5

Little Worlds

ALGORITHMS AND NESTED PATTERNS

In Chapter 2 it was noted that the scheme of scales and keys in western music could be viewed as a nested system of twelve tones set within twelve keys or tonal centers; and in Chapter 4, the underlying structure of tonal music was unpicked further, down to the level of three domains of oscillation, each founded upon the ascending whole number relationships and proportions of the harmonic series – with the domains of timbre (tone color) and pitch (notes and harmony) in principle at least, nested within the low frequencies of the metrical realm (rhythm and duration). In this chapter we shall continue investigating the topic of nested structures, with the aid of the AWK computer programming language and the 'little worlds' of cellular automata. The aim here is to explore the conception of tonal composition considered from the perspective of each piece forming a *tonal universe* – a self-contained pseudo-physical 'little world' evolving from opening notes to final cadence solely through the agency of its own internal relationships. But first a little about AWK itself, the programming language which led me to begin this *Journey to the Heart of Music*.

The Free Software Foundation's GNU AWK package – generally abbreviated to *gawk* – is the classic text manipulation utility for the UNIX computer operating system. Information on the AWK interpreter plus manual and extras can be found at (http://www.gnu.org/software/gawk/gawk.html). Gawk is available for almost all operating systems and is a most useful tool, with many hidden depths; and, as it is so widely implemented, AWK programs (scripts) are entirely transportable between computers with different operating systems. Less of a heavyweight package than its big brother Perl – an expansive programming language that developed out of AWK – the AWK utility has remained focused and compact, a feature which makes it much easier to learn and use. AWK scripts generally have the file suffix .awk, though this is not obligatory. The AWK interpreter will also accept plain text files as scripts. However, it is not actually necessary to get involved with running AWK scripts if you are not familiar with this area of computers; there are example text files provided in the Chapter 19 directory/folder as well as many printed figures spread throughout this chapter.

Cellular Automata

In the Chapter 19 directory/folder online and on the *Journey to the Heart of Music* CD, can be found the AWK script *ca.awk,* which generates patterns of text characters by repeatably applying a simple *rule* to a given initial condition. The output of such programs or scripts is usually called *cellular automata,* and their intriguing patterns were researched by John von Neumann in the 1950s, John Conway from the 1960s (The Game of Life) and Stephen Wolfram from the 1980s, among others. The name cellular automata derives from the combination of their ‘cellular’ structure, rather like graph paper, and the ‘automatic’ application of a fixed rule. Because AWK is a text-based package, the output from *ca.awk* is a text file of Xs and spaces, which, if viewed in a text editor *with a monospaced* font, mimics this grid-like cellular structure. A monospaced font is crucial, e.g. Courier, so that the individual cells line up and form the intended pattern. Figure 5.1 is an example cellular automaton (rule18) generated by ca.awk.

```
_______________________________X_______________________________
                              X X
                             X   X
                            X X X X
                           X       X
                          X X     X X
                         X   X   X   X
                        X X X X X X X X
                       X               X
                      X X             X X
                     X   X           X   X
                    X X X X         X X X X
                   X       X       X       X
                  X X     X X     X X     X X
                 X   X   X   X   X   X   X   X
                X X X X X X X X X X X X X X X X
               X                               X
              X X                             X X
             X   X                           X   X
            X X X X                         X X X X
           X       X                       X       X
          X X     X X                     X X     X X
         X   X   X   X                   X   X   X   X
        X X X X X X X X                 X X X X X X X X
       X               X               X               X
      X X             X X             X X             X X
     X   X           X   X           X   X           X   X
    X X X X         X X X X         X X X X         X X X X
   X       X       X       X       X       X       X       X
  X X     X X     X X     X X     X X     X X     X X     X X
 X   X   X   X   X   X   X   X   X   X   X   X   X   X   X   X
X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X
```

Figure 5.1 The cellular automaton produced by ‘rule18’ and generated by the script ca.awk. Notice the many levels of nested triangles. The pattern was discovered by the mathematician Waclaw Sierpinski early in the 20th century.

The script ca.awk functions by loading instructions from a four-line text file and applying the instructions as it runs. The script’s output is sent to a text file named *ca_new*, placed in the same location as the instruction file. The file of instructions can have any name; here they are called ‘rule’ plus a number from 0 to 255, e.g. rule90. Figure 5.2 shows the contents of the rule90 instruction file. The rule is a series of eight comma separated instructions, the first is ‘XXX:_’, which determines from the current line how the next line of the pattern is to be constructed. The script looks at each sequence of three character positions in the initial condition (and subsequent lines generated thereafter) and applies the particular instruction that fits those three characters. This is known as a *next neighbour dependent rule* as the character plus its two neighbours, to the left and right, determine the outcome for that character position on the next line.

```
#rule90 %0101 1010
rule: XXX:_,XX_:X,X_X:_,X__:X,_XX:X,_X_:_,__X:X,___:_
initial_condition: _________________X__________________
generations: 50
```

Figure 5.2 The contents of a rule text file.

Thus the first three characters of the initial condition in Figure 5.2 ‘___’ match the last of the eight instructions, so on the next line, character position ‘two’ will be what follows the colon ‘___:_’ in the instruction. In this case an underscore ‘_’ character. Character position one is determined from wrapping around the last character of the previous line or the initial line. (If you look at the script/program itself, the underscore character ‘_’ is used within the script as a visible stand-in for the space character – once the processing is complete the space character is substituted for the underscore, giving clearer output.) The number of lines created by the script equals the number of generations. You can experiment with the initial condition, have more Xs, no Xs, different positions, wider, narrower, etc. and have more or fewer generations. Just create/copy a text file/rule, make the appropriate changes and pass it to the ca.awk script.

In Chapter 3, of Stephen Wolfram’s book *A New Kind of Science* he introduces a neat scheme for the codification of a set of rules by their binary digits. Thus in *rule124* – one hundred and twenty-four has the binary representation 0111 1100 – the zeros equal ‘_’ and the ones equal ‘X’ for the character output to the next line (the part of the instruction after the colon). Here is a fragment of output from rule124.

```
___X______________________
   XX
   XXX
   X XX
   XXXXX
   X   XX
   XX  XXX
   XXX X XX
   X XXXXXXX
   XXX     XX
   X XX    XXX
   XXXXX   X XX
   X   XX  XXXXX
   XX  XXX X   XX
   XXX X XXXX  XXX
   X XXXXX  XX X XX
   XXX   XX XXXXXXXX
   X XX  XXXX      XX
   XXXXX X  XX      XXX
   X   XXXX XXX    X XX
   XX  X  XXX XX   XXXXX
   XXX XX X XXXXX  X   XX
   X XXXXXXXX   XX XX   XXX
```

Figure 5.3 Output from AWK with rule 124.

If you are familiar with a computer command line environment, you might well skip this section and if you don’t want to run AWK scripts there are example cellular automata included on the CD. However, given below is brief outline of running the ca.awk script for what might be described as a generic command prompt/terminal. This overview is provided as a general description and not a detailed guide;

please seek specific advice from your system administrator or other competent source before proceeding if you are unsure.

After opening a command environment, locating the AWK interpreter (probably already included in your PATH variable) and setting up the script ca.awk and rule files in appropriate directories, it is simply a matter of issuing the command:

```
gawk -f ca.awk rule'n'
```

(where 'n' is a number 0 to 255)

The output generated by running the script is a text file named `ca_new` which will appear in the current working directory. Confirmation that the ca.awk script has completed the task is printed to the terminal.

Using the Command Prompt (Terminal Window)

Commands are of the form: `command parameter1 parameter2` etc., separated by spaces and followed by pressing *Return*. They are messages to the computer's operating system requesting it to run the program named in the command. This program might be the `dir`(ectory) program, the `date` program, the `gawk` program, etc. All other space-separated expressions which follow the command are items of information which the operating system will pass on to the program.

Thus taking the (g)AWK command apart – `gawk -f ca.awk rule90`

1) `gawk` – the command which runs the gawk interpreter (program). This assumes the computer can find it, so if you get a grouchy 'not found' error, the operating system has not been told where gawk is located on your system.

2) `-f` – an option which tells gawk to load the first following non-option parameter (file path/name) as the program/script. Options have a leading dash(es) '-' or '--'.

3) `ca.awk` – the script, gawk knows that this is the program from the -f option. Otherwise it would assume it was an input file.

4) `rule90` – Anything after the script is taken to be the name/path of an input file(s): there can be zero, one or more of them. (Though in the case of ca.awk, giving the script zero instruction files is not very useful).

```
C:\>
C:\>cd \scripts
C:\scripts>gawk -f ca.awk rule90
ca.awk script completed run
C:\scripts>_
```

Figure 5.4 A sequence of two 'messages' to the command line interpreter, followed by confirmation from the ca.awk script that the job is finished, displayed in a terminal window by the operating system.

```
_________________X________________
                X X
               X   X
              X X X X
             X       X
            X X     X X
           X   X   X   X
          X X X X X X X X
         X               X
        X X             X X
       X   X           X   X
      X X X X         X X X X
     X       X       X       X
    X X     X X     X X     X X
   X   X   X   X   X   X   X   X
  X X X X X X X X X X X X X X X X
```

Figure 5.5 A fragment of output generated by the command:
`gawk -f ca.awk rule90`

‘Xland’

The patterns produced by ca.awk could be interpreted as little experimental systems stripped to the minimum – toy worlds. Julian Barbour in his challenging book *The End of Time* evokes the charmingly named *Triangleland*. In a similar vein, the ‘toy text-file world’ generated by ca.awk might be called *Xland* as its sole inhabitants are Xs and something not X, usually the space character or underscore. X and !X (not X) could be interpreted as an archetypal cycle, peak and trough, both vertically and horizontally, a visual representation of simple oscillation, with unitary amplitude. At the most basic level – Figure 5.6. – Xland patterns could be interpreted as a unary representation of number relationships, that is a succession of prime state (i.e. purely additive) mutable numbers:

h1	MBN 1_1	Factor	format:	1×1
h1, 2	MBN 2_1	"	"	1×2
h1, 2, 3	MBN 3_1	"	"	1×3
h1, 2, 3, 4	MBN 4_1	"	"	1×4
etc.				

However, most Xland patterns contain spaces (!X) as well as Xs, and this introduces the possibility of visually arresting nested patterns appearing within the overall configuration. Where the sequences of Xs and spaces form regular recurring row or column arrangements. These structures could be interpreted as nested harmonic series, or in other words multi-column mutable numbers. Figure 5.7 shows a slice cut through the pattern of rule90 which yields the mutable number sixteen in its ground state:

```
h1,2,4,8,16             MBN Factor format: 1×2×2×2×2
```

You can visit Xland in any text editor (for wide patterns set the line length appropriately large and always use a monospaced font) and if you don’t want to run AWK scripts yourself, a selection of the text files are on the *Journey to the Heart of Music* CD in the Chapter 19 directory.

```
     h123456789............h24
 h1 __X_______________________
 h2   XX
 h3   XXX
 h4   XXXX
 h5   XXXXX
 h6   XXXXXX
 h7   XXXXXXX
 h8   XXXXXXXX
 h9   XXXXXXXXX
h10   XXXXXXXXXX
h11   XXXXXXXXXXX
h12   XXXXXXXXXXXX
h13   XXXXXXXXXXXXX
h14   XXXXXXXXXXXXXX
h15   XXXXXXXXXXXXXXX
h16   XXXXXXXXXXXXXXXX
h17   XXXXXXXXXXXXXXXXX
h18   XXXXXXXXXXXXXXXXXX
h19   XXXXXXXXXXXXXXXXXXX
h20   XXXXXXXXXXXXXXXXXXXX
h21   XXXXXXXXXXXXXXXXXXXXX
h22   XXXXXXXXXXXXXXXXXXXXXX
h23   XXXXXXXXXXXXXXXXXXXXXXX
h24   XXXXXXXXXXXXXXXXXXXXXXXX
```

Figure 5.6 The cellular automaton generated by rule220, which counts downward and outward from the first to the nth character position.

Stephen Wolfram distinguishes four categories of behaviour in the 256 possible Xland cellular automata patterns. Simple repetition (rule1) and nested structures (rule90) being the two most common. However, despite the simplicity of the underlying rules, a smaller number display apparently random features as well as repetition and nesting (rule30, 45 & 73); and rule124 contains all four categories: repetition, nesting, random features and local structures. I shall leave you, the reader, to experiment and explore Xland’s topography, either through directly running the ca.awk script or by looking at the example output files.

```
_________________X_________________h1
                X X                h2
               X  |X                |
              X X X X              h4
             X    |  X              |
            X X   | X X             |
           X   X  |X   X            |
          X X X X X X X X          h8
         X        |      X          |
        X X       |     X X         |
       X   X      |    X   X        |
      X X X X     |   X X X X       |
     X       X    |  X       X      |
    X X     X X   | X X     X X     |
   X   X   X   X  |X   X   X   X    |
  X X X X X X X X X X X X X X X X h16
```

Figure 5.7 A slice through the cellular automaton pattern generated by rule90 which could be interpreted as a representation of the mutable base number sixteen. Factor format 1x2x2x2x2 or subscript format MBN $2_2\,0_2\,0_2\,0_1$.

Little Worlds

Xland patterns, though notional toy worlds, are of course also tiny fragments of the *System of the Great World* – the real material world – physical arrangements: particles, atoms, molecules, ink and paper, computer random access memory, electron beam and screen, LED array,... And, perhaps, the material world is nothing other than a patchwork of innumerable such fragments, sub-systems operating at all scales and over every imaginable range. Musical compositions, in written format or performance, are likewise tiny fragments, toy oscillatory worlds. And in the case of tonal music, these fragments form logically self-consistent systems of numerical relationships expressed in sound and stored as written scores – systems of twelve tones nested within twelve tonal centers. Abstract mathematics, as I have termed it, is yet another of these toy worlds, a conceptual universe constructed upon a set of axioms similarly extracted from our experience of the material world. However, when all the possible separate sets of axioms and their mathematical universes are considered, the myriad possible worlds of mathematics may well exceed in extent the great material world out of which they have grown. Indeed, somewhat reminiscent of the relation between limited computer hardware and the freedom of software, there is perhaps a seeming limitlessness to abstract mathematics.

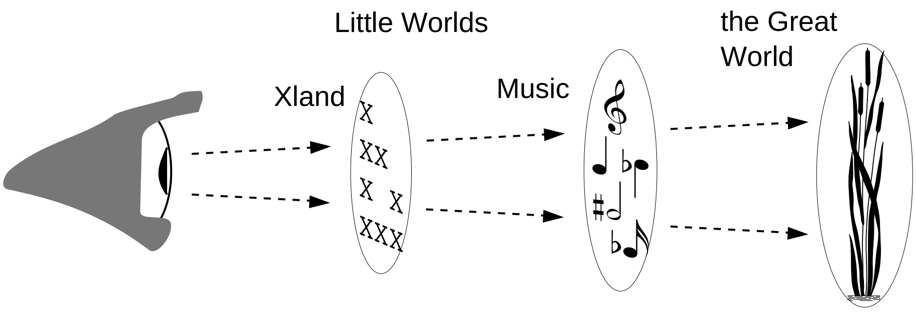

Figure 5.8 Looking into tonal music's system of organisation (and beyond it), with the aid of Xland patterns.

However, the little relational worlds formed by musical compositions, though considerably constrained by the hardware limitations of acoustics and the human hearing system, have a particular virtue: unlike abstract mathematics, they stand on the threshold of material existence, mixing together characteristics of formal symbolic systems with aspects of self-organisation typical of the material world. The study of such little worlds, whether Xland cellular automata, musical compositions or any other fragment, can probably yield insights which might have general application across many systems. It is after all one of the basic procedures of scientific method to isolate and reduce a problem to a minimal state, so as to be able to gain an understanding of each individual component. Xland, the Xland cellular automata, like the number patterns discussed in Chapter 1, are simple visual tools which can help us to see relationships and focus on the essential details of music – as well as perhaps, other more fully natural physical systems.

In the remainder of this chapter, we shall take something of a walk on the speculative wild side, following paths of thought, and thought experiments, through to sometimes rather extreme and bizarre positions. This is admittedly a rather exploratory bivouac across difficult terrain. The overall aim is to break ingrained habits of thought concerning music: to gain a fresh perspective on tonally organised music by viewing compositions in computational terms, as self-organising quasi-physical systems evolving logically through ordered sets of relationships. This is the core idea to keep in mind in this chapter, tonal compositions viewed in principle as *patterns of oscillatory relationships, evolving from chord to chord, through the application of a simple rule or algorithm*. Which is to say, systems which build up structure by computed steps from the first chord to the final cadence, rather like the growth of cellular automata patterns – though the harmonies of music quickly die away, they are not entirely lost from the mind. As with the Xland patterns, the current position in the evolution of the structure (wherever that may be in the piece of music) could not have been reached, in principle, except through the sequence of steps that took it there – the 'history' of the structure. The construction of a building's roof only becomes possible after the walls below it has been put in place. I hope you find the ideas interesting and stimulating.

NEW PERSPECTIVES

It is often said that Newton gave us the 'clockwork' universe; however, viewed from another perspective, it might be said that clockwork gave us the Newtonian universe. Clocks, the epitome of European high technology in the seventeenth century, were at the forefront of attempts to solve the problem of reliably gauging longitude, a crucial matter in the age of navigation. This preoccupation with building precise mechanisms perhaps gave a slant to the thinking of the time, a mind-set primed to discover a mechanistic principle in Nature. Similarly today's technology is leading (and facilitating) our thinking toward another view of Nature: the additional perspective of information and computation. Stephen Wolfram[2] makes the point in his book *A New Kind of Science* that the 'Notion of Computation' (page 638) provides a unifying approach to investigating the material world:

> "And by thinking in terms of such computations, it then becomes possible to imagine formulating principles that apply to a very wide variety of different systems."

The Nested Dimension

Continuing in the spirit of this dictum; and remembering, as discussed above, that an Xland pattern laid out in a text editor's window could be interpreted as a *little world* – a simple two-dimensional toy system of horizontal and vertical relationships of X and !X – 'not-X' being the space character. We could proceed to associate the left/right configurations of X and !X – the horizontal dimension – with (toy/logical) pitch and up/down configurations of X and !X with (toy/logical) time. Both of these dimensions are basically an ordered set of relationships, configurations of Xland coordinates that we are choosing to interpret as units of pitch (notes/timbre) and time (tempo/duration). But what distinguishes the logical time dimension running down the page is that it also happens to be the direction in which the whole pattern is evolving. As the ca.awk script computes each character and line, so the whole Xland pattern expands from the top, downward.

Such automata are echoing the grand theme of an expanding universe. An Xland cellular automaton viewed as a 'toy' universe evolving outward along one of its *dimensions*, carrying the other (notes/timbre) dimension(s) nested within that expanding fundamental (time) dimension. In essence the direction/ dimension of expansion of the Xland pattern defines an ultimate, lowest level of nested structure. We have chosen to associate this set of relationships with time, the tempo/duration domain of music, as these metrical units are the lowest frequency cycles found in musical compositions. The output generated by rule110 in Figure 5.9 illustrates this nesting of one dimension within another.

```
 <-- pitch-space dimension -->
____________________________X_
   t                       XX
   e                      XXX
   m                     XX X
   p                    XXXXX
   o                   XX   X
   /                  XXX  XX
   d                 XX X XXX
   u                XXXXXXX X
   r.              XX     XXX
                  XXX    XX X
   e             XX X   XXXXX
   v            XXXXX  XX   X
   v           XX   X XXX  XX
   o          XXX  XXXX X XXX
   l         XX X XX  XXXXX X
   u        XXXXXXXX XX   XXX
   t       XX      XXXX  XX X
   i      XXX     XX  X XXXXX
   o     XX X    XXX XXXX   X
   n    XXXXX   XX XXX  X  XX
       XX   X  XXXXX X XX XXX
      XXX  XX XX   XXXXXXXX X
     XX X XXXXXX  XX      XXX
    XXXXXXX    X XXX     XX X
   XX     X   XXXX X    XXXXX
  XXX    XX  XX  XXX   XX   X
 XX X   XXX XXX XX X  XXX  XX
```

Figure 5.9 Output from rule110 viewed as a two-dimensional musical universe, with left/right assigned the dimension of notes and timbre – ‘pitch-space’, and up/down assigned the dimension of tempo/duration – ‘musical time’.

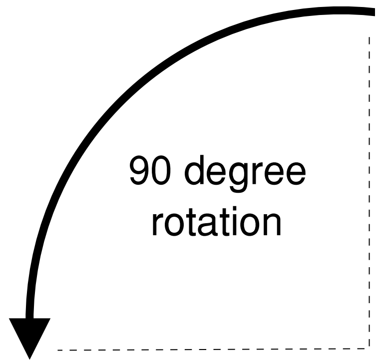

Figure 5.10 Output from rule110 with dimensions rotated by 90 degrees.

```
---> tempo/dur evolution --->
|
XXXXXXXXXXXXXXXXXXXXXXXXXXXX
|XX X XX X X XX X X XX X X X
| XXX  XXXXX  XXXXX  XXXXX
|  XX   X XX   X XX   X XX
|   XXXXX  XXXXX  XXXXX  XXX
|    XX X   XX X   X XX   XX
|  p  XXX    XXXXXXX  X    X
|  i   XX     X XX XXXX
|  t    XXXXXXX  X  X X
|  c     XX X XXXXXXXXXXXXXX
|  h      XXX  XX XX X XX X
|          XX   X  XXX  XXXX
|  s        XXXXX   XX   X X
|  p         XX X    XXXXX
|  a          XXX     XX XXX
|  c           XX      X  XX
|  e            XXXXXXXX   X
|                XX X XX
|  d              XXX  XXXXX
|  i               XX   X XX
|  m                XXXXX  X
|  e                 XX X
|  n                  XXX
|  s                   XX
|  i                    XXXX
|  o                     XX
|  n                      XX
|                          X
```

There is an alternative *ca2.awk* script (in the Chapter 19 scripts directory on the CD) that prints directly to the terminal window so as to simulate this continuous expansion along the up/down dimension. It is easy to fall into the habit of seeing these little worlds (and music scores) as static printouts rather than as dynamic processes in motion, or better, in computation and performance.

Now imagine that the output from ca.awk for rule110 is rotated ninety degrees anti-clockwise, so that the tempo dimension expands across the page. And then next convert the pattern in your mind into a musical cellular automaton – a piece of music – the Xs are notes and chords, and the !X rests – the absence of notes on the staff. Each (now vertical) column of evolution, running from left to right across the page, represents the pulse or some unit of musical time, with the columns of Xs forming chords, rather like a MIDI editor's *piano roll* display (Figure 5.11).

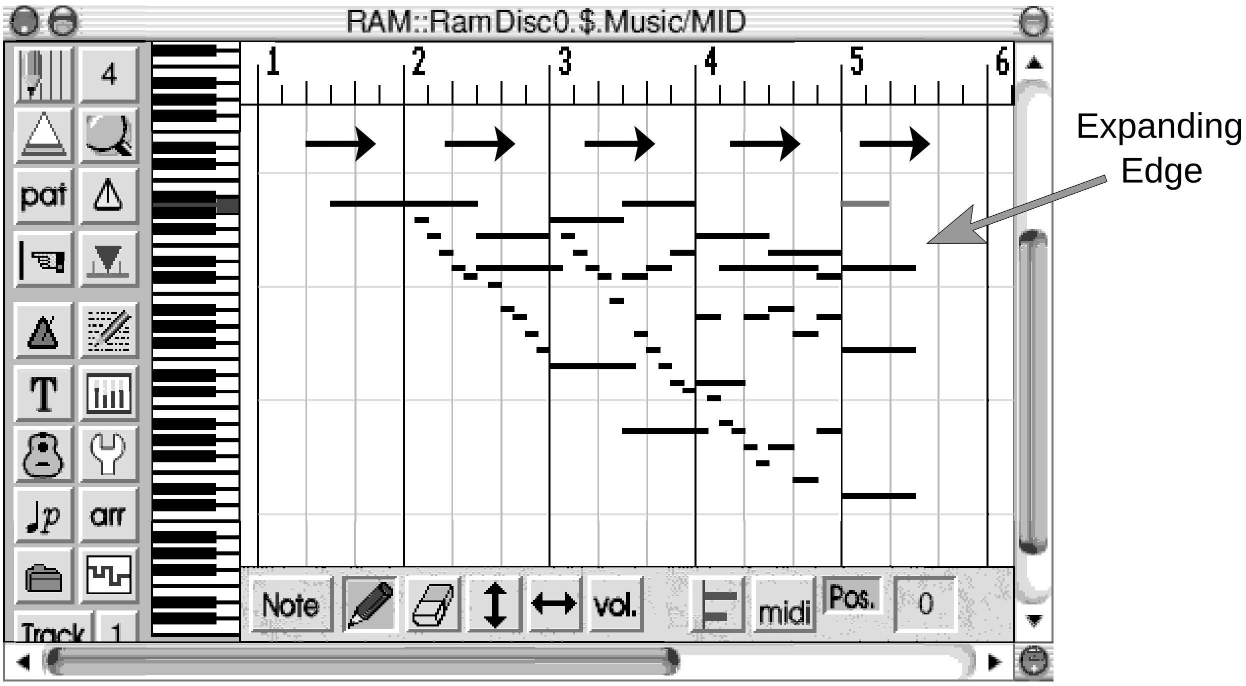

Figure 5.11 Output similar to that from rule110 in Figure 5.10, displayed in a MIDI editor's piano roll window. (This MIDI editor program doesn't distinguish between 6/8 and 3/4 time.)

It is the dimension of tempo, toy time, the passage through the beats and measures (marked by black arrows in Figure 5.11) which forms the *expanding edge* of this *little world of music*.

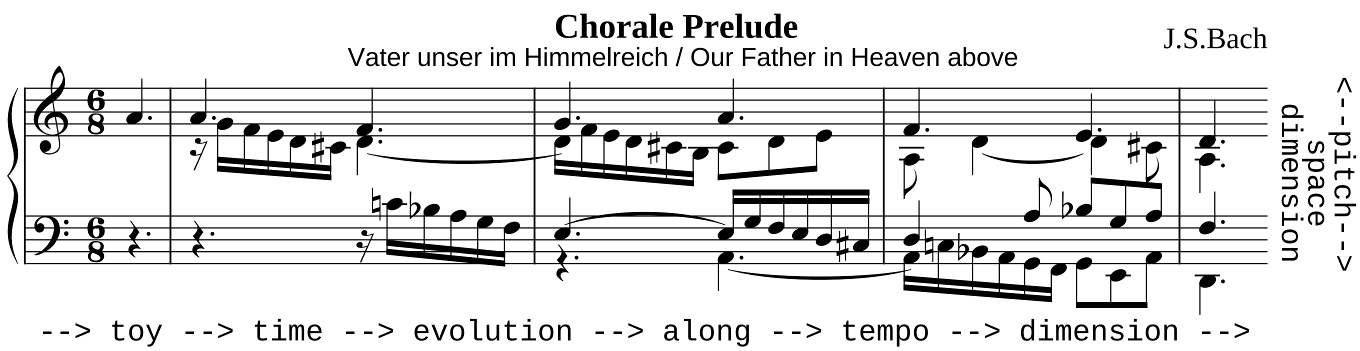

Figure 5.12 The output shown in Figure 5.11 (Vater unser im Himmelreich) similar to that from rule110, displayed as a music score.

And it's not just any edge that is evolving the music along its dimension, it is the bottom edge, the longest waves, the most fundamental frequencies of tempo: the ratios which, in principle, nest or hold within themselves the periods and cycles of the other dimensions of music – notes and timbre. Effectively the dimensions of pitch and timbre are being carried forward, embedded within the expanding edge of the tempo dimension.

A Thought Experiment

If a piece of music is viewed as a *toy world* then the tempo, the cyclic motion of the metronome's pendulum (in the real world) is marking out an equally *toy time*. We can play with toy time: run it fast or slow – largo or allegro, speed up or slow down – accelerando or ritardando. But would such differences be felt inside our little world? Imagine life within the relationships of such a *little world*, if you can.

Without any recourse to external measures of time, a toy world inhabitant would have to find an internal *tick or oscillation*, the shortest note value present perhaps, to measure out durations of time. But would not this reference tick expand and contract as the tempo varied? With the result that changes of tempo applied externally could not be perceived by the toy world inhabitant? Ah, you say what about an 'atomic clock' reading the frequency of A=440Hz? This would yield an immutable oscillation with which the passage of time could be measured. The piano tuner answers, with a sigh, "The sad truth is that many pianos are not tuned to standard concert pitch, and indeed, some aren't tuned at all!" And how would the toy world inhabitant know which note was middle A, even if it was 440Hz? The internal relationships could indicate which note was the tonic, but not what note-letter the tonal center actually was. My (tentative) conclusion is that motion, in the sense of musical events – the changes in a composition's 'spatial configuration' – would mark or make ticks, experienced as *real time* internally, while also agreeing with variable toy time, externally.

Now, continuing with this vein of thought experiment, where we are inhabitants of a musical *little world*, bound within the relationships of a tonal composition, imagine standing on the top stave at the beginning of a composition ready to keep time (4/4) by pacing out the beats as they occur. The example consists entirely of quarternotes – Figure 5.13. (The wedge-shaped structure is discussed later.)

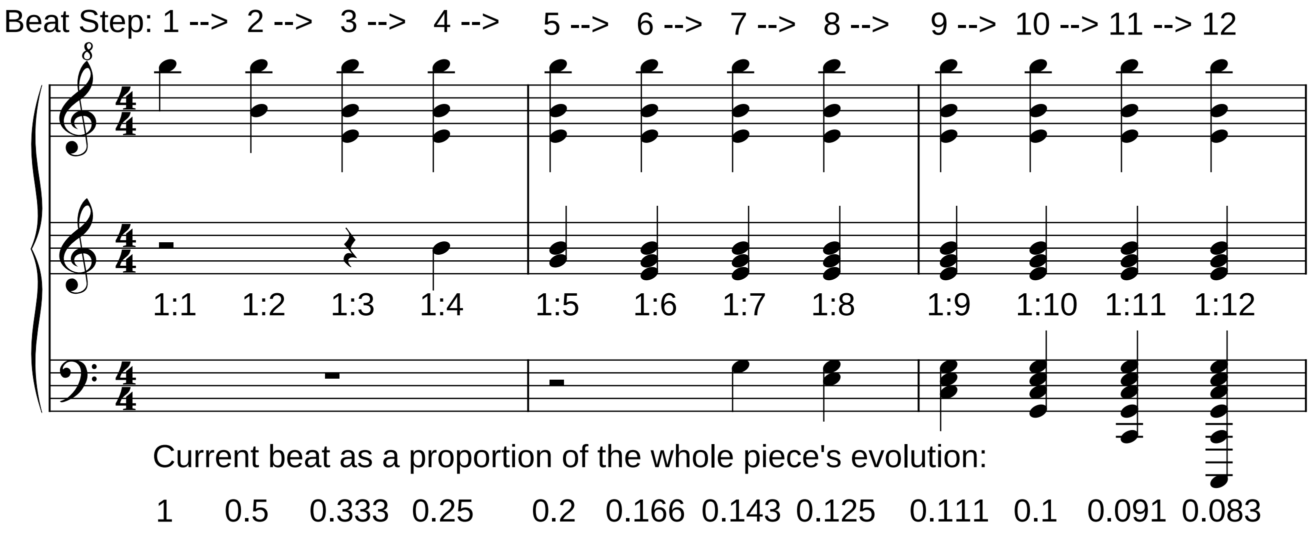

Figure 5.13 Walking the beat – with each step delineating a 'temporal harmonic series'.

The music starts and we take the first step. We travel one beat and the period of the whole piece from start to this position is one beat – the ratio of our step (motion) to the performed extent of the whole piece is 1:1. The second beat takes us another single step along the tempo dimension. Now our situation is that in this beat our motion was one-half the period of the whole performed extent, which is at this point two beats long. In the third beat we take another step forward and find that the distance we have covered in this step is one-third the period of the performed extent of the piece. And with the fourth beat we only cover one-quarter the time of the period of the performed extent, which is now one measure, four beats long. You will no doubt have recognised the sequence 1:1, 1:2, 1:3, 1:4, ... as the harmonic series; and this progression, the subdivision of the cycle or period of the whole composition, continues to the final cadence – a step by step forward motion along the relationships of the tempo dimension: h1, h2, h3, h4, ... hn. With the cycle of the whole piece, the ultimate fundamental period of this toy world, growing outward, expanding from zero at the start, to encompass, eventually, the composition's entire length. It is, I suspect, the stepwise evolution of this most fundamental wave, powered by the computations of harmonic progression, which is the music's ultimate driving force; and our awareness of the passage of (musical) time is essentially the perception of this expansion. Rule114 below, Figure 5.14, encapsulates this subdivision with the first (wrapped around) cycle of X and !X being gradually divided by each succeeding beat.

```
      divisions of h1 cycle
 1 ____________X____________
 2  b         X X         t
 3  e        X X X        e
 4  a       X X X X       m  d
 5  t      X X X X X      p  i
 6  s     X X X X X X     o  m
 7       X X X X X X X       e
 8      X X X X X X X X      n
 9     X X X X X X X X X     s
10    X X X X X X X X X X    i
11   X X X X X X X X X X X   o
12  X X X X X X X X X X X X  n
```

Figure 5.14 Output from rule114, with each step of the triangular number pattern's evolution creating the whole number series 1, 2, 3, 4, 5, 6, etc. of Xs, the natural numbers.

Now, standing at the beginning of the second bar and having got bored with constantly moving in one direction (forward), we decide to turn around and go the other way – backward down the tempo dimension (ratios). How do we do it? Our current ratio to the period of the performed extent is 1:4. To go forward takes to 1:5, to turn back we need to step to the 1:3 relationship with the period of the performed extent in the next beat. That is, take a longer stride, one with a ratio of one-third to the extent. But there is a problem: in the next beat the extent will be five beats long and our step will need to be 1.666 beats (1/3 of 5). Effectively to achieve our goal of backward motion in the tempo dimension we must step through the leading edge of the music's evolution! To travel back down the temporal harmonic series to h3 and beyond to h2 and h1, we need to have more motion than the expansion of the little world in which we are bound will allow: Each moment of existence lies on the evolving boundary of the pattern, only the present beat is.

A Cheat?

Is there a cheat? Could we step two beats forward in one stride, thereby forcing the evolution of the period of the whole piece to six beats... and having covered two beats in one step, have the desired ratio of three to the whole (1:3)? Here again there is a problem. Our steps are like the ticks in the original thought experiment: any one step, however large or small (when viewed from outside the little world) will represent only one beat inside it – there are no divisions smaller than a quarternote amongst the chords. However wide (or short) our step, it is still only one step of the music's evolution – one beat – viewed from inside the relationships of our experimental composition. From our position inside the system the choices are rather limited. Tempo flows forward (to the next ratio of the harmonic series) if we take a step of any size; and, if we pause, no beat passes – a pause would only be visible from outside the system. No musical time passes within individual beats as the smallest note division in this piece is the one-beat-long quarternote.

Thus from within the system the temporal dimension (which is founded on the music's most fundamental wave – the extent of the composition at whatever point it has reached in its evolution) has no 'longer waves' available to it – i.e. ratios lower than h1 – to allow backward motion. Tempo, musical time, appears to be a one-way street viewed from inside the system; and you can't stop (or start) it either.

However, we make and listen to music in the real world, outside the confines of music's patterns, and so can easily find ratios larger than one step along the ratios of the harmonic series – like the 1.666 beat – otherwise called ritardando (and 'cheat' the disciplines of the system's logic in other ways too). This has brought us back to the original thought: Music's measure of time – tempo and duration – is real/hard time viewed from inside the system, but for us in the Great World outside, no more than a toy to play with.

Temporal 'Harmony'

A key element of music's texture lies in a rich diversity of durational values – a pleasing mix of long and short notes. These temporal relationships based on the pulse and the bar, music's longer waves, form in principle the foundation frequencies upon which the whole edifice of notes, harmony and timbre is built (see Figure 5.18 later). However, while life in the basement is perhaps not as much fun as in the garden flat or penthouse, there is life down on the bottom floor. And when the ratios of temporal duration are examined they turn out to be not so different from those of pitch or timbre.

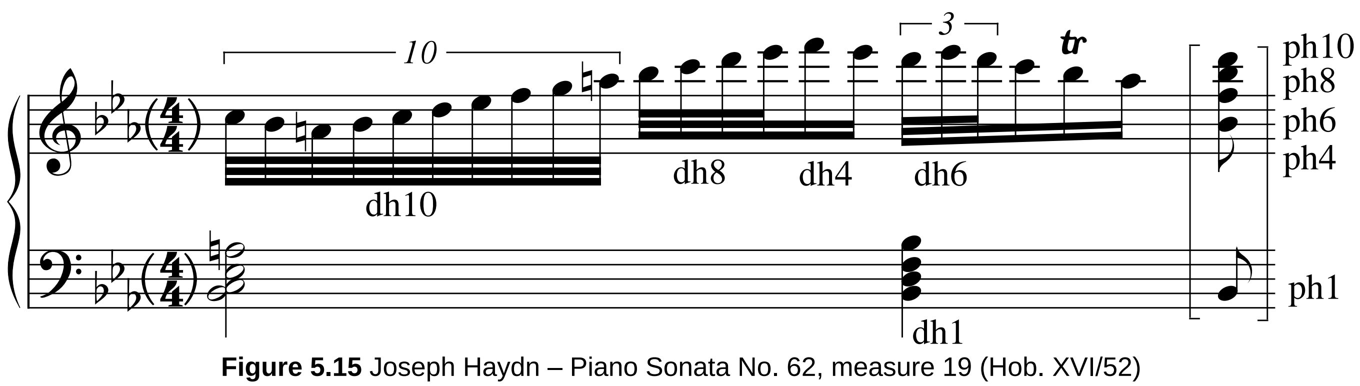

Figure 5.15 Joseph Haydn – Piano Sonata No. 62, measure 19 (Hob. XVI/52)

In Figure 5.15, an extract from a piano sonata by Haydn, the 4/4 quarternote pulse is labelled d(uration)h1 and its further subdivisions: dh4, dh6, dh8 and dh10. These ratios of duration are then laid out at the end of the bar as if ratios of pitch: ph1, ph4, ph6, ph8 and ph10. As can be seen, the range and structure of the rhythmic ratios in this measure are describing very much the same information as do the ratios of pitch. Thus, although the actual frequencies of rhythm and notes lie far apart, they sing a similar song, a nested 'harmony'. At a tempo of quarternote = 100MM and concert pitch, Bflat dh1 equals 1.66Hz and Bflat ph1 equals 116.5Hz, the separation is roughly a factor of 70:1. I must confess that it took a while to find this example; generally, life in the basement is of a plainer, humbler kind, particularly in pieces of quick tempo where there is often only room for a simple structure like: dh1, dh2 and dh4. Yet even under such restrictive circumstances there is some degree of nested structure.

Hardware/Software Entanglement

Musicians often have strong opinions about the 'right' tempo for a particular piece of music and this has led me to wonder if there might be some (unconsciously felt) connection between the 'configuration' of note frequencies and the rhythmic frequency of a 'right' tempo. But, as no two musicians will ever agree on what this 'right' tempo actually is, it would seem to be a hard proposition to pin down.

	C Scale (ratio)	G Scale (ratio)	Eq.Temp'd (hertz)
C	1:1024h32	1:1008h21	1024
B	1:960-h30	1:960-h20	966.53
A#	1:896-h28	1:912-h19	912.28
A	1:832-h26	1:864-h18	861.08
G#	1:800-h25	1:816-h17	812.75
G	1:768-h24	1:768-h16	767.13
F#	1:704-h22	1:720-h15	724.08
F	1:672-h21	1:672-h14	683.44
E	1:640-h20	1:624-h13	645.08
D#	1:608-h19		608.87
D	1:576-h18	1:576-h12	574.70
C#	1:544-h17	1:528-h11	542.45
C	1:512-h16		512
B	1:480-h15	1:480-h10	483.26
A#	1:448-h14		456.14
A	1:416-h13	1:432-h9	430.54
G	1:384-h12	1:384-h8	383.57
F#	1:352-h11		362.04
F		1:336-h7	341.72
E	1:320-h10		322.54
D	1:288-h9	1:288-h6	287.35
C	1:256-h8	(middleC)	256
B	1:240-h5		241.63
A#	1:224-h7		228.07
G	1:192-h6	1:192-h4	191.79
E	1:160-h5		161.27
D		1:144-h3	143.67
C	1:128-h4		128
G	1:96 -h3	1:96 -h2	95.89
C	1:64 -h2		64
G		1:48 -h1	47.95
C	1:32 -h1		32

Figure 5.16 Natural scales based on the harmonic series of C-32Hz and G-48Hz; both of which are derived from the fundamental unit of tempo, which can be expressed as: 60MM, or 1Hz, or H1.

However, continuing this line of thought, it is possible to construct an ideal system, a note-scale which has a direct proportional relationship with a particular tempo, such that a 'harmonic *software* level' can be *run* on the extrapolated ratios of the fixed patterns of '*hardware'* relationships that emanate from the low level of temporal duration. In effect each beat or unit of tempo is acting like a computer's cpu tick and the ratios of the tick's harmonic series represent the allowable whole number values which can occupy the hardware registers. If we choose the easy tempo of one quarternote per second, 60MM = 1Hz and pitch of middle C=256Hz, then by tracing up the ratios/harmonics of this h1 beat (1Hz) to the level of 1:1024 (1024Hz) *natural* scales derived from this harmonic series can be constructed, using selected 'register' values – Figure 5.16.

Nested 'Software Keys'

Using these temporal hardware register values (the harmonic ratios of the two natural scales above) we can encode musical information in a software system of 'Two Keys' – Figure 5.17. (Notwithstanding that the differing perspectives of the natural scales built on C and G lead to disagreement over some note ratios, this flexing of relationships is a characteristic of tonally organised systems.)

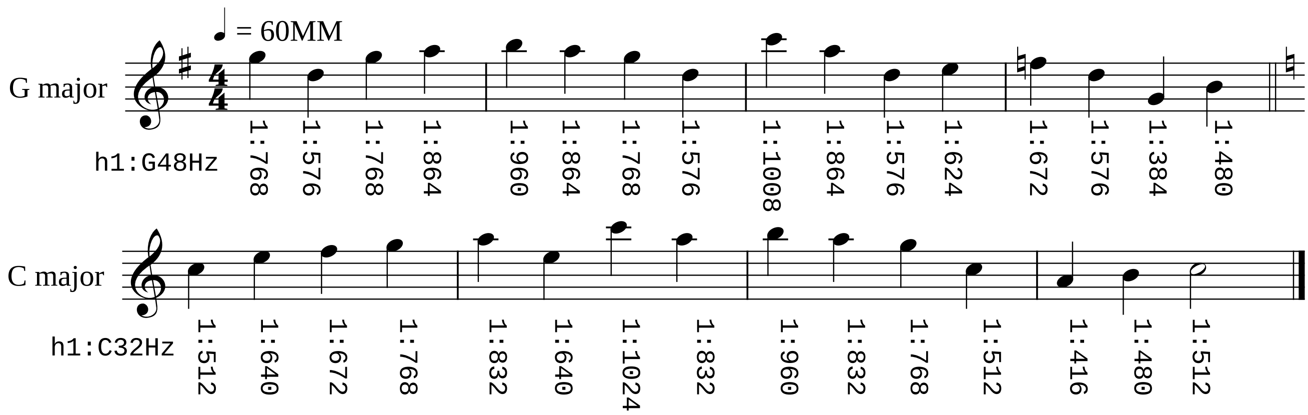

Figure 5.17 A melody which moves from a natural scale (tonal center or key) built on G 48Hz to a natural scale built on C 32Hz. Notice as a consequence that in the first line A = 1:864 but in the second A = 1:832.

Though a trivial melody – starting life in the key of G major, a scale built on G, and then in measures 4 and 5 modulating to a scale built on C, the key of C major – this simple melody involves motion around the pitch-space dimension: both the motion from note to note (ratio to ratio) and the motion between two reference ratios, the keys of G and C. The whole melodic/harmonic system is nested within a larger (hardware) system, the harmonic series founded on the tune's temporal duration, with the software key system (and the harmonics of timbre) forming nested self-similar patterns – copies of the harmonic series of temporal duration built on higher frequency fundamentals: 48Hz and 32Hz (and 768Hz for the first frequency in the domain of timbre).

A Software Beat

If instead of stepping out the beat of tempo at 1Hz in the thought experiment of *Walking the Beat* above (Figure 5.13), one imagines climbing inside the software system of the key of G major. There one would find a similar software fundamental duration to pace out – the 'beat' of G major: 48Hz – *a harmonic 'tempo'*. And after four bars paced at 48Hz this *software 'tempo'* would decelerate to 32Hz, a 1.5 beat!

This is precisely what we were unable to do at the hardware level above when *Walking the (hardware) Beat* we found that it required more motion than the system within which we were bound would allow, to find a 1.666... beat. This tune in two keys, though a trivial example of the freedom and flexibility which arises at the level of ‘software’ in such a systems, does point to something remarkable: On apparently limited, deterministic, one-dimensional foundations, the *hard-wired* temporal harmonic series produces a route to the creation of novelty and variety through the agency and interaction of *software copies of itself*, nested at higher ratios.

```
                Beat/Tempo <--gap-->   Notes/Harmony    Timbre
Note        hn X X X X X X X X X X X X X X X X X X X X X X X X
Timbre      .  X                   X       X         X     X
Software    .  X                   X       X      h3 X   X
Level       .  X                   X       X      h2 X X          G major
            768X--timbre--------h24X----h16X------h1=768Hz        Nested
C major     .  X                   X       X       X              Software
Nested      .  X                   X       X     X                System
Software    .  X                   X    h3 X   X
System      .  X                   X    h2 X X
            h48X--Gmajor-----------X----h1=48Hz
            .  X                   X     X                        Fundamental
            .  X                h3 X   X                          Temporal
            .  X                h2 X X                            Hardware
            h32X--Cmajor--------h1=32Hz                           System
            .  X                 X
            .  X               X
            .  X             X
            .  X           X
            .  X         X
            .  X       X
            h4 X     X
            h3 X   X
            h2 X X
            h1=1Hz_the_Beat_(quarternote=60MM)________________
```

Figure 5.18 Schematic representation of a system of nested harmonic series – a ‘Tune in Two Keys’. Tempo/ duration, the fundamental level of nesting is on the left hand side (harmonic series built on f=1Hz) with the tonal centers nested within this harmonic series in the middle set at f=32Hz C major and f=48Hz G major; and to the right-hand side, the nested harmonic series of timbre f=384Hz to 1024Hz.

The interactions of software have the potential to generate structure (i.e. the relationship of home and subdominant keys, G and C above) which may accumulate in the system – the sum of past events (beats) – and so provide a context for the present beat. This is rather like the computer, where at the lowest level, a limited set of operations on a limited set of natural numbers is transformed through the mediation of many software layers into a cornucopia of astounding variety. It is as if a hardware system, the fundamental level of nesting, by means of creating *imaginary* characters (i.e. nested series) in its own likeness and allowing them to play-act on the stage of the harmonic series, finds a way to transcend its own confines in a ‘musicke of division’. And one imaginary character (key) might well imagine another doubly imaginary character of its own, that would look upon its (imaginary) creator as *real and* *fundamental* – just as a program running on an emulator sees software as hardware, thus opening up a vista of ever wider ranging nested structures. In such an ideal system, music's software layers – harmonics of timbre, nested in scales of twelve notes, nested in a system of twelve key-centers – could be conceptualized as running on the hardware platform of temporal duration, the system's fundamental level. Yet of course, we in the 'Great World' outside know this hardware platform of tempo is just a toy, a software emulation (of our view?) of the real thing – time.

```
  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
  X X X X X X X X X X X X X X X X
  XX  XX  XX  XX  XX  XX  XX  XX
  X   X   X   X   X   X   X   X
  XXXX    XXXX    XXXX    XXXX
  X X     X X     X X     X X
  XX      XX      XX      XX
  X       X       X       X
  XXXXXXXX        XXXXXXXX
  X X X X         X X X X
  XX  XX          XX  XX
  X   X           X   X
  XXXX            XXXX
  X X             X X
  XX              XX
  X               X
  XXXXXXXXXXXXXXXX
  X X X X X X X X
  XX  XX  XX  XX
  X   X   X   X
  XXXX    XXXX
  X X     X X
  XX      XX
  X       X
  XXXXXXXX
  X X X X
  XX  XX
  X   X
  XXXX
  X X
  XX
__X_____________________________________
```

Figure 5.19 The cellular automata output for rule102 (Binary 0110 0110) under 180 degree rotation displays a similar structure of nested triangles to that shown in Figure 5.18.

These various levels of nesting can be charted schematically as overlapping triangles (Figure 5.18) producing a pattern similar to the output of some cellular automata, for example, rule102 illustrated in Figure 5.19. The top right triangle in Figure 5.18, represents the timbre of the first note of the tune in Two Keys (based on G-768Hz). This triangle moves about with each note in the tune (ignoring the more static formant element of timbre). Other triangles in Figure 5.18 represent the key of G major based on the harmonic series of G-48Hz, which after 4 bars changes to the key of C major based on the harmonic series of C-32Hz. And finally the triangle of the whole pattern, the temporal hardware system based on the harmonic series of the beat at metronome MM60, 1Hz.

Connections

Finally, and rather adventurously, we may borrow the concept of *invariance* from Relativity Theory, where it is used to describe the sum of all an object's motion through *spacetime*, and apply the notion to music. But first a little about invariance itself. The term spacetime denotes the combination of the three space and one time dimensions, and in this geometric approach, time is portrayed as equivalent to distance or length, through the connection of the fixed speed of light, i.e. distance divided by time. The spacetime model was famously introduced in a lecture in 1908 by Herman Minkowski, one of Einstein's teachers[3]. And the aggregate of all spacetime, past and future, is invoked here as forming a relational system somewhat analogous (though hugely more extensive and complex) to a complete tonal composition.

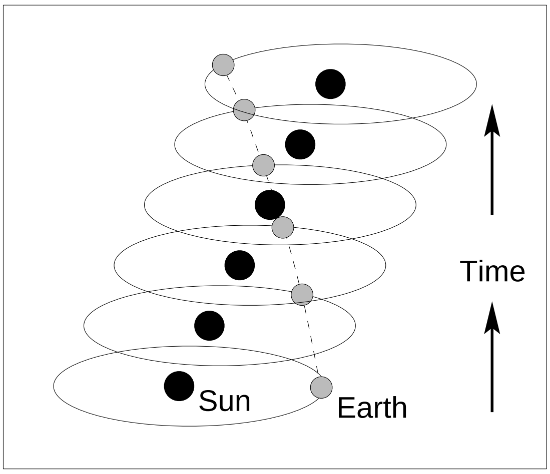

Figure 5.20 Time slices of the Earth's orbit viewed as an elongated spiral through spacetime. (Not to scale.)

Viewed from the geometric four-dimensional perspective of spacetime in relativity theory, a slow-moving object such as the planet Earth, expends most of its total motion travelling through the time dimension. In contrast, photons – particles of light – move around the spacial dimensions so rapidly that they have no motion leftover to travel through time (from their own perspective). To visualise this, because the ellipse of the Earth's orbit about the Sun is essentially two-dimensional, the time dimension could be substituted for the normal vertical (third) spacial dimension, yielding a structure of one second interval time-slices stacked on top of each other, rather like a set of film frames. In this '3-D' heap, the Earth's motion around the Sun describes a spiral up through the one-second time-slices. Travelling through time together, the Sun remains at the center (a foci) of each ascending time-slice, while the Earth's orbital motion moves our planet's position gradually around it in a spiral. Although this seems quite weird, the link between motion in the spacial dimensions and the passage of time, lies in the connection between the distance travelled in a unit of time – at light speed, approximately 186,000 miles per second. And from this huge number it can be seen that the spiral motion of the Earth in spacetime is enormously elongated. In each second the Earth travels along its orbit approximately 18.5 miles, which leaves roughly 185,999.9991 miles between the time-slices! (See Figure 5.21, calculated by treating the Earth's spiral track as something like the hypotenuse of a narrow right-angled triangle, and the orbital distance travelled as one, much lesser, side.) Next imagine the Earth's motion increased to near light speed – Figure 5.21 top left. Now instead of a narrow upward pointing triangle being formed between the space and time dimensions, a narrow flat sideways-pointing triangle will result from most of the motion being transferred into the spacial dimensions. Taken to the limit, at light speed in the spacial dimensions, the triangle is flattened completely and no motion occurs in the time dimension at all – Figure 5.21 bottom left. Or conversely, were all motion to cease in the spacial dimensions, the triangle is flattened to a vertical line in the time dimension (not illustrated).

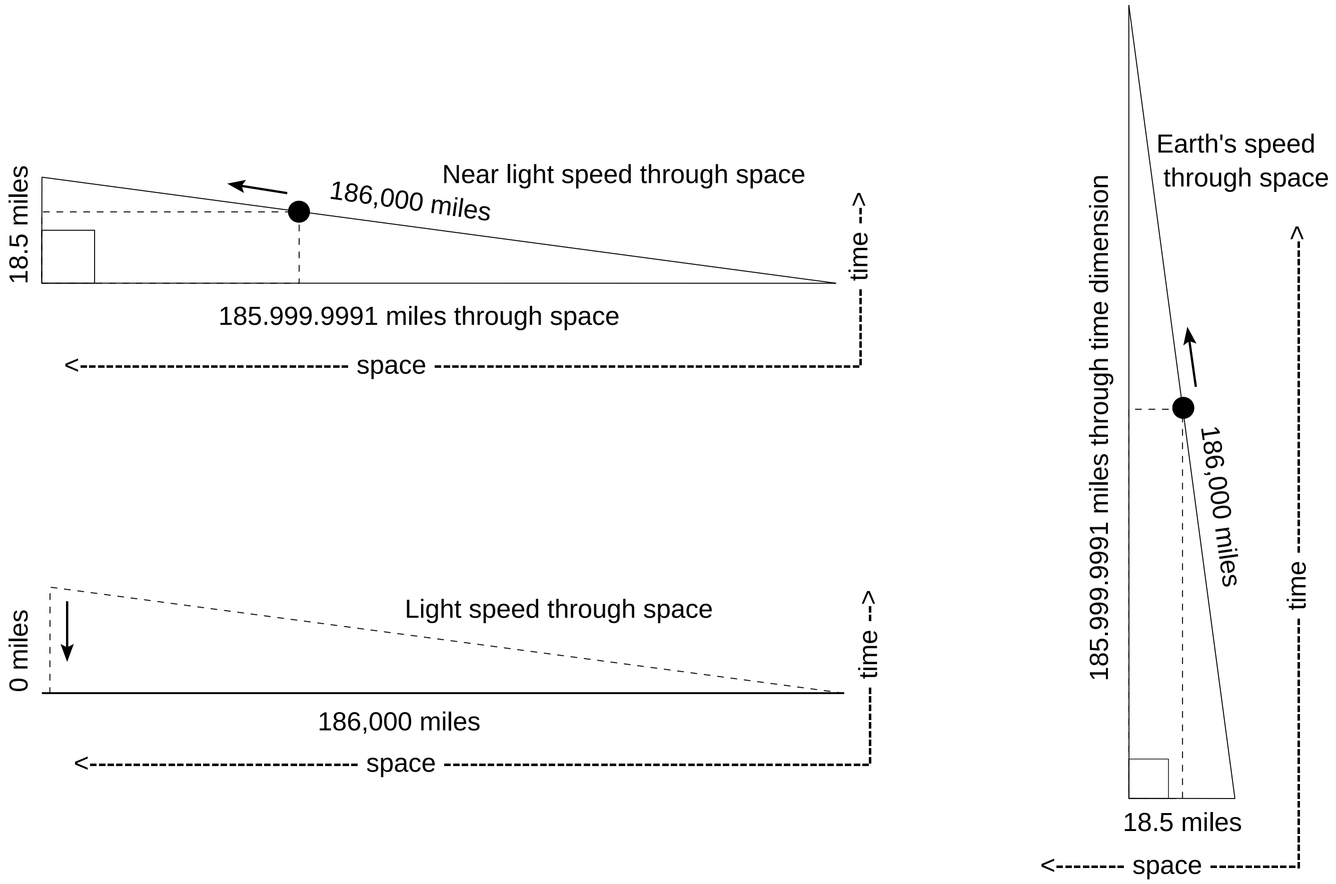

Figure 5.21 The Earth's rather slow motion around the Sun implies a large distance is travelled in the geometrically equivalent time dimension (right triangle). In contrast, electromagnetic radiation, which includes light, travels so fast through space that it has no motion in the time dimension (bottom left triangle).

Now applying the concept of invariance to a '2-D' model consisting of a pitch-space dimension and tempo dimension, i.e. musical time. For example, if a melody descends in pitch with the trajectory of an inverted harmonic series, as in the first six notes of the *Walking the Beat* (Figure 5.13 – B, B, E, B, G, E descending) its relationship to the expanding wave of the whole piece will remain constant – Figure 5.22.

Beat	Note-Hz	Whole-Hz	Ratio
1	B 960:	1	960:1
2	B 480:	0.5	960:1
3	E 320:	0.333	960:1
4	B 240:	0.25	960:1
5	G 192:	0.2	960:1
6	E 160:	0.166	960:1

Figure 5.22 The frequency ratios of the descending melody B, B, E, B, G, E from Figure 5.13, relative to the frequency of the expanding wave of *Walking the Beat*, from beat 1 to 6.

The melody, though moving rapidly by descending through pitch-space, maintains a unchanging relationship with the tempo dimension as a whole – Figure 5.23.

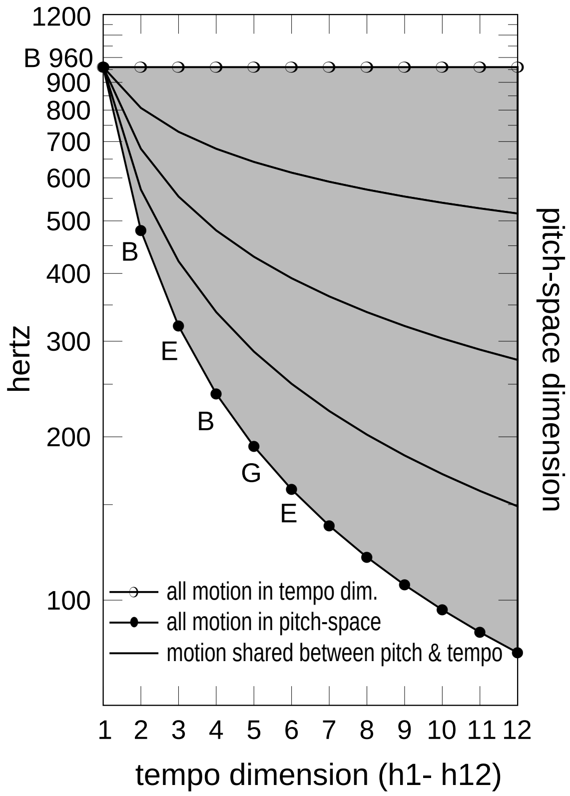

Figure 5.23 Motion in the tempo and pitch-space dimensions of music. The shaded area is analogous to 'light cones' in relativity theory.

Similarly a long held or repeated note, like the top notes in *Walking the Beat*, will have no motion in the pitch-space, all its motion being devoted to travelling through the relationships of the tempo dimension as it carries forward the work of the first beat – Figure 5.24.

Beat	Note-Hz	Whole-Hz	Ratio
1	B 960:	1	960:1
2	B 960:	0.5	1920:1
3	B 960:	0.333	2880:1
4	B 960:	0.25	3840:1
5	B 960:	0.2	4800:1
6	B 960:	0.166	5783:1

Figure 5.24 The frequency ratio of the repeating (first) note 'B' to the frequency of the expanding wave of *Walking the Beat*, from beat 1 to 6.

Between these two extremes, motion (i.e. changing relationships), are shared between the two dimensions of pitch and tempo in music, and so, in a manner reminiscent of relativity, motion in one dimension will have an impact upon relationships with the other dimensions. In this way the patterns and events in this simple model (and less rigorously in music generally) appear to be very much woven together into an interconnected whole: the work of past generations (beats) of the pattern travelling forward along the ratios of tempo/duration to inform and enrich the present beat with context and variety.

Returning briefly to the spiral model of the Earth's orbit about the Sun through spacetime. If an observer were to look down upon the model from a good distance above, the motion of the Earth would appear to be that of its more normally imagined planar ellipse. Somewhat analogous to these two views of the Earth's motion – the vertical and horizontal perspective – the spiral of true fifths of ratio 2:3 would, given an appropriate corkscrew-angle to absorb the difference in 'relational length' between the true and tempered ratios, similarly appear as a closed cycle of tempered fifths (1:1.49...) when viewed from above and a spiral when viewed from the side.

Conclusion

The topics covered in this chapter have been somewhat diffuse and exploratory in character, and they may require a little thoughtful reflection to fully appreciate. Essentially, through the examination of analogous systems to music, and the investigation of the familiar in music by unfamiliar means, hopefully, new perspectives and approaches may be imagined and invoked. Of course, there is no reason for the 'useful' harmonics of the tune in two keys to emerge from all the other ratios; the appropriate numbers to make natural scales built on C and G were just selected. And these rather far-fetched thought experiments are simply attempts to explore something of the nature and characteristics of perhaps more generally applicable principles also relevant to the full understanding of tonal music. Of particular interest are the connections between different levels or domains within systems and how, by approaching apparently different phenomena in abstract and imaginative ways, an overarching unity may be perceived. However, though music's patterns are imbued with elements of self-organisation arising from their internal tonal logic, they are ultimately driven by composers and musicians arbitrarily from outside the system; and so these external operatives have access to any ratio, even fractional ones, of pitch or duration whether occurring naturally from within the system or not – e.g. the 1.666 beat or the equal-tempered scale. In Chapter 15 of *Journey to the Heart of Music* the subject of how 'software' based on 'tonal logic' might function within a system, without intervention from outside agents, will be explored.

With regard to tonal music, overall, the pursuit of recursive algorithms applied to nested structure tends to produce schemes consisting of a fundamental series based on a unit of temporal duration, which is driven forward through a sequence of ratios (h1, h2, h3, etc.) as the system expands, and which is capable of accommodating, at higher frequencies, nested (software) copies of itself. For example, in a manner similar to the nested hierarchies of triangular patterns produced in the cellular automata, e.g. h2, h4, h6, etc. or h300, h600, h900, etc., nested within a fundamental harmonic series of H1, H2, H3, ..., Hn. It is the 'computations' of these higher level nested series – essentially the harmonic progressions found in tonal music – which provides the motive energy, in principle, to drive the whole system forward. Yet, ultimately, all these higher level relationships can be reduced to the ratios of one extensive fundamental series. (The equivalent of reducing multi-column mutable numbers to a single column, wholly additive, numbers.) All the ratios outlined as triangles of Figure 5.18 could be pushed leftward into the fundamental (hardware) series column, the column built on C-h1(1Hz) – though in so doing, structure, variety and differentiation are crushed into homogenous uniformity.

In Chapter 7 an AWK script charting the systematic nesting of harmonic series and the patterns that arise between these nested series will be introduced and investigated. The text output generated by this script, *th.awk*, I call the *Table of (Nested) Harmonic Series,* often abbreviated to THS; alternatively, I would happily accept the title *Table of Mutable Base Numbers* – though mathematicians will undoubtedly prefer its original name: *The Sieve of Eratosthenes*.

Notes

1. Barbour, J.B., *The End of Time,* (Weidenfeld and Nicolson, London, 1999).

2. Wolfram, S., *A New Kind of Science*, (Wolfram Media Inc., Champaign, IL. 2002).

3. Presumably by this time, Minkowski had revised his earlier assessment of the student Einstein as “a lazy dog”! The probable accuracy of his judgement at the time it was made, notwithstanding, the character of Einstein’s intellect was also thoroughly dogged in another way, in that once he became engaged with a problem in physics, terrier-like, he wrestled with it relentlessly, applying his remarkable powers of creative thought.