasonably self-sufficient for the creation of a complete musical work, and is reasonably open-ended, offering possibilities for enhancement and customization via third-party plug-ins using a recognized standard. The third part of this research consisted of testing the validity and viability of the theoretical principles and the usefulness of the software through the creation of a musical work for tape: Erwin’s Playground.

Concerning the last point, it is important to stress that neither theory nor software are intended to guarantee satisfactory aesthetic results—or even musical coherence. I do believe that these are the responsibility of the individual who realizes their potential through the compositional process, a system described by Vaggione [2001] as an “action/perception feedback loop.” In fact, it is reasonable to assume that most mathematical constructs are neutral to musical logic and that their usefulness is dependent on the actual way they are mapped into musical processes by the composer. Nevertheless, theoretical principles are vital in providing “formal tools as generative and transformative devices” [Vaggione 2001] within which a specific work may be realized, projecting the latter beyond an exclusively sensory sphere. Furthermore, decisions regarding the correspondence between the model’s parameters and musical attributes, as well as their realization by means of the design and functionality of the software (which, at first sight, might seem to be exclusively related to theoretical and technical development) already belong to the compositional process, since they carry specific aesthetic presumptions and aims. (For a discussion of this issue in a more general context, see Vaughan 1994.) For instance, recent published work presents different ontological approaches to correspondences between quantum mechanics and music [Delatour 2000; Sturm 2001].

After a brief description of Schrödinger’s Equation, this article presents the framework developed for the correspondence between the equation and the creation of granular clouds and strategies for their organization into coherent musical structure. This will be followed by discussion regarding the realization of these principles in Erwin’s Playground, a work for computer-generated tape. Subsequently, a general description of the software will be provided. An appendix is offered to readers who may wish to explore the mathematics in more detail. (Reference to equations in the appendix uses the prefix “A” followed by a number.)

Schrödinger’s Equation

Schrödinger’s Differential Equation forms the basis for the techniques described herein, which establish a correspondence between the solutions of the equation and musical attributes. We will examine relevant aspects of these solutions, with a more detailed mathematical discussion featured in the appendix (equations A1–A8).
The particular version of Schrödinger’s Equation examined here results from consideration of the atom as a center that produces a potential. The latter is known as an atomic potential with radial symmetry, because, if we look at a spherical surface of radius $r$, the potential will be symmetric with respect to the center. Thus, Schrödinger’s Equation assumes the following form:

$$-\frac{\hbar^2}{2m} \nabla^2 \Psi + V(r)\Psi = i\hbar \frac{\partial \Psi}{\partial t} \quad (1)$$

where $V(r)$ is the atomic potential, $m$ is the mass of the electron, $r$ is the distance from the center of the potential (e.g., the nucleus of an atom) to the electron, and $\hbar$ is the ratio between Plank’s constant $\hbar$ and $2\pi$ ($\hbar = h/2\pi$).

The family of functions corresponding to the solutions of the equation is represented by the symbol $\Psi$, also known as the wave function. Examination of the general form of the wave function reveals that $\Psi$ consists of the product of four smaller functions (equation A2) that may be treated separately, because each is dependent on a different parameter. Three of these were found particularly useful: $R$, which is exclusively dependent on $r$, the distance of a point to the center of the potential; $\Theta$, which is exclusively dependent of the point’s elevation angle $\theta$; and $\Phi$, which is exclusively dependent on its horizontal angle $\phi$.

Furthermore, the solutions depend on a set of quantum numbers that restrict the physical attributes of the electron to multiples of certain energy levels. The first of these values is $n$, the principal quantum number ($n = 1, 2, 3, \ldots$), which determines the overall energy of the electron (equation A7). The second value is $l$, the angular momentum quantum number ($l = 0, 1, 2, \ldots, n-1$), which generates additional allowed energies (quantum states) within a single value of $n$. (The term shell is used to denote a particular value of $l$.) Thus, four types of shells have been encountered in nature and are named S, P, D, and F, corresponding to $l = 0, 1, 2, \text{ and } 3$, respectively. Therefore, a particular energy level is characterized by a combination of $n$ and $l$, which, by convention, is named using the value of the principal quantum number and the letter of its shell (e.g., $1S$, $2S$, and $2P$).

The third quantum number, $m$, is the magnetic quantum number ($-l \leq m \leq l$). Its values produce variations in the orientation of these shells, known as orbitals, but $m$ does not affect the overall energy in this model.

The physical meaning of an orbital is a region within which there is some probability of finding an electron. It is possible to show that this probability is associated with $R$ and $\Theta$ (equations A9 and A10). As an example of this correlation, Figure 1 shows plots of these distributions and the resulting orbital for particular values of $n$, $l$, and $m$, from which we may infer the most probable distances and orientations. For instance, Figure 1b indicates that the electron is more likely to be found in the vicinity of $35.27^\circ$ and $144.73^\circ$.

The quantum numbers were successfully used to explain the existence of common characteristics between known chemical elements, supporting their classification into the Periodic Table of Elements, originally devised by the chemist Dmitri Mendeleev in the 19th century but only more recently explained by Schrödinger’s Equation. Roughly speaking, elements that have the same number of outermost electrons (electrons with highest quantum energy) in the same type of shell share similar physical characteristics, belonging to the same group and occupying the same column in the periodic table. For instance, the group of halogens consists of elements that have five electrons in their P shell as follows: fluorine has five electrons in shell 2P, chlorine in 3P, bromine in 4P, iodine in 5P, and astatine in 6P.

**Generation of Granular Clouds**

My generation of musical material using Schrödinger’s Equation focuses on granular clouds (sounds composed of a large number of short elementary sonic particles, or grains, each lasting a few hundredths of a second) derived from statistical distributions obtained from the equation. A parallel is established between these clouds and the charge density surrounding the nucleus (electronic clouds). The algorithms affect a variety of time-varying attributes of the granular clouds. Special importance
is given to the creation of formal relationships between the inner structure of the source sound—particularly its amplitude envelope—and the structure of the cloud and its constituent grains. Source sounds are stored as wave audio files.

There are two main types of parameters: those affecting the cloud as a whole and those affecting each grain individually. These are now each discussed in turn.

### Cloud Parameters

These parameters refer to global attributes of the cloud, such as density, overall amplitude, and spatial and spectral trajectory.

#### Cloud Density

To determine the time-varying cloud density (number of grains per time unit), a correlation between density and the amplitude envelope of the source sound is established: the amplitude envelope drives the radial and elevation distributions, which, in turn, are proportional to the density fluctuation. In practice, the envelope is extracted from the source and stored in a time–amplitude breakpoint table. The cloud density is also stored as pairs of breakpoints. However, because the cloud duration can be different from that of the envelope, it is necessary to convert both times and amplitudes from the envelope table into corresponding values for cloud density [equation A11].

Currently, there are three different methods of establishing a correspondence between source amplitude envelope and cloud density. The first two calculate the density by using the envelope to drive the radial and angular distributions [equation A12], either by making both $r$ and $\theta$ directly proportional to the envelope [equation A13] or by making $r$ directly proportional to the envelope and $\theta$ directly proportional to its first derivative [equation A14]. The third method establishes a direct proportionality between the envelope and the density using a straightforward proportion rule [equation A15].
Cloud Amplitude Scaling

The amplitude of the cloud may be scaled by a value that fluctuates between a minimum and maximum gain. The fluctuation follows the envelope of the source (equation A16).

Cloud Motion

Motion is carried out in three dimensions. Two of these are spatial [width and depth], and the remaining one is articulated as a spectral dimension [pitch shift].

Width and Depth

These are calculated within a stereo image: \( x \) indicates width, and \( y \) indicates depth, as illustrated in Figure 2. [See also equation A17.]

Spectral Motion

The cloud “moves” in the frequency domain as a result of time-varying transposition within limits specified by the composer, establishing an analogy to movement across a height axis \( z \). Transposition is carried out in the frequency domain via Short-Time Fourier Transforms, thereby preserving the duration of the grains.

Trajectory Creation

To implement motion, a three-dimensional trajectory is created that is analogous to physical movement within the boundaries of a shell [orbital]. This consists of the stochastic generation of breakpoints according to Schrödinger’s radial and angular distributions. The breakpoints consist of position-time pairs within the duration of the cloud, converted from \( r, \theta, \) and \( \phi \) into width, depth, and height (equation A18).

The algorithm that generates the trajectory normalizes the values so that the cloud reaches the maximum width, maximum depth, and one of the transposition boundaries. (These are not reached simultaneously.) Also, to generate the trajectory breakpoints, the algorithm requires minimum and maximum values for the time interval provided by the composer. The intervals between the breakpoints in the trajectory will fall randomly between these two values. For instance, if the minimum and maximum intervals are 0.2 and 0.5 sec, respectively, and the last breakpoint was generated at time \( t_c = 1.4 \) sec, the time \( t_c \) for the next breakpoint will fall between 1.6 and 1.9 sec.

Finally, trajectories may be repeated an arbitrary number of times so that the sum of the durations of the repetitions is the overall duration of the cloud. (If required, this may be equal to the value of the orbital quantum number \( l \) plus one.) Repetitions may be identical or fluctuate within a percentage given by the composer.

Cloud Volume (Overall Grain Scatter)

The volume of the cloud depends on the maximum scatter of the grains in each dimension [width, depth, and pitch shift]. As a simple illustration, assume that the cloud positioned at the center of the speakers and does not move. In this case, low scatter values will produce a very narrow cloud around the center of the speakers. On the other hand, high scatter values will spread the grains from left to right, producing a wide cloud. This is illustrated in Figure 3.
Width and depth scatter values vary from 0 [no scatter] to any positive numbers \(x\) and \(y\), with a value of 1 indicating a scatter distance equal to the distance from the center to one of the speakers.

Spectral scatter values vary from 0 [no scatter] to any positive number of semitones (or any frequency ratio).

To implement scatter, a time-varying volume function is created in a similar manner to that of the motion trajectory described above. Breakpoints are also generated stochastically according to Schrödinger’s radial and angular distributions (equation A19).

Therefore, a grain created at time \(t_c\) will move with the cloud at a fixed “distance” from its center. This distance is generated stochastically within width, depth, and height boundary values given by the composer. However, to comply with the representation of depth \((y > 0)\), the resulting instantaneous depth of the grain as it moves with the cloud is always kept positive, which means that the grain may change its distance with respect to the center of the cloud.

Spectral scatter may be anchored or unanchored. In the former case, the sign of the current scatter value indicates whether the grains will be positioned to the right or left of [or transposed above or below] the cloud. For instance, when the scatter value is positive, all the grains are positioned to the right of the cloud. When the scatter becomes negative, the grains are positioned to the left. Unanchored scatter means that the grains are positioned both to the right and left of [and transposed above and below] the cloud.

The algorithm that generates the volume function normalizes the values so that the cloud reaches the maximum width, depth, and transpositional scatter. (Again, these are not reached simultaneously.) In a similar fashion to the motion trajectory, the algorithm also requires the composer to provide minimum and maximum values for the time interval to generate time-volume breakpoints. The volume function may also be repeated an arbitrary number of times; this may also be equal to the value of the orbital quantum number \(l\) plus one. Repetitions may be identical or fluctuate within a percentage given by the composer.

**Grain Parameters**

These parameters control the attributes of individual grains, such as their envelope, duration, and individual scatter patterns.

**Grain Amplitude Scaling**

The amplitude of individual grains is scaled by a gain factor according to their onset times relative to the beginning of the cloud and implemented according to various algorithm types. The first option consists of a scaling function that follows the envelope of the source (equation A20). Other options follow Schrödinger’s distribution \(|r^2 \sin(\theta)|\Psi(r,\theta,\phi)^2\) in equations A21–A23), the complement of Schrödinger’s distribution (equation A24), and Schrödinger’s cumulative probability and its complement (equations A25 and A26). It is worth noting that if the cumulative probability is followed, amplitude scaling will assume its minimum value at the beginning of the cloud and increase towards its maximum value at the end of the cloud, producing a nonlinear crescendo. This is because the cumulative probability always increases monotonically. Similarly, following the complement of the cumulative probability will result in a nonlinear diminuendo.

Finally, scaling may be generated from the various distributions resulting from Schrödinger’s Equation (equations A27–A33).

**Grain Duration**

The actual duration of individual grains fluctuates according to their onset times relative to the begin-
ning of the cloud. The function controlling the duration fluctuation is calculated in a manner similar to that of the scaling factor, offering the same options to the user (equations A34 and A35). However, in addition to the options available for amplitude scaling, grain duration allows the repetition of the duration function similar to that of the cloud’s motion trajectory and its volume function: repetitions may also be identical or fluctuate within a percentage given by the composer. The main algorithms available consist of the following schemes: duration proportional to the source envelope, duration proportional to Schrödinger’s distribution and its complement, duration proportional to Schrödinger’s cumulative probability and its complement, and duration generated randomly from various distributions.

It is worth noting that because the cumulative probability always increases monotonically, grains have minimum duration at the beginning of the cloud, becoming gradually longer until their duration reaches its maximum value at the end of the cloud. The inverse of the cumulative probability produces the opposite effect, namely, grains become gradually shorter until their duration reaches its minimum value at the end of the cloud.

**Grain Envelope**

The envelope applied to each individual grain consists of either the envelope extracted from the source or a linear envelope shaped according to the seven parameters shown in Figure 4 and some of Schrödinger’s distributions (equations A36–A40).

**Grain Scatter Path**

In addition to the fixed scatter resulting from the volume of the cloud, each grain is provided with an individual trajectory. Grain scatter is given in the same units discussed above in relation to cloud volume.

The algorithm is implemented by means of a time-varying scatter function created for each dimension, with normalized duration and maximum value of unity (equations A41–A43). To generate the trajectory breakpoints, the algorithm requires minimum and maximum values for the time interval between a pair of breakpoints provided by the composer. However, these are given as percentages of the grain duration, as the function is normalized. When a grain is created, the function is scaled to fit its duration and maximum amplitude. It is possible to change the shape of the function according to a jitter value given by the composer, which results in a different path for each grain.

Finally, Doppler Shift is applied to the overall motion of the grain, which is the resultant of the following components: individual scatter path, individual scatter position with respect to the cloud, and overall cloud motion (equation A44). The pitch shift relative to the listener is invariant under rotation. Therefore, it is possible to simplify the calculation of the Doppler Effect if we rotate the coordinate system by an angle so that the line joining the speakers is kept parallel to the instantaneous velocity of the grain, as shown in Figure 5 (equations A45–A49).

**Structure**

We will now proceed to examine ways in which the generated material can be structured into a musical work. It is worth noting that these are particular choices out of a multitude of possibilities and that these choices, in turn, will have repercussions on the generation of materials (selection of source sounds, cloud and grain characteristics, etc.). Nevertheless, it is also important to note that choices were made with full consideration and regard for their musical viability and formal integrity.
Pyramidal Shape

A relatively straightforward strategy for structuring musical works may be linked to a survey through the atomic shells resulting from changes in the overall energy. For instance, the survey may begin in the lowest energy shell and leap to higher shells as this energy increases. (This case also corresponds to an excursion through the periodic table according to ascending atomic number.) In fact, it is worth adopting this particular instance as a starting point to develop a basic model that can subsequently be modified and extended.

Following the analogy with the periodic table, major sectional boundaries may correspond to the reappearance of a shell with an increasing value of \( n \) (which also corresponds to a change of row in the periodic table). Because \( S \) is the first shell, major sectional boundaries will start at \( 1S, 2S, 3S, \ldots, 7S \). Subsections within these boundaries may correspond to the shells surveyed in between. Figure 6 illustrates the resulting structure obtained from the energy levels in the periodic table; this may be compared to a pyramid that is surveyed from top to bottom. Examination of this structure brings us to several observations.

First, sections become longer by virtue of the number of subsections they include. Thus, section 1 contains only one shell \( [S] \), sections 2 and 3 contain two shells \( [S, P] \), sections 4 and 5 contain three shells \( [S, P, D] \), and sections 6 and 7 contain four shells \( [S, P, D, F] \). Second, it is possible to identify four independent threads of development corresponding to the reappearance of each particular shell in subsequent sections. For instance, shell \( D \) reappears in sections 4 to 7. This suggests the possibility of creating and additional developmental dimension that, following the analogy with the pyramid, would consist of four “vertical” paths identified by different grayscale shades in Figure 6. For instance, the reappearance of a shell could develop the material of previous appearances.

Two remaining issues must be determined before the structure is finalized: section duration and section differentiation.
Section Duration

We can follow the path set by the first observation above [that sections become longer by virtue of the number of subsections they include] and establish a further principle of growth related to the periodic table: the minimum duration of each subsection cloud is proportional to the sum of atomic numbers of its related shell by setting a correspondence between a chosen time unit and atomic numbers as follows:

\[
\text{Section Duration} = \left( \sum \text{atomic_numbers} \right) \times \text{time unit} \quad (2)
\]

For example, if the time unit is 0.05 sec, the minimum duration of the subsection corresponding to shell 2P [atomic numbers 5 to 10] will be

\[
(5 + 6 + 7 + 8 + 9 + 10) \times 0.05 = 2.25 \text{ sec}
\]

Applying this principle, we obtain a further refinement of the pyramidal shape, whereby a new appearance of a shell is normally longer than its previous appearances. Table 1 shows the minimum duration values for shell D with atomic unit duration of 0.0342 sec, and Figure 7 shows a schematic version of the refined pyramid.

To allow some freedom when taking into account musical context, only the minimum duration of each subsection is set. This is not inconsistent with formal assumptions because, according to quantum mechanics, an electron remains in a given state as long as the energy pumped into (or extracted from) it is equal to or greater than the transition energy required to move to an immediate neighboring state, as calculated by equation A8.

Section Differentiation

Identification of distinctive threads according to shells requires that sections be differentiated by their content. This can be achieved by using different source sounds in each shell. This will also influence overall cloud and individual grain characteristics because, as explained above, many of these are derived from the attributes—particularly the envelope—of the source sound. Another way sections may be differentiated is by setting different grain attributes [e.g., different durations, envelopes, spatial locations, or by using different distributions to generate amplitudes and durations].

Extensions of the Pyramidal Shape

Two other structural shapes can be derived from the pyramid: an inverted pyramid and a diamond. The former is analogous to a survey of the periodic table from the highest to the lowest atomic number. The latter consists of an original pyramid fol-

Table 1. Minimum duration values for shell D, with atomic unit duration of 0.0342 sec

<table>
<thead>
<tr>
<th>Shell</th>
<th>Duration (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D</td>
<td>8.7210</td>
</tr>
<tr>
<td>4D</td>
<td>14.8770</td>
</tr>
<tr>
<td>5D</td>
<td>25.8210</td>
</tr>
<tr>
<td>6D</td>
<td>36.7650</td>
</tr>
</tbody>
</table>

Fischman
Figure 8. (a) Inverted pyramidal structure and (b) diamond structure.

Other Structural Possibilities

Apart from the simple trajectories related to ascending and descending atomic number order, we can generate more complex trajectories. For instance, shells, or even subdivisions of shells corresponding to individual atomic numbers, can be ordered according to stochastic algorithms or generative rules. Alternatively, they may follow attributes of a source sound such as the envelope where the peak may correspond to the maximum atomic number. All these are options beyond the scope of the current research plan, nevertheless, they are well worth exploring.

Another issue that may be examined from a different perspective is that of section duration. For instance, instead of establishing a correlation between section duration and atomic number, one may adopt a principle of constant rate of energy accumulation (or dissipation) expressed as a fraction of the ionization energy $E_a$ per second. As an illustration, assuming that $K_e = 0.01 E_a$ and that shell 3D progresses to shell 4P, then the duration of the former is

$$3D_{\text{SectionDuration}} = \frac{abs\left(\frac{1}{n_a^2} - \frac{1}{n_b^2}\right) \times E_a}{K_e}$$

Equation (3)

$$= \frac{abs\left(\frac{1}{3^2} - \frac{1}{4^2}\right) \times E_a}{0.01 E_a} = 4.86 \text{ sec}$$

The problem with this method is that energy transitions at lower shells are significantly higher than those at higher shells. For instance, the transition from 1S to 2S requires an energy value over one hundred times higher than a transition from 6D to 7P. Consequently, the duration of 1S is in the same proportion to that of 6D, yielding extremely long durations for 1S or extremely short ones for 6D. This may be overcome by subdividing longer sections according to proportions similar to those dividing the whole work. Alternatively, the rate of energy flow could correspond to a nonlinear function, being faster for lower shells and slower for higher ones. Another possible strategy may consist of generative or stochastic algorithms to regulate the rate of energy flow. In general, equation A10 may be modified to cater for any energy flow function $K_e(t)$:

$$\int_0^{\text{SectionDuration}} K_e(t) dt = \frac{abs\left(\frac{1}{n_a^2} - \frac{1}{n_b^2}\right) \times E_a}{K_e}$$

Equation (4)

Erwin’s Playground

Erwin’s Playground is a work for tape lasting 9 min 12 sec. Except for the recording of the samples and basic noise and click removal, it was created exclusively with AL and ERWIN software (Fischman 2003), which was developed to implement the principles described in this article. As a component of the current project, Erwin’s Playground is intended to demonstrate the validity and viability of these theoretical principles in the generation of a musical
work as well as the usefulness of the software as a conduit for the realization of these principles. However, its aesthetic value is intended to be independent of the generating principles, because no theory is capable of guaranteeing satisfactory aesthetic results or musical coherence. These are of course the responsibility of the composer.

The name of this work is an allusion to Erwin Schrödinger’s imaginary field of action, namely, the inner shells of the atom. Its structure is modeled on the diamond shape described above and linked to a survey that begins in the lowest energy state, leaps up until it reaches the highest shell, and then returns back to the initial state. This may also be compared to an excursion through the periodic table that ascends and then descends according to atomic number order.

Four different types of sound sources were assigned to each shell to achieve section differentiation. The S shells consisted primarily of vocal sounds (audio files MonoC1, Uba, UlulowL). P shells were assigned three different types of sound, which, nevertheless, share some common spectral characteristics: the “buzz” of a fly, a vocal “buzz,” and the sound of a sparking welder (audio files DieFly, BZZZ, Welder).

D shells were assigned sounds that ranged from pitched to “squeaky.” The pitched sounds were the result of rubbing the edge of a glass cup with different amounts of liquid to obtain different pitches (audio files Crystal E♭, Crystal F♯, Crystal G♯). Squeaks were recorded from the un-oiled wheels of a trolley (audio file Squeak1). A sample containing a transition from pitch to squeak was achieved by rubbing the glass cup with different degrees of pressure (audio file Crystal B♭ to Squeak). Finally, a recording of the pitched sound of cicadas was also included toward the middle of the piece (audio file Cicadas).

F shells were normally associated with local and global climaxes; therefore, they spread through a wider bandwidth by means of three spectral types. The first type consists of two noise-based sounds obtained from a busy motorway and sea waves gurgling through rocks (audio files Motorway1 and Waves1). The second sound is metallic, consisting of the recording of a frying pan that was hit or rubbed against other objects (audio file PBg1f). The third was mainly used to create low-frequency textures and drones from the recording of a piece of plywood that was bent at a more or less regular rate (audio file PBdLow). (Examples of the source files, as well as passages characterizing each shells and the complete work, are available on the forthcoming Computer Music Journal volume 27 sound example disc.)

Grain duration was another significant attribute used to characterize the various shells. Finally, the atomic number duration equivalent used in the piece was 0.0342 sec per atomic number.

Software Implementation

The software implementation of this procedure consists of two separate components: AL, a compositional environment for the creation and manipulation of musical events, and ERWIN, a plug-in that implements the application of Schrödinger’s model described above. These were developed and tested using Visual C++ 6.0 running under Windows 2000 and later under Windows XP. Both applications use Microsoft DirectX 8 features and require that DirectX 8 be installed in the host computer. For this reason, the AL and ERWIN installation program also installs DirectX 8, if required. Currently, AL and ERWIN are distributed as free software under the GNU public license and are
### Table 2. Structural hierarchy list of Erwin’s Playground ("X" denotes section extensions)

<table>
<thead>
<tr>
<th>Section</th>
<th>Shell</th>
<th>Start Time</th>
<th>Duration (sec)</th>
<th>Grain Duration (msec)</th>
<th>Source Audio Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1S</td>
<td>0:00.0000</td>
<td>0.1026</td>
<td>5–25</td>
<td>MonoC1</td>
</tr>
<tr>
<td></td>
<td>X</td>
<td>0:01.026</td>
<td>3.7258</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2S</td>
<td>0:03.7384</td>
<td>0.2394</td>
<td>5–50</td>
<td>DieFly</td>
</tr>
<tr>
<td>2P</td>
<td></td>
<td>0:03.9778</td>
<td>1.5390</td>
<td>5–50</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X</td>
<td>0:05.5168</td>
<td>2.1572</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3S</td>
<td>0:07.6740</td>
<td>0.7866</td>
<td>5–50</td>
<td></td>
</tr>
<tr>
<td>3P</td>
<td></td>
<td>0:08.4606</td>
<td>3.1806</td>
<td>7.5–75, 7.5–20</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X</td>
<td>0:08.6566</td>
<td>3.1777</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4S</td>
<td>0:11.6412</td>
<td>1.3338</td>
<td>5–100</td>
<td>DieFly</td>
</tr>
<tr>
<td>3D</td>
<td></td>
<td>0:12.9750</td>
<td>8.7210</td>
<td>10–60, 30–60</td>
<td>Crystal E♭</td>
</tr>
<tr>
<td>4P</td>
<td></td>
<td>0:21.6960</td>
<td>6.8742</td>
<td>5–60</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X</td>
<td>0:28.5702</td>
<td>3.1777</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5S</td>
<td>0:31.7479</td>
<td>2.5650</td>
<td>5–25</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0:33.3091</td>
<td>10–160</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4D</td>
<td></td>
<td>0:34.3129</td>
<td>14.8770</td>
<td>10–150, 100–150</td>
<td>Crystal E♭</td>
</tr>
<tr>
<td>5P</td>
<td></td>
<td>0:49.1899</td>
<td>10.5678</td>
<td>10–60, 30–60</td>
<td>DieFly</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0:50.7577</td>
<td>0.5818</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>6S</td>
<td>1:00.3395</td>
<td>3.7962</td>
<td>5–25</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1:01.0357</td>
<td>2.9595</td>
<td>10–160</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1:04.0357</td>
<td>100–500</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1:07.0952</td>
<td>30.4038</td>
<td>1000–4000, 2000–6000</td>
<td>Waves1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1:12.6856</td>
<td>100–500</td>
<td>1000–4000, 2000–6000</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1:14.1856</td>
<td>100–500</td>
<td>1000–4000, 2000–6000</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1:19.2592</td>
<td>2000–6000</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1:28.9052</td>
<td>36.7650</td>
<td>200–350, 10–500</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1:30.1852</td>
<td>100–500</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1:35.1852</td>
<td>100–500</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2:05.9799</td>
<td>0.8494</td>
<td>30–60</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2:06.8293</td>
<td>17.1342</td>
<td>20–80, 20–150 Welder</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2:08.6923</td>
<td>17.1342</td>
<td>20–80, 20–150 Welder</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2:37.9635</td>
<td>5.0250</td>
<td>5–1000</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>7S</td>
<td>2:28.9885</td>
<td>5.9850</td>
<td>5–25</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2:30.0885</td>
<td>200–700</td>
<td>10–200</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2:34.9735</td>
<td>4.3442</td>
<td>5–25, 200–700</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2:40.5131</td>
<td>10.5110</td>
<td>200–350, 10–500, 10–4000, 20–150, 10–50, 30–50</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3:35.5541</td>
<td>36.7650</td>
<td>200–350, 10–500, 10–4000, 20–150, 10–50, 30–50</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3:43.0662</td>
<td>1.2090</td>
<td>7.5–20, 7.5–75, 10–60, 30–60</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3:49.0662</td>
<td>1.2090</td>
<td>7.5–20, 7.5–75, 10–60, 30–60</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>7P</td>
<td>4:40.2752</td>
<td>7.5–20, 20–40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6D</td>
<td></td>
<td>5:03.9758</td>
<td>36.7650</td>
<td>7.5–20, 20–40</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:05.9758</td>
<td>36.7650</td>
<td>7.5–20, 20–40</td>
<td></td>
</tr>
</tbody>
</table>

[continued]
### Table 2. (Continued)

<table>
<thead>
<tr>
<th>Section</th>
<th>Shell</th>
<th>Start Time</th>
<th>Duration (sec)</th>
<th>Grain Duration (msec)</th>
<th>Source Audio Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>7S</td>
<td></td>
<td>6:26.4662</td>
<td>5.9850</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>6P</td>
<td>6:32.4512</td>
<td>17.1342</td>
<td>30–60 20–80, 100–250 5–1000</td>
<td>DieFly Welder BZZZ</td>
</tr>
<tr>
<td>5D</td>
<td></td>
<td>6:49.5854</td>
<td>25.8210</td>
<td>200–350, 1000–4500 20–150 10–50 50–250</td>
<td>Crystal E♭ Crystal F♯ Crystal G♯ Squeak1</td>
</tr>
<tr>
<td>4F</td>
<td></td>
<td>7:15.4064</td>
<td>30.4038</td>
<td>100–5000 1000–4000 100–300, 4000–8000 100–10000, 150–200</td>
<td>Waves1 PBdLow PBg1f Motorway1</td>
</tr>
<tr>
<td>X</td>
<td>6S</td>
<td>7:45.8102</td>
<td>1.0000</td>
<td>5–25 10–160, 10–200 100–500</td>
<td>MonoC1 Uba UlulowL</td>
</tr>
<tr>
<td>10</td>
<td>5P</td>
<td>7:50.6064</td>
<td>10.5678</td>
<td>10–60, 30–60 30–60 5–1000, 10–60</td>
<td>DieFly Welder BZZZ</td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>8:01.1742</td>
<td>3.1552</td>
<td>200–350, 100–150 20–60, 50–250 50–150</td>
<td>Crystal E♭ Squeak1</td>
</tr>
<tr>
<td>4D</td>
<td></td>
<td>8:04.3294</td>
<td>14.8770</td>
<td>80–450, 200–350 40–400, 50–250, 100–350</td>
<td>Crystal B♭ to Squeak MonoC1 Uba</td>
</tr>
<tr>
<td>5S</td>
<td></td>
<td>8:19.2064</td>
<td>2.5650</td>
<td>5–25 10–160</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>8:21.7714</td>
<td>8.0546</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>4P</td>
<td>8:29.8260</td>
<td>6.8742</td>
<td>5–60, 10–30 5–60 5–1000</td>
<td>DieFly BZZZ</td>
</tr>
<tr>
<td>3D</td>
<td></td>
<td>8:36.7002</td>
<td>8.7210</td>
<td>80–450, 200–350 40–400, 50–250, 100–350</td>
<td>Crystal E♭ Squeak1</td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>8:45.4212</td>
<td>0.0268</td>
<td>5–25, 5–100</td>
<td>MonoC1</td>
</tr>
<tr>
<td>4S</td>
<td></td>
<td>8:45.4480</td>
<td>1.3338</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>8:46.7818</td>
<td>3.6696</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>3P</td>
<td>8:50.4514</td>
<td>3.1806</td>
<td>7.5–20, 7.5–75 5–25, 5–50</td>
<td>DieFly MonoC1</td>
</tr>
<tr>
<td>3S</td>
<td></td>
<td>8:53.6320</td>
<td>0.7866</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>8:54.4186</td>
<td>4.6046</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>2P</td>
<td>8:59.0232</td>
<td>1.5390</td>
<td>5–50 5–50 15–50</td>
<td>DieFly MonoC1</td>
</tr>
<tr>
<td>2S</td>
<td></td>
<td>9:00.5622</td>
<td>0.2394</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>9:00.8016</td>
<td>4.5972</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>1S</td>
<td>9:05.3988</td>
<td>0.1026</td>
<td>5–25 10–200, 70–200</td>
<td>MonoC1 Uba</td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>9:05.5014</td>
<td>6.1339</td>
<td></td>
<td></td>
</tr>
<tr>
<td>END</td>
<td></td>
<td>9:11.6353</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
available online at www.keele.ac.uk/depts/mu/staff/Al/Al_software.htm. The programs are also distributed with the Composer’s Desktop Project.

**AL**

AL is a multiple document interface (MDI) application, with frame windows that present views of different documents and/or several views of the same document. It is designed for the creation and manipulation of sonic events and their organization into a structured musical work.

To aid user familiarity with the software and promote a reasonably short learning curve, care was taken to develop an interface within an accepted general standard for Windows applications, including a menu bar, a tool bar, common window components, and typical mouse “drag-and-drop” operations. It also contains detailed online help pages, including a tutorial. Figure 9 shows an instance of the user interface.

Another important consideration in the design of AL was the desire to provide a reasonably open system with the option for development of third-party plug-ins, thus avoiding the constraints imposed by a single type of musical process. Future developments by the author and others will ideally implement a variety of compositional approaches in addition to the application of Schrödinger’s Equation in the existing plug-in (see the description of ERWIN below).
Documents and Views

AL documents store the information necessary to define and characterize the composed musical work. They consist of a set of ordered events, each with its own data and attributes, as well as general data such as creation time, session number, overall duration, sampling rate, etc. When a document is viewed, events are represented by rectangles on the screen, as shown in Figure 9. These rectangles may be dragged and stretched with the mouse.

Events may be empty or they may represent an actual audio file. The audio file can be a pre-existing file or the result of a plug-in operation, in which case the event stores the parameters exported by the plug-in. Events associated with an audio file adopt its name and are given a three-dimensional edge in the graphical user interface. Events may be locked so that they do not change their onset times when dragged, allowing only vertical movement. It is also possible to mute events so that they are not performed during playback and not included in any mix.

Main Features

In addition to standard Windows features (such as saving and loading events and their generating parameters, zooming, scrolling, undo, redo, cutting, copying, and pasting), a number of operations can be performed on an event by clicking on its rectangle with the right mouse button.

It is possible to classify the items in this menu according to four categories: playback; modification of event status, including properties such as onset time, color, locking, and muting; simple audio file operations, including association with an existing audio file, gain, fade, and channel swapping of stereo events; and finally plug-ins consisting of external processes, which create an audio file and associate it with the event. Users can also audition an audio mix of events or bounce the latter to an audio file.

ERWIN Plug-in

This application is provided as a separate dynamic-link library (DLL) conforming to the Microsoft Component Object Model (COM) standard. It may be called by AL through the right-button event menu to create a granular cloud according to the algorithms described above. The source sound for granulation is read from a wave file, and the cloud is stored in an output wave file that AL associates with the calling event.

The user can supply the necessary parameters via property pages, which fall into four categories. The first category consists of input/output and general settings, including the name of the input file, the name of the output file, and parameters for envelope extraction, grain extraction, cloud density calculation, Doppler Shift, and FFT operations. The second category is section attributes, consisting of the values of quantum numbers, atomic number, and the minimum and maximum radii for a particular atomic shell. The third category includes cloud attributes, that is, parameters for duration, grain density, amplitude scaling, motion, and volume. Finally, grain attributes consists of parameters for amplitude fluctuation, duration fluctuation, envelope, and scatter.

Except for the input and output names, the property pages present a set of defaults. The default for cloud duration is taken from that of the calling event.

ERWIN’s graphical interface presents the results of the process as shown in Figure 10, including the following data: plots of the distributions and their maxima and minima; a plot of the extracted input envelope, the location and value of its maximum, and the analysis window duration; plots of cloud attributes including density, motion, and scatter; and plots of the grain attributes, including the envelope and motion functions (“archetypes”).

Future Software Development

As with any piece of software, AL could be developed almost ad infinitum, and a variety of plug-ins could be created to enhance its capabilities. We will therefore focus on some of the developments that are relevant to the aesthetic path chosen here.

An immediate improvement might consist of al-
algorithmic automation of structure. For instance, AL could include a menu option for plug-ins that create structural hierarchies that could then be displayed in the document view. Specific colors could be assigned to events in different sections, and views could display dividing lines and section names. Furthermore, these algorithms could implement structural hierarchies in addition to those described above, where it has already been suggested that it may be possible to generate more complex trajectories with random or generative algorithms or by following the attributes of a source sound.

The generation of events could also be automated using, for instance, random algorithms based on Schrödinger’s distribution or other distributions for the determination of the number of events, their onsets, and their durations. Finally, it might be possible to develop software that applies the same principles to visual grains, integrating sounds with corresponding “animated” clouds seen on a screen or in a virtual reality environment.

**Conclusion**

A work of art may be articulated at various levels. At one level, articulation takes place through a medium that carries with it a set of principles resulting from its physical properties. For instance, the use of stone in sculpture leads to an organization of space ruled by visual proportion, texture, and the physical constraints resulting from gravitational forces [such as those that determine whether a structure may collapse or not].

A second level from which properties and functionality are derived is more specific to particular cultures, periods, or genres. In the case of most music, it depends on the particular aspects of sound a piece articulates. For instance, Western music in the 18th and 19th centuries is structured according to harmonic relationships between keys [e.g., fugue and sonata forms].

One may find a third level relevant to music that consists of an extra-musical dimension and corre-
sponding to the concepts of intrinsic and extrinsic referral discussed by Nattiez (1990). In this case, an additional set of principles operates by establishing a relationship between musical articulation and non-musical concepts, or, in other words, by mapping extra-musical models to the generation of musical discourse and structure. This is typical, but not exclusive, of programmatic music. As an example of work that is articulated at an extra-musical level without becoming programmatic, we may cite Xenakis’s use of parabolic trajectories for the design of string glissandi in *Metastasis*. The principles described in this article, and their realization in Erwin’s *Playground*, fall within this category.

It is hoped that the realization of the extra-musical dimension proposed here is reasonably faithful to the essence of a quantum mechanical model and the non-deterministic view of the universe emanating from the model, while allowing the creation of musically coherent works. Nevertheless, it is vital to point out that mapping the equations to musical parameters presented here is neither unique nor exhaustive. It is only one of a multitude of alternatives, which, to the author, makes musical sense. Furthermore, the model developed here can still be enhanced and explored, for example with improvements to the construction of structural hierarchies other than those presented above and the algorithmic generation of larger structural units, such as the number of events in a section and their individual attributes (e.g., onsets, durations, and granular parameters).

Also, the specific direction assumed during this research may lead to other paths. One of the most obvious consists of the application of the theoretical model to other media, including animation and virtual reality. The possibility of applying other mathematical models (e.g., those describing relativistic phenomena, heat convection, behavior of fluids, etc.) or alternative interpretations of Schrödinger’s Equation may be worth investigating.

Finally, I hope that *Erwin’s Playground* is able to contribute in some constructive way to the already considerable body of compositions deriving from mathematical logic and that the principles developed here—and their software implementations—may be of use to other composers, resulting in other musical works.

Acknowledgments

I am grateful to the Arts and Humanities Research Board (AHRB), UK, for its support in the form of a Research Leave Award, which made possible the realization of this project. Special thanks to Dalit, my partner in life, for her patience and moral support. Many thanks to my colleagues in the Keele University Music Department for their effective coverage of my duties during the period of research leave. Also, thanks to Richard Dobson and Archer Endrich of the Composers’ Desktop Project for their technical advice.

References


### Appendix

#### Schroedinger’s Equation for an Atomic Potential with Radial Symmetry

\[ -\frac{\hbar^2}{2m} \nabla^2 \Psi + V(r)\Psi = i\hbar \frac{\partial \Psi}{\partial t} \]  \hspace{1cm} (A1)

Using spherical coordinates,

\[ \Psi(r,\theta,\phi,t) = R(r) \cdot \Theta(\theta) \cdot \Phi(\phi) \cdot T(t) \]  \hspace{1cm} (A2)

where

- \( r \) is the distance of a point to the center of the potential,
- \( \theta \) is the elevation angle,
- \( \phi \) is the horizontal angle, and
- \( t \) is the time.

\( R(r) = |Zr|^i e^{Zr/m} \sum_{k=0}^{n-i-1} b_k \left( \frac{Zr}{m_r} \right)^k \]  \hspace{1cm} (A3)

\[ \Theta(\theta) = P_i^{i/2} (\cos \theta) \]  \hspace{1cm} (A4)
\[ \Phi(\phi) = e^{-i\mathcal{H}\phi} \]  
\[ T(t) = e^{iE_{a}/\hbar t} \]

where

- \( n \) is the principal quantum number \( \{n = 1, 2, 3, \ldots\} \),
- \( l \) is the angular momentum quantum number \( \{l = 0, 1, 2, \ldots, n - 1\} \),
- \( m \) is the magnetic quantum number \( \{-l \leq m \leq l\} \),
- \( r_0 \) is the minimum possible distance from the potential center,
- \( Z \) is the atomic number, denoting the charge that originates the radial potential as a multiple of the charge of a proton (equal in value and of opposite sign to that of the electron),
- \( q_e \) and \( m_e \) are the charge and mass of an electron, respectively, and
- \( P_i^m \) is a family of polynomials known as the associated Legendre functions.

The principal quantum number determines the overall energy of the electron according to the following equation:

\[ E_n = -\frac{1}{n^2} E_a \]  
\[ E_{a-b} = \left[ \frac{1}{n_a^2} - \frac{1}{n_b^2} \right] \times E_a \]  
\[ \text{Calculation of Density from the Input Envelope} \]

As a result of radial symmetry \(|\Phi(\phi)|^2 = 1\) and a uniform temporal distribution \(|T[t]|^2 = 1\), it is convenient to work only with the distribution functions \(|\exp\{i\phi|\exp\{i\phi^2 = \sin(|\theta^2|\theta^2\} \). Physically, this means that there is the same probability of finding an electron at any time interval \( dt \) and that there is no favored horizontal angle. Substituting these values in equation A9 and rearranging terms, we obtain

\[ \text{probability}_{dv} = |r^2|R(r)|^2 \sin(|\theta^2|\theta^2) \, drd\theta d\phi \]  

It is also worth noting that \( R \) is only dependent on \( Z, n, \) and \( l \), and that \( \theta \) is dependent only on \( l \) and \( m \).

\text{Cloud Density}

To convert time from source duration to cloud duration, we use the relation

\[ t_c = t_e \times \frac{\text{cloud duration}}{\text{envelope duration}} \]

where

\( t_e \) is the timing of the cloud density breakpoint,
\( t_e \) is the timing of the envelope breakpoint,
\( \text{envelope duration} \) is extracted from the envelope of the source,
and \( \text{cloud duration} \) is given by the composer.

\text{Calculation of Density from the Input Envelope}
In equation A12, \( r \) and \( \theta \) may be determined according to equations A13 or A14 below at the composer’s discretion. If \( r \) and \( \theta \) are both directly proportional to the envelope, then we have

\[
\begin{align*}
  r(t_e) &= r_{\min} + (r_{\max} - r_{\min}) \times \frac{\text{amp}(t_e)}{\text{amp}_{\max}} \quad (A13) \\
  \theta(t_e) &= \frac{[\theta_{\max} - \theta_{\min}]}{2} \times \frac{\text{amp}(t_e)}{\text{amp}_{\max}}
\end{align*}
\]

If \( r \) is directly proportional to the envelope and \( \theta \) is directly proportional to its slope, we have

\[
\begin{align*}
  r(t_e) &= r_{\min} + (r_{\max} - r_{\min}) \times \frac{\text{amp}(t_e)}{\text{amp}_{\max}} \\
  \theta(t_e) &= \arctan(\text{amp}'(t_e))
\end{align*}
\]

where

\[
\begin{align*}
  r_{\min}, r_{\max} & \text{ are the minimum and maximum radial values given by the composer,} \\
  \theta_{\min}, \theta_{\max} & \text{ are the minimum and maximum elevation angle values (set to 0 and \( \pi \)),} \\
  \text{amp}(t_e) & \text{ is the corresponding envelope amplitude value at time } t_e, \\
  \text{amp}_{\max} & \text{ is the maximum amplitude extracted from the envelope, and} \\
  \text{amp}'(t_e) & \text{ is the derivative of the envelope at time } t_e.
\end{align*}
\]

Alternatively, we can require direct proportionality between the source envelope and the density:

\[
density(t_e) = \density_{\min} + (\density_{\max} - \density_{\min}) \times \frac{\text{amp}(t_e)}{\text{amp}_{\max}} \quad (A15)
\]

Cloud Amplitude Scaling

We can scale the cloud amplitudes according to the following equation:

\[
\begin{align*}
  \text{gain}(t_e) &= \text{gain}_{\min} + (\text{gain}_{\max} - \text{gain}_{\min}) \\
  & \times \frac{[\text{amp}(t_e) - \text{amp}_{\min}]}{[\text{amp}_{\max} - \text{amp}_{\min}]} \quad (A16)
\end{align*}
\]

where \( \text{gain}_{\min} \) and \( \text{gain}_{\max} \) are the values of the minimum and maximum gain (provided by the composer), and \( \text{amp}_{\min} \) and \( \text{amp}_{\max} \) are the minimum and maximum amplitudes extracted from the envelope.

Cloud Motion

Width and Depth

\[
A_L = \frac{\sqrt{2}}{2} \frac{(1 - x)}{\sqrt{(1 + y)^4 + x^4}}; \quad A_R = \frac{\sqrt{2}}{2} \frac{(1 + x)}{\sqrt{(1 + y)^4 + x^4}} \quad -1 \leq x \leq 1
\]

\[
A_L = \frac{\sqrt{2}}{\sqrt{(1 + y)^4 + x^4}}; \quad A_R = 0 \quad x < -1
\]

\[
A_L = 0; \quad A_R = \frac{\sqrt{2}}{\sqrt{(1 + y)^4 + x^4}} \quad x > 1
\]

where

\( x \) is the width, \\
\( y \) is the depth (see Figure 2), and \\
\( A_L \) and \( A_R \) are amplitude factors that multiply the left and right channels, respectively.

Trajectory Creation

\[
x(t_e) = m\text{Width}_{\max} \times \frac{r}{r_{\max} - r_{\min}} \sin \theta \cos \phi
\]

\[
y(t_e) = m\text{Depth}_{\max} \times \frac{r}{r_{\max} - r_{\min}} \sin \theta \abs{\sin \phi}
\]

\[
z(t_e) = m\text{Semitones}_{\max} + \{m\text{Semitones}_{\max} - m\text{Semitones}_{\min}\} \times \frac{r}{r_{\max} - r_{\min}} \cos \theta
\]

where \( r, \theta, \) and \( \phi \) are generated from distributions \( R(r), R(\theta), \) and \( R(\phi) \). The other parameters are given by the composer: \( r_{\min}, r_{\max} \) are the minimum and maximum radial values, \( m\text{Width}_{\max} \) and \( m\text{Depth}_{\max} \) are the maximum absolute value of the width and the maximum depth, and \( m\text{Semitones}_{\min} \) and
\( m_{\text{Semitones max}} \) are the low and high transposition boundaries.

**Cloud Volume (Overall Grain Scatter)**

\[
x_{\text{Scatter}}(t_c) = s_{\text{Width max}} \times \frac{r}{r_{\text{max}} - r_{\text{min}}} \, \text{abs}(\sin \theta \, \cos \phi)
\]

\[
y_{\text{Scatter}}(t_c) = s_{\text{Depth max}} \times \frac{r}{r_{\text{max}} - r_{\text{min}}} \, \text{abs}(\sin \theta \, \sin \phi)
\]

\[
z_{\text{Scatter}}(t_c) = s_{\text{Semitones max}} \times \frac{r}{r_{\text{max}} - r_{\text{min}}} \, \cos \theta
\]

where \( r, \theta, \) and \( \phi \) are generated randomly from distributions \( R(t_c), \theta(\theta), \) and \( \Phi(\phi) \). The other parameters are given by the composer: \( r_{\text{min}}, r_{\text{max}} \) are the minimum and maximum radial values, and \( s_{\text{Width max}}, s_{\text{Depth max}}, \) and \( s_{\text{Semitones max}} \) are the width, depth, and height (in semitones) of the scatter.

**Grain Amplitude Scaling**

*Scaling proportional to Source Envelope*

\[
g_{\text{Gain}}(t_c) = g_{\text{Gain min}} + (g_{\text{Gain max}} - g_{\text{Gain min}}) \times \frac{\text{amp}(t_c)}{\text{amp}_{\text{max}}}
\]

where

- \( g_{\text{Gain min}} \) and \( g_{\text{Gain max}} \) are the minimum and maximum gain (given by the composer),
- \( \text{amp}(t_c) \) is the corresponding envelope amplitude value at time \( t_c \), and
- \( \text{amp}_{\text{max}} \) is the maximum amplitude extracted from the envelope.

*Scaling Proportional to Schrödinger’s Distribution*

\[
g_{\text{Gain}}(t_c) = g_{\text{Gain min}} + (g_{\text{Gain max}} - g_{\text{Gain min}}) \times F(t_c)
\]

where \( F(t_c) \) is given by

\[
F(t_c) = \left( \frac{r(t_c)}{r_{\text{max}}} \right)^2 \sin(\theta(t_c)) \left| R(r(t_c)) \right|^2 \left| \Theta(\theta(t_c)) \right|^2
\]

and

\[
r(t_c) = r_{\text{min}} + (r_{\text{max}} - r_{\text{min}}) \times \frac{t_c}{\text{cloudDuration}}
\]

\[
\theta(t_c) = \theta_{\text{min}} + (\theta_{\text{max}} - \theta_{\text{min}}) \times \frac{t_c}{\text{cloudDuration}}
\]

Here, \( r_{\text{min}} \) and \( r_{\text{max}} \) are the minimum and maximum radial values [given by the composer], \( \theta_{\text{min}} \) and \( \theta_{\text{max}} \) are the minimum and maximum elevation angles.

**Scaling Proportional to the Complement of Schrödinger’s Distribution**

Replace the following definition of \( F(t_c) \) in equation A22:

\[
F(t_c) = 1 - \left( \frac{r(t_c)}{r_{\text{max}}} \right)^2 \sin(\theta(t_c)) \left| R(r(t_c)) \right|^2 \left| \Theta(\theta(t_c)) \right|^2
\]

**Scaling Proportional to Schrödinger’s Cumulative Probability**

Replace the following definition of \( F(t_c) \) in equation A22:

\[
F(t_c) = \int \int \int \left( \frac{r(t_c)}{r_{\text{max}}} \right)^2 \sin(\theta(t_c)) \left| R(r(t_c)) \right|^2 \left| \Theta(\theta(t_c)) \right|^2 \, dr \, d\theta \, d\phi
\]

**Scaling Proportional to the Complement of Schrödinger’s Cumulative Probability**

Replace the following definition of \( F(t_c) \) in equation A22:

\[
F(t_c) = 1 - \int \int \int \left( \frac{r(t_c)}{r_{\text{max}}} \right)^2 \sin(\theta(t_c)) \left| R(r(t_c)) \right|^2 \left| \Theta(\theta(t_c)) \right|^2 \, dr \, d\theta \, d\phi
\]

Fischman
Scaling generated randomly from various distributions

Replace \( F(t_c) \) in equation A22 with values generated by one of the following distributions:

\[
\begin{align*}
& r^2|R(r)|^2 \\
& \sin(\theta)|\Theta(\theta)|^2 \\
& |\Phi(\phi)|^2 \\
& r^2 \sin(\theta)|R(r)|^2|\Theta(\theta)|^2 \\
& r^2 \sin(\theta)|R(r)|^2|\Phi(\phi)|^2 \\
& \sin(\theta)|\Theta(\theta)|^2|\Phi(\phi)|^2 \\
& r^2 \sin(\theta)|\Theta(\theta)|^2|\Phi(\phi)|^2
\end{align*}
\] (A27-A33)

Grain Duration

Duration Proportional to Source Envelope

\[
gDuration(t_c) = gDuration_{\text{min}} + (gDuration_{\text{max}} - gDuration_{\text{min}}) \times \frac{amp(t_c)}{amp_{\text{max}}}
\]
(A34)

where \( gDuration_{\text{min}} \) and \( gDuration_{\text{max}} \) are the minimum and maximum durations \( \text{[given by the composer]} \).

Duration Proportional to Schrödinger’s Distributions

\[
gDuration(t_c) = gDuration_{\text{min}} + (gDuration_{\text{max}} - gDuration_{\text{min}}) \times F(t_c)
\]
(A35)

The various functions used are analogous to those used for grain amplitude, with \( F(t_c) \) given in equations A22, A24, A25, and A26.

Duration Generated from Various Distributions

The distributions used are the same as those in equations A27-A33.

Grain Envelope

Distributions Used as Envelopes

\[
\begin{align*}
& r^2|R(r)|^2 \\
& \text{Reverse of } r^2|R(r)|^2 \\
& \sin(\theta)|\Theta(\theta)|^2 \\
& \text{Reverse of } \sin(\theta)|\Theta(\theta)|^2 \\
& \sin(\theta)|\Theta(\theta)|^2|\Phi(\phi)|^2 \\
& \text{Reverse of } \sin(\theta)|\Theta(\theta)|^2|\Phi(\phi)|^2
\end{align*}
\] (A36-A40)

Grain Scatter Path

\[
\begin{align*}
gxScatter(t_c) &= gWidth_{\text{min}} + (gWidth_{\text{max}} - gWidth_{\text{min}}) \\
gyScatter(t_c) &= gDepth_{\text{min}} + (gDepth_{\text{max}} - gDepth_{\text{min}}) \\
gzScatter(t_c) &= gSemitones_{\text{max}} + (gSemitones_{\text{min}} - gSemitones_{\text{min}})
\end{align*}
\]
(A41)

where

\[
\begin{align*}
gWidth_{\text{min}}, gWidth_{\text{max}} & \text{ are the minimum and maximum width scatter,} \\
gDepth_{\text{min}}, gDepth_{\text{max}} & \text{ are the minimum and maximum depth scatter, and} \\
gSemitones_{\text{min}}, gSemitones_{\text{max}} & \text{ are the minimum and maximum semitone scatter.}
\end{align*}
\]

All of these values are given by the composer. The values \( gWidth, gDepth, \) and \( gSemitones \) are derived from the mean and standard deviation of the distribution functions:

\[
\begin{align*}
gxScatter(t_c) &= rScatter \times \sin(\thetaScatter)\cos(\phiScatter) \\
gyScatter(t_c) &= rScatter \times \sin(\thetaScatter)\sin(\phiScatter) \\
gzScatter(t_c) &= rScatter \times \cos(\thetaScatter)
\end{align*}
\]
(A42)

Here,

\[
\begin{align*}
rScatter &= \frac{r - r_{\text{mean}}}{r_{\text{stdDeviation}}} \\
\thetaScatter &= \frac{\theta - \theta_{\text{mean}}}{\theta_{\text{stdDeviation}}} \\
\phiScatter &= \frac{\phi - \phi_{\text{mean}}}{\phi_{\text{stdDeviation}}}
\end{align*}
\]
(A43)

where

Computer Music Journal
Grain Overall Motion

\[ x(t) = gxScatter(t) + xScatter + x(t) \]  
\[ y(t) = gyScatter(t) + yScatter + y(t) \]

The values \( gxScatter(t) \) and \( gyScatter(t) \) are defined in equation A42, \( xScatter \) and \( yScatter \) are defined in equation A19, and \( x(t) \) and \( y(t) \) are defined in equation A18.

Grain Doppler Shift

The pitch shift relative to the listener is invariant under rotation. Therefore, it is possible to simplify the calculation of the Doppler Effect if we rotate the coordinate system by an angle \( \alpha(t) \), so that the line joining the loudspeakers is kept parallel to the instantaneous velocity of the grain, as shown in Figure 5. The angle \( \alpha(t) \) is then

\[ \alpha(t) = \arctg \left( \frac{dy}{dx} \right) \]  

Also, to maintain consistency, \( \alpha(t) \) must be positive if it produces a counterclockwise rotation.

The Doppler Shift may be expressed as a time-varying frequency ratio calculated according to the following equation:

\[ \text{frequencyRatio}(t) = \frac{1}{1 - \frac{V(t) \cos \theta}{V_s}} \]  

where

- \( V(t) \) is the instantaneous velocity of the grain,
- \( V_s \) is the velocity of sound in air, and
- \( \theta \) is the angle of the grain’s velocity relative to the listener (see Figure 5).

In the rotated system, \( V_s \) and \( \alpha(t) \) can be calculated from the rotated coordinates \( x_{\alpha} \) and \( y_{\alpha} \), which can be obtained from the standard equations for a rotation:

\[ x_{\alpha}(t) = x(t) \cos \alpha + (y(t) + 1) \sin \alpha \]  
\[ y_{\alpha}(t) = -x(t) \sin \alpha + (y(t) + 1) \cos \alpha - 1 \]

Thus we have

\[ V_s = \frac{dx_{\alpha}}{dt} \]  

and

\[ \cos \theta = -\frac{x_{\alpha}}{\sqrt{(1 + x_{\alpha})^2 + y_{\alpha}^2}} \]