# MUSICAL APPLICATIONS OF FRACTAL GEOMETRY

Derek White

Math 150 - Fractal Geometry

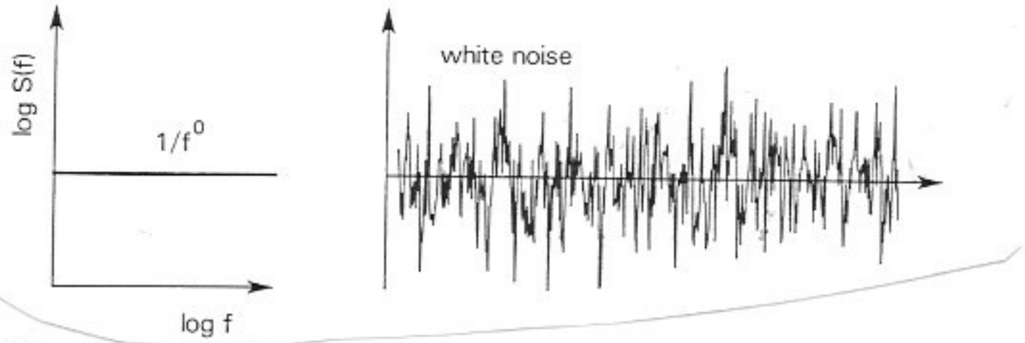
Spring 1989

## INTRODUCTION

When you stop to think about why music is pleasing to us it is utterly mindblowing; many changes in sound pressure in certain patterns somehow induce moods and feelings. The inherent qualities of self-similarity and the fine balance between randomness and order should lead one to believe that fractal geometry is ideally suited for music. In this paper I will discuss some of the applications of fractal geometry to music composition, theory and acoustics. It will be a review of research already done in this field as well as a few of my own ideas.

## NOISE

The distinction between music and noise varies with the taste of the listener though a good song is generally agreed upon. What is this distinction in mathematical terms? Well consider completely random noise (like the stuff that goes along with the "snow" on your television), sometimes called "white" noise or Gaussian noise. The spectral density, ususally denoted Sv(f) where f is the frequency, is a useful tool in analyzing noise. It is a measure of the mean square variation <V > within a bandwidth centered on f where V(t) is the quantity fluctuating in time, in this case the sound pressure, or the voltage signal leading to a speaker. The spectral density of "white" noise is a flat line meaning it represents all frequencies equally, or is correlated from point to point.



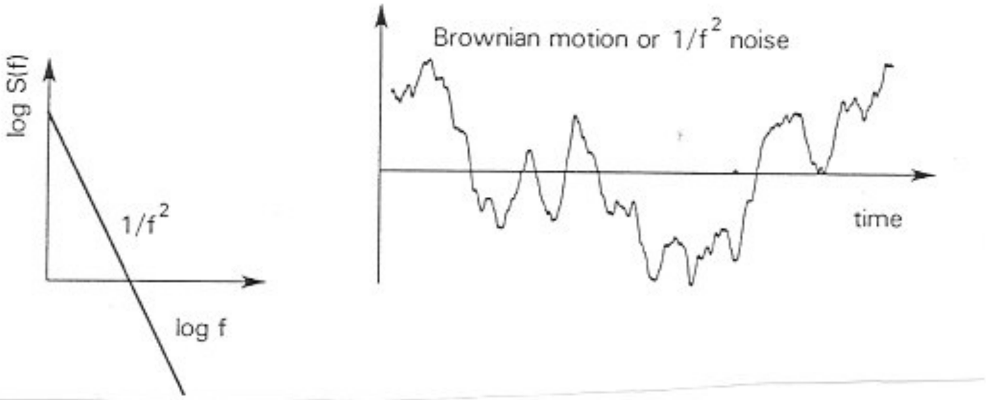
FROM [PEITGEN]

The behavior of V(t) can also be characterized by the <V(t)V(t+ )>, the autocorrelation function which tells how the the fluctuating quantities at times t and t+ are related. It can also be expressed in terms of Sv(f) as stated by the Weiner-Khintchine relations:

$$<V(t)V(t+ \ )> = \ Sv(f)\cos(2 \ f \ )df$$

The slope of Sv(f) for white noise is constant, which shows that there is no correlation on time scales of 1/2 f.

Integrating white noise yields what is known as "Brown" noise, so called because it represents brownian motion or the random walk. Its spectral density goes as 1/f , and the steep slope of the graph of Sv(f) implies that it is highly correlated, that is there is not much change in frequency in a time scale of about 1/2 f.

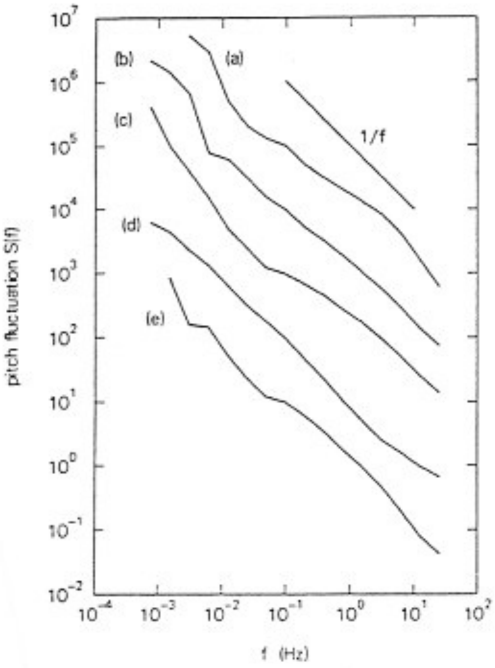


From [peitgen]

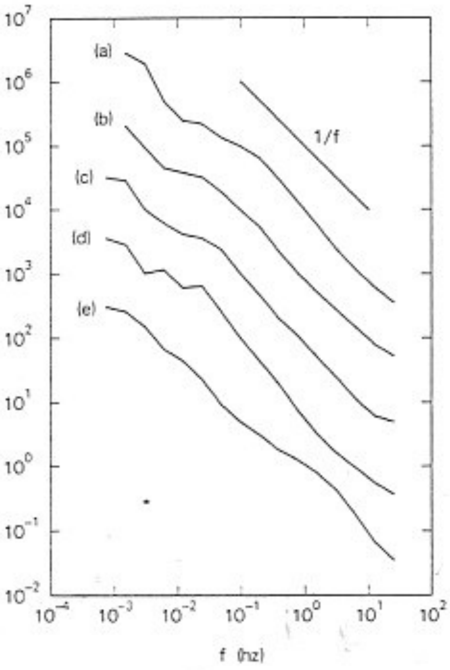
Inbetween these two extremeties of completely random white noise and the the very predictable brown noise is what physicists refer to as 1/f noise or flicker noise. More specifically it is a fluctuating quantity V(t) whose spectral density Sv(f) varies as f where 0.5 < < 1.5. Flicker noise has been studied extensively since it seems to be very common in nature, yet there is no complete theory that explains its origin.

From [peitgen]

One would expect music to fall into this category since it is a pleasing mixture of randomness and predictability. Indeed it does as discovered by Richard Voss and John Clarke. They measured variations in frequency fluctuations and audio power and found their spectral densities to vary as 1/f. They did this by converting either the fluctuations in frequency or variations in audio power into a voltage. Sv(f) can be measured by passing the voltage signal V(t) through a tuned filter of frequency f and bandwidth df. The average of te squared output of the filter divided by df yields Sv(f). Amazingly this 1/f correlation is characteristic of virtually all music from differnt cultures, from Bach to Beethoven.



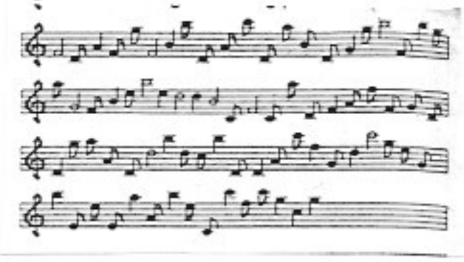
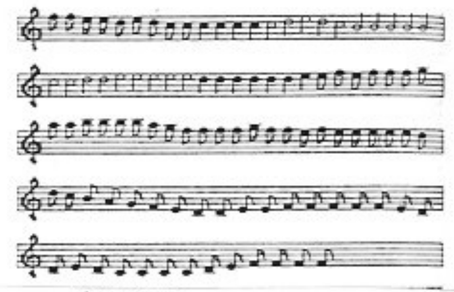
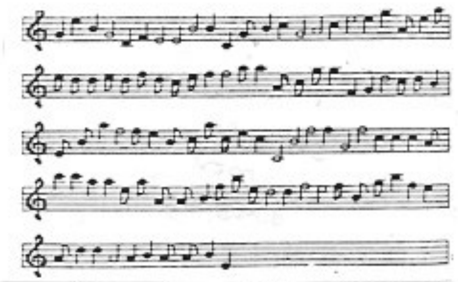
Pitch fluctuations from different musical cultures
- (a) the Ba-Benzele Pygmies
- (b) traditional music of Japan
- (c) classical ragas of India
- (d) folk songs of old Russia
- (e) American blues



Pitch fluctuations in western music
- (a) Medieval music up to 1300
- (b) Beethoven, 3rd Symphony
- (c) Debussey, piano works
- (d) R. Strauss, ein Heldenlebe
- (e) the Beatles, Sgt. Pepper

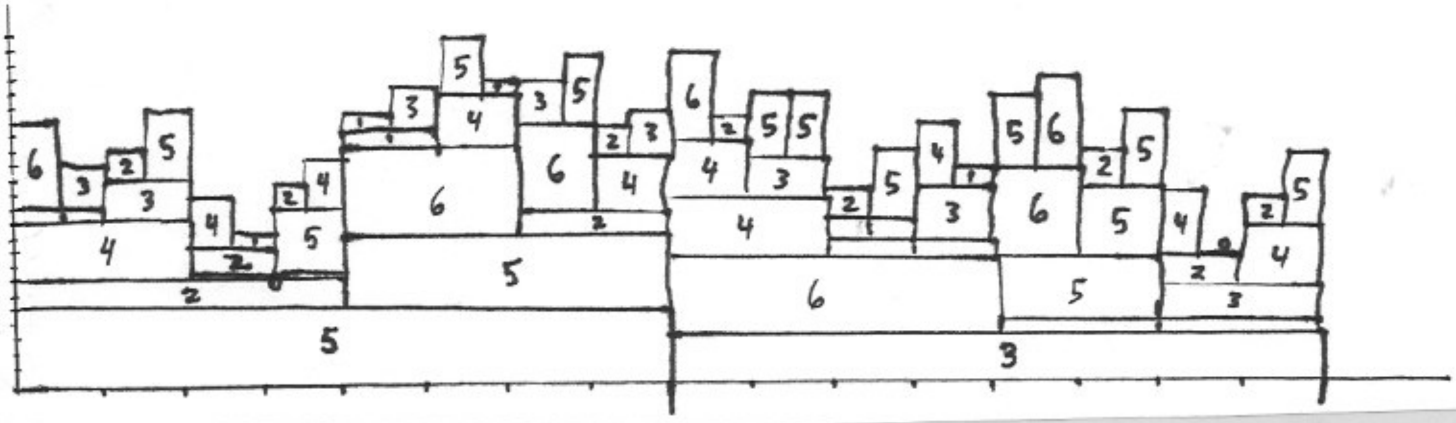
From [peitgen]

"The measurements suggest that music is imitating the characteristic way our world changes in time." [Peitgen]

Voss also used this idea to compose "1/f music". Basically he used a random number generator based on a 1/f noise source. 1/f noise noise was taken from the voltage fluctuations across a current-biased transistor, then digitally sampling it thusly converting it into a series of numbers. These numbers were then rounded and scaled to represent notes of a standard musical scale over a two octave range. The duration of the notes were also gotten in a similar manner. This method is analogous to taking an empty music score, laying it over the trace of 1/f noise, and assigning notes at sucessive intervals. They also did this for white noise and brown noise for comparison. They played it for many people and it was generally agreed upon that the 1/f noise sounded most like "music".

WHITE

BROWN

1/f from Science news vol. 117

I also made a program that lets the user select between melodies generated with constant spectral density variation, 1/f variation and 1/f variation. The program for white noise based melodies simply selects note frequencies totally at random. Since I also had a choice of the duration and length of the note I generated these in a similar manner. The notes can be chosen from either a twelve-tone chromatic scale, major scale, minor scale or the Slendro scale from Indonesian Gamelan music. For the brownian variation I used a algorithm similar to the one used by Dodge (See reference), where the notes have variation but in smaller intervals. The 1/f noise was harder and I had to rely on an approximation that was based on an idea of Voss in [Bolognesi]. This algorithm is also interesting because it displays a lot of self-similarity. The idea is to generate random numbers in a hierarchy. Think of it as rolling four die, one of them you reroll only every eigth time, another you reroll every fourth time, another every second time and the last you would reroll every time. All four die are added up after every turn (even if only one was rethrown) and are assigned to note values. In this case there are four dice with seven sides (0-6) that can generate 25 possible values. Each level has a structure similar to the overall structure.

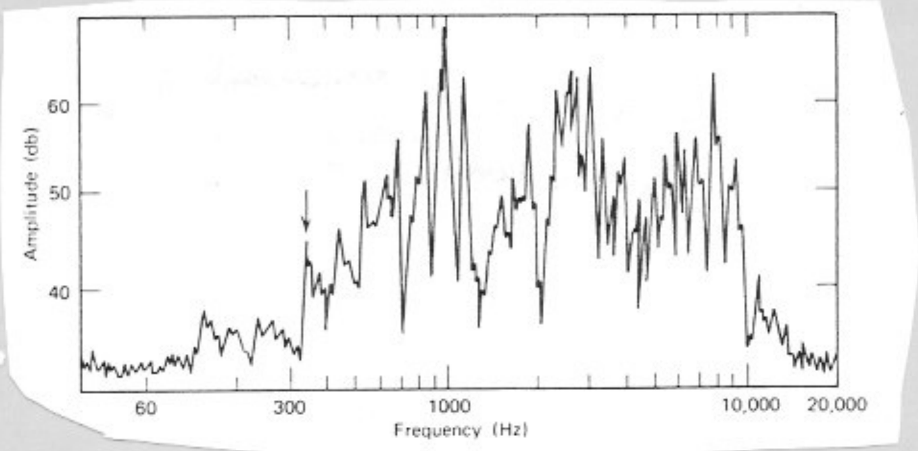
Example



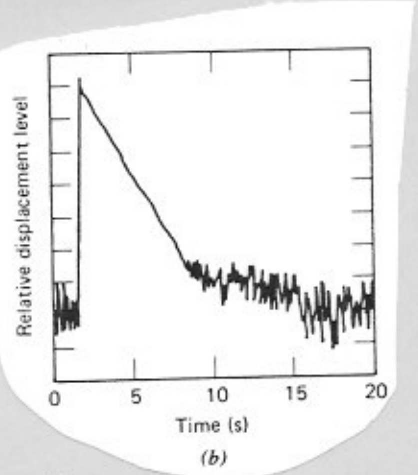
Although this method is slightly limited in this sense that it only produces a monophonic melody, it still shows us that the mixture of correlation and randomness inherent in 1/f noise sources applies most closely to the degree of predictability in which we enjoy our music.

## DIRECT 1/F NOISE

If you look at single notes by various instruments they also display a trace similar to the trace of 1/f noise:



Violin response Curve
from physics today, 1968.

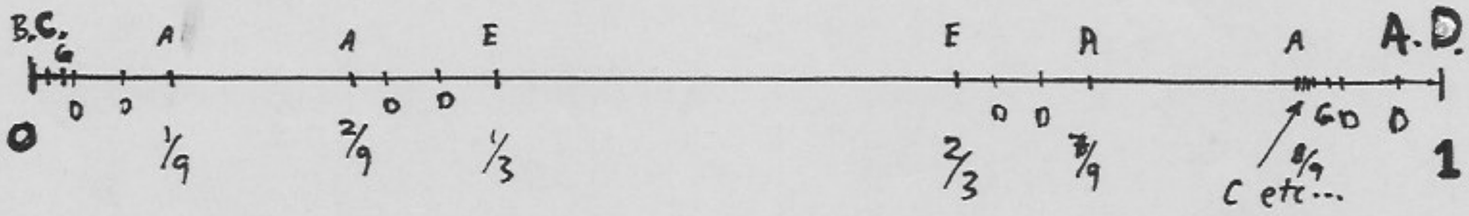


Piano

Adding up notes from different instruments, especially when chords or harmony are also played, would give us a very complex signal, but it would still be expected to show self-similarity. When you look at music through an oscilloscope it does resemble the trace of 1/f noise. Therefore rather then use 1/f noise to drive a random number generator to pick notes in a melody, I would think it would be interesting to hear the 1/f noise direcly. The problem with this is that it takes up massive amounts of computer time and memory. I tried to produce just the inormation for a few seconds of 1/f noise and it took about fifteen minutes. According to the sampling theorem, to record one second of sound whose bandwidth is about 20,000 Hz would require storing 40,000 samples (this is also the standard set by compact disk manufacturers). A random number generator could be used to create these numbers which would represent the amplitude of the soundwave. The reason this would be interesting is that it would show fractal complexity and self-similarity at all levels, not just on the basis of notes selected from our limited 12-note chromatic scale.

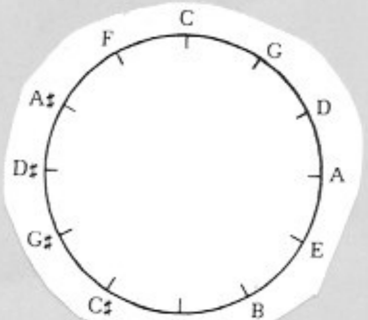
## THE CANTOR SET AND THE CIRCLE OF FIFTHS

Fundamental in music composition is the concept of a scale, i.e. certain notes when played together set the mood of the song, and which culture it is from. If one examines the roots of songs, especially folk music, they usually follow the major triad. For example a song in the key of C major would contain mainly the chords Cmaj, Fmaj and Gmaj. Middle C is defined as having a frequency of 261.6 Hz, F is 349.2 Hz and G has a frequency of 392.0. If you double or quadruple one of these you will still retain the same note, just an octave higher. These notes follow the pattern that C = 2/3G (2/3 times 392 is 261.3) and F = 2/9G (2/9 times 392 is 87.1, which is F two actaves lower then the one above. In other words they follow a pattern similar to the cantor set, one of the first and most famous examples of an object with fractal dimension. Furthermore, if you use B as the base note of the cantor set where B = 247, then by taking 2/3 B where = 1,...,11 you would generate the following sequence of frequencies: 329.3, 439.1, 292.7, 390.3, 260.2 and 346.9 which are represented by the sequence of notes B,E,A,D,G,C and F, which are the notes in the key of C. Extending this further we would get all the notes of the chromatic scale.



In other words the chromatic scale follows the reiteration scheme X = 2/3X where = 1,...,11, which generates the chromatic sequence in the following order: C,F,A#,D#,G#,C#,F#,B,E,A,D,G amd C. Arranged these in a circular would give us the "circle of fifths", fundamental to music theory.

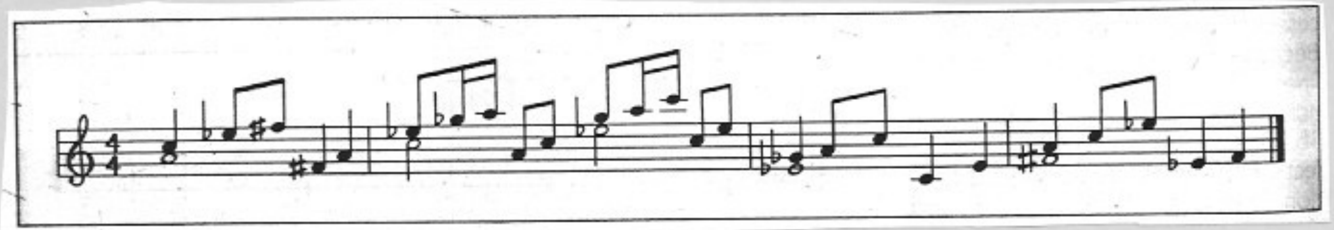
you can say that I reinvented the wheel, no pun intended. Thus our chromatici scale which contains

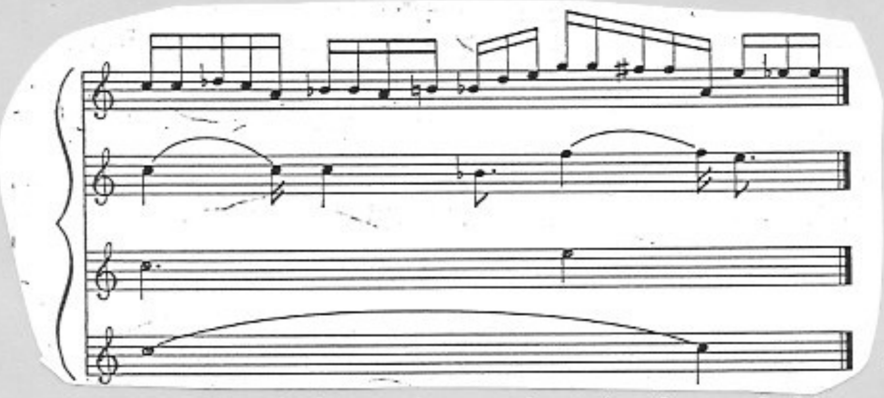


circle of fifths

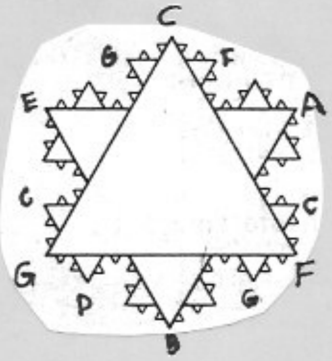
all the of ours as well as most other cultures scales has a fractal basis. Not only is it self-similar in the usual way of the cantor set but after 12 iterations the sequence repeats itself.

## COMPOSITION BY KOCH CURVE

The use of 1/f noise as well as other means of stochastic note generation to make songs neglects an important aspect of musical composition, musical structure. Charles Dodge and Curtis Bahn were able to create polyphonic compositions based on the Koch curve which displayed a hierarchy of structure characteristic to good music. They took the idea of the Koch snowflake and applied it to compostion in the following manner. They chose a "generating motif" as a basis or the slowest moving line. This can be thought of as the largest triangle in the Koch snowflake. Then they created a faster repetition based on the same motif which they layered over the first, and so on, just as you would with the Koch curve. "The result is a time filling musical structure analogous to the space-filling curve of the Koch snowflake".

from Byte June 86

To elaborate on this idea and make a composition that sounded less robotic, they also made random offsets of the original using a 1/f noise based algorithm that produced selfsimilar patterns.

from byte June 86

This is more like compositions from classical to folk music. In classical music, a composition is usually divided into several movements specified by a change in key, mood, tempo, etc. Folk and most modern music also can be broken down into such a hierarchy (denoted by chorus, bridge, etc.). Maybe this idea could be extended further, the generating motif could consist of the major triad, for example in the key of Cmaj it would be Cmaj, Fmaj and Gmaj. Each of these sections could be broken down into the notes of that particular chord, Cmaj consists of the notes C,E and G, Fmaj is made up of the notes F,C and A and Gmaj has the notes G,B and D. A faster repetion with 1/f variation of this could reveal the melody.



## WEIERSTRASS FUNCTIONS

Another interesting application of fractal geometry to music lies in Weierstrass functons [Schroeder]. These are functions that were defined by Karl Weierstrass that had boundaries that were everywhere continuous but nowhere differentiable. Wierstrass functions can be used to pattern a chord which when played back at twice the speed will sound a semitone lower instead of an octave higher like a normal chord would. To show how this is true let us construct a simple weierstrass function:

$$w(t) = \sum_{k=-\infty}^{\infty}{}' \cos(\gamma^{k} t)$$

If instead of t we insert $\gamma$t into this equation we would get:

$$w(\gamma t) = \sum_{k}{}' \cos(\gamma^{k+1} t) = \sum{}' \cos(\gamma^{k} t)$$

Note this consists of all the notes in the scale of Cmaj.

i.e. by scaling the time by $\gamma$ we get back the original function. Now if we scale the time in the same manner that we scale our chromatic scale, namely let $\gamma = 2^{13/12}$, then:

$$w(t) = \sum_{K}{}' \cos(2^{k\,13/12} t)$$

If we think of this as an audible sound and played it back at twice the speed (you could record it on magnetic tape then double the tape speed) we would get:

$$w(2t) = \sum_{K}{}' \cos(2^{K\,13/12} t) = \sum_{K'}{}' \cos(2^{K'\,13/12} 2^{-1/12} t)$$

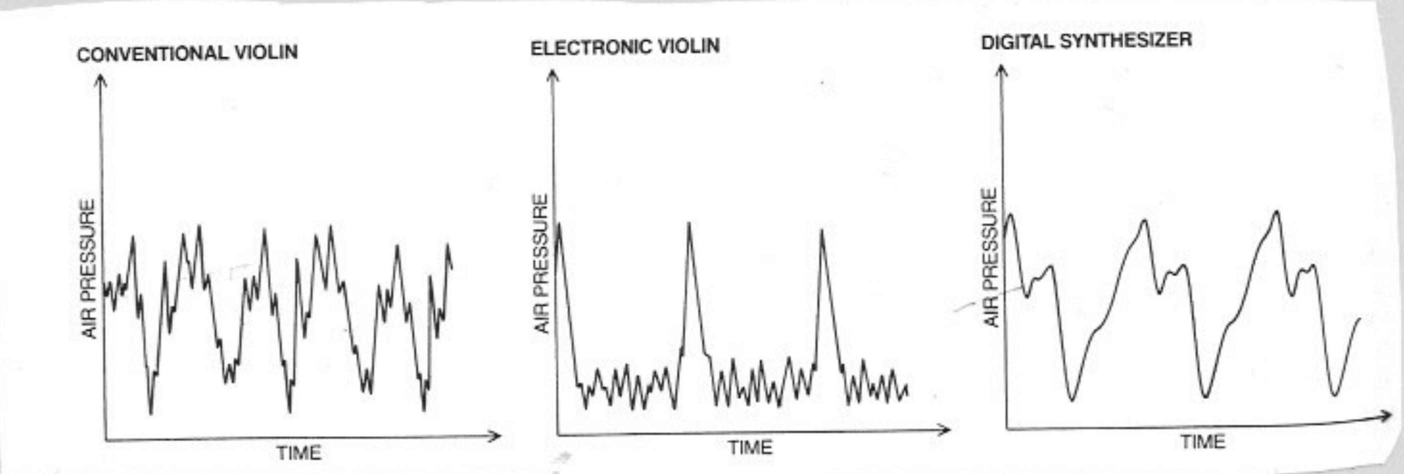
If the k's are chosen to cover the entire audible range then it simplifies to:

$$w(2t) = w(2^{-1/12} t)$$

Thus the sound will be lowered by $2^{-1/12}$ or a semitone when you double the speed.

## FRACTAL DIMENSIONS OF MUSICAL INSTRUMENTS

The instrument on which music is played is also of importance to the overall enjoyment of the music. Anybody can tell the difference between a stradivarius and a synthesized violin, and it is usually agreed that the stradivarius sounds better. Even a good synthesized or sampled violin is just an approximation to the real thing:

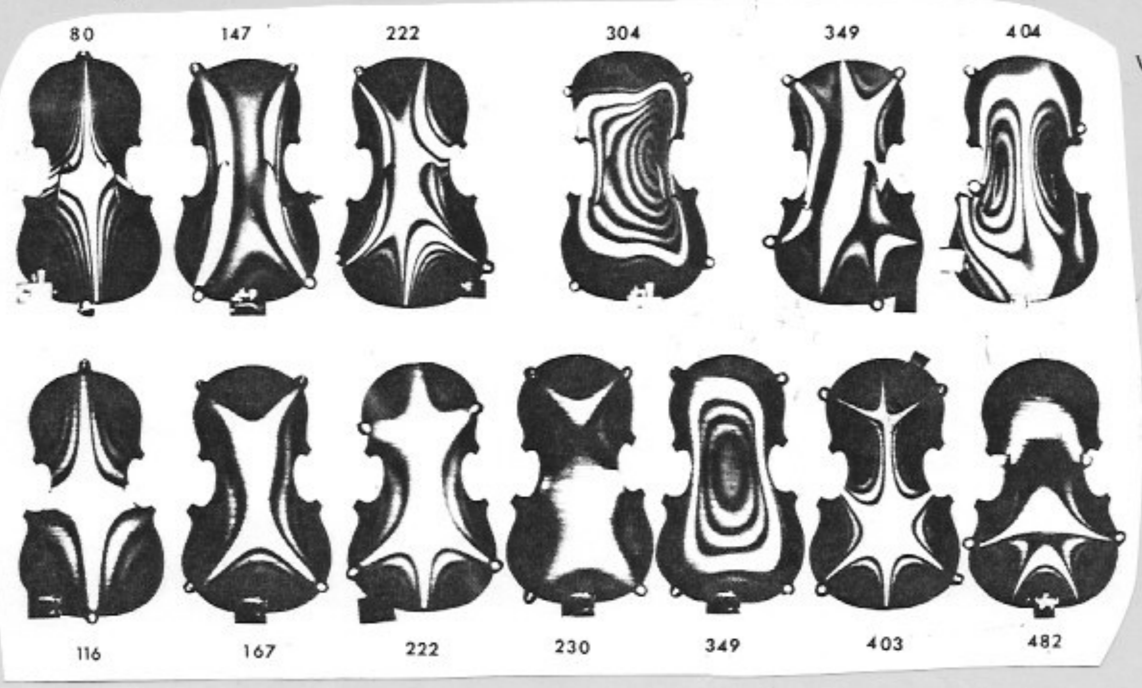


from Scientific American Feb. 1987

This complexity of the signal adds yet another fractal layer of self-similarity to the overall structure of a song.

The tension of a string on a violin or any other stringed instrument determines the fundamental frequency. But harmonics of the fundamental frequency, or vibrations that occur at integer values times the fundamental frequency, can also be generated. The wave equation d f/dt = f, where f = f/ x + f/ y + f/ z, is useful in studying the acoustics of musical instruments and actually was invented with that purpose in mind by Euler in the 18th century. The characteristic frequencies of an instrument are given by the eigenvalues of f, the Laplacian, or the solutions of f + f = 0, where are eigenvalues. Hermann Weyl made a study of instruments and concluded that vibrating objects (with smooth boundaries) have a spatial distribution of characteristic frequencies that obey regular laws. Namely, let N( ) be the number of characteristic frequencies less then a value , and k be the dimension of the vibrating object in question. Weyl concluded that N( ) is approximately proportional to , where the constant of proportionality is related to the volume of the object. As goes to infinity this approximation becomes better and better.

In 1980 Micheal Berry improved on this approximation by determining the degree of error in this approximation. His idea was that the error in Weyls depended on the boudary effects and was on the order of , where d is the dimension of the boundary. This conjecture was not limited to objects with smooth boundaries, Berry suggested that it should apply equally well to objects with fractal dimension. In this case more vibrations would be possible, since if you looked at the finer detail there there would be many crevices for higher frequencies to penetrate. Thusly N( ) would be expected to be larger. For example consider a circular drum whose boundary dimension is ideally one. If it is more realistically treated as an object with a fractal boundary, then its Hausdorff dimension would be slightly more, maybe somewhere between 1 and the Hausdorff dimension of the Koch curve, 1.2618. Since d>1 then > , or the error in Weyls formula would be less; more frequencies would be generated. Unfortunately this was later disproved by R. Carmona in 1986, but just recently Ladipus and Fleckinger-Pelle proved that it would hold if the Minkowski dimension was used in place of the Hausdorff dimension. This is probably due to the fact that Minkowski dimension includes the behavior of points near the boundary unlike the Hausdorff dimension, and what matters in acoustics is the vibration of strings (or drumheads etc.) near the boundary but not on the boundary itself.



Vibrational frequencies of violin

## CONCLUSION

Just as fractal geometry was devised to explain the irregular and complex nature of shapes in the real world, it can be similarly be applied to music and its composition. The next step may be to make musical compositions using iterated function systems. For example, just as a song usually ends in the note of the key it is in, and their is a certain tension until this is achieved, an iterated function system has its attractor that it is pulled to. Hopefully I will be able to pursue this more.

```
{Comparison of different types of randomly generated melodies}
{Derek White, Math 150, Spring 1989}

program musictals (input, output);

 var
  frequencies, loudly, plucked, held : array[1..25] of integer;
  count, pluck, hold, loud, dumb, choice, scale : integer;
  a, b, c, d, p, w, x, y, z : integer;
  i, j, k, l : integer;
  dummy, rx, ry, rz : real;
  quit : string;

 procedure chromatic;      {stores the pitches of the 12-tone }
 begin                     {chromatic scale}
  frequencies[1] := 262;
  frequencies[2] := 278;
  frequencies[3] := 294;
  frequencies[4] := 311;
  frequencies[5] := 330;
  frequencies[6] := 349;
  frequencies[7] := 370;
  frequencies[8] := 392;
  frequencies[9] := 415;
  frequencies[10] := 440;
  frequencies[11] := 466;
  frequencies[12] := 494;
  frequencies[13] := 524;
  frequencies[14] := 556;
  frequencies[15] := 588;
  frequencies[16] := 622;
  frequencies[17] := 660;
  frequencies[18] := 698;
  frequencies[19] := 740;
  frequencies[20] := 784;
  frequencies[21] := 830;
  frequencies[22] := 880;
  frequencies[23] := 932;
  frequencies[24] := 988;
  frequencies[25] := 1048
 end;    {chromatic}
```

```
end;    {gamelan}

procedure flicker;    {algorithm to generate 1/f random melodies}
 var
  count : integer;
  a, b, c, d, w, x, y, z : integer;
begin
 for count := 1 to 25 do
  begin
   for a := 1 to 2 do    {1/f variation in note frequency}
    w := trunc(abs(7 * (random / maxint)));
   for b := 1 to 2 do
    x := trunc(abs(7 * (random / maxint)));
   for c := 1 to 2 do
    y := trunc(abs(7 * (random / maxint)));
   for d := 1 to 2 do
    begin
     z := trunc(abs(7 * (random / maxint)));
     plucked[count] := (w + x + y + z + 1)
    end;    {1/f note variation}
   for a := 1 to 2 do    {1/f variation in loudness}
    w := trunc(abs(13 * (random / maxint)));
   for b := 1 to 2 do
    x := trunc(abs(13 * (random / maxint)));
   for c := 1 to 2 do
    y := trunc(abs(13 * (random / maxint)));
   for d := 1 to 2 do
    begin
     z := trunc(abs(13 * (random / maxint)));
     loudly[count] := 5 * (w + x + y + z) + 1
    end;    {1/f loudness variation}
   for a := 1 to 2 do    {1/f  note duration variation}
    w := trunc(abs(2 * (random / maxint)));
   for b := 1 to 2 do
    x := trunc(abs(2 * (random / maxint)));
   for c := 1 to 2 do
    y := trunc(abs(2 * (random / maxint)));
   for d := 1 to 2 do
    begin
     z := trunc(abs(2 * (random / maxint)));
     dumb := (w + x + y + z);
     if dumb = 0 then
      hold := 10;
```

```
     if (dumb = 1) or (dumb = 2) then
      hold := 20;
     if dumb = 3 then
      hold := 40;
     if dumb = 4 then
      hold := 80;
     held[count] := hold
    end;    {1/f note  length variation}
  end;
end;    {flicker}

procedure brown;    {brown noise based melodies}
 var
  count, j, k, l : integer;
begin
 for count := 1 to 25 do
  begin
   rx := 0;
   ry := 0;
   rz := 0;
   for j := 1 to 8 do     {brownian variation in frequency}
    begin
     rx := rx + abs(random / maxint);
     dummy := (rx - 4)
    end;    {frequency}
   pluck := trunc(dummy) + 12;
   for k := 1 to 8 do     {brownian variation in note duration}
    begin
     ry := ry + abs(random / maxint);
     w := trunc(ry - 4) + 4
    end;
   if (w = 1) or (w = 2) then
    hold := 10;
   if (w = 3) or (w = 4) then
    hold := 20;
   if (w = 5) or (w = 6) then
    hold := 40;
   if (w = 7) or (w = 8) then
    hold := 80;      {duration}
   for l := 1 to 24 do     {brownian variation in loudness}
    begin
     rz := rz + abs(random / maxint);
     dummy := abs(12 * (rz - 24));
```

```
      loud := 5 * (trunc(dummy / 6) + 1)
     end;    {loudness}
    held[count] := hold;
    plucked[count] := pluck;
    loudly[count] := loud
   end;
 end;    {brown}

 procedure white;    {generates completely random melodies}
  var
   count, x, y, z : integer;
 begin
  for count := 1 to 25 do
   begin
    x := trunc(abs(4 * (random / maxint))) + 1;    {random note length}
    if x = 1 then
     hold := 10;
    if x = 2 then
     hold := 20;
    if x = 3 then
     hold := 40;
    if x = 4 then
     hold := 80;
    pluck := trunc(abs(25 * (random / maxint))) + 1;    {random frequencies}
    loud := trunc(abs(240 * (random / maxint))) + 1;    {random durations}
    held[count] := hold;
    plucked[count] := pluck;
    loudly[count] := loud
   end;
 end;    {white}

begin    {main program}
 quit := 'yes';
 writeln('This program is designed to let the user hear the difference');
 writeln('between three types of randomly generated melodies: ');
 writeln('1.white or gaussian noise based melody, whose spectral');
 writeln('density goes as a constant, i.e. totally random');
 writeln('2.brownian noise based melody whose spectral density');
 writeln('goes as one over f squared; very correlated.');
 writeln('3.flicker or 1/f noise, melodies whose spectral density');
 writeln('goes as 1/f.  Just perfect?  Decide for yourself.');
 writeln('You also have the choice of hearing these melodies in');
 writeln('either the 12-note chromatic scale, major scale, minor');
```

646
348
13e)

```
  frequencies[12] := 466;
  frequencies[13] := 524;
  frequencies[14] := 524;
  frequencies[15] := 588;
  frequencies[16] := 588;
  frequencies[17] := 622;
  frequencies[18] := 698;
  frequencies[19] := 784;
  frequencies[20] := 784;
  frequencies[21] := 830;
  frequencies[22] := 830;
  frequencies[23] := 932;
  frequencies[24] := 1048;
  frequencies[25] := 1048
 end;   {minor}

 procedure gamelan;    {stores pitches in gamelan slendro scale}
 begin
  frequencies[1] := 262;
  frequencies[2] := 262;
  frequencies[3] := 262;
  frequencies[4] := 278;
  frequencies[5] := 278;
  frequencies[6] := 278;
  frequencies[7] := 311;
  frequencies[8] := 311;
  frequencies[9] := 3922;
  frequencies[10] := 392;
  frequencies[11] := 311;
  frequencies[12] := 392;
  frequencies[13] := 415;
  frequencies[14] := 415;
  frequencies[15] := 415;
  frequencies[16] := 524;
  frequencies[17] := 524;
  frequencies[18] := 524;
  frequencies[19] := 556;
  frequencies[20] := 556;
  frequencies[21] := 622;
  frequencies[22] := 622;
  frequencies[23] := 622;
  frequencies[24] := 784;
  frequencies[25] := 784
```

```
procedure major;    {stores pitches in major scale}
begin
 frequencies[1] := 262;
 frequencies[2] := 262;
 frequencies[3] := 262;
 frequencies[4] := 294;
 frequencies[5] := 330;
 frequencies[6] := 349;
 frequencies[7] := 349;
 frequencies[8] := 392;
 frequencies[9] := 440;
 frequencies[10] := 494;
 frequencies[11] := 494;
 frequencies[12] := 524;
 frequencies[13] := 588;
 frequencies[14] := 588;
 frequencies[15] := 660;
 frequencies[16] := 698;
 frequencies[17] := 698;
 frequencies[18] := 784;
 frequencies[19] := 784;
 frequencies[20] := 880;
 frequencies[21] := 988;
 frequencies[22] := 988;
 frequencies[23] := 1048;
 frequencies[24] := 1048;
 frequencies[25] := 1048
end;    {major}

procedure minor;    {stores pitches in minor scale}
begin
 frequencies[1] := 262;
 frequencies[2] := 262;
 frequencies[3] := 262;
 frequencies[4] := 294;
 frequencies[5] := 394;
 frequencies[6] := 311;
 frequencies[7] := 349;
 frequencies[8] := 392;
 frequencies[9] := 392;
 frequencies[10] := 415;
 frequencies[11] := 415;
```

BIBLIOGRAPHY

Berry, Micheal, Proceedings of Symposia on Pure Mathematics vol.36, 1980 pp. 13-29.

Bolognesi, Tommaso "Automatic Composition: Experiments with Self-Similar Music" Computer Music Journal, vol.7, no.1, spring 1983. pp 25-36.

Borish, Jeffrey "An Efficient Algorithm for Generating Colored Noise Using a Pseudorandom Sequence" J. Audio Eng. Soc.,vol. 33, no.3, March 1985.

Dodge, Charles and Bahn, Curtis. "Musical Fractals" Byte, June 1986. pp. 185-196.

Gardner, Martin "White and Brown Music, Fractal Curves and 1/f Fluctuations" Scientific American (1978). pp. 14-32.

Lorrain, Denis "A Panoply of Stochastic 'Cannons'" Computer Music Journal, Vol.4, no. 1, 1980. pp. 53-81.

Mathews, Max V. and Pierce, John R. "The Computer as a Musical Instrument" Scientific American, Feb. 1987. pp. 125-133.

Myhill, John "Controlled Indeterminacy: A First step Towards a Semi-Stochasti Music Language" Computer Music Journal, vol.3, no.3, Fall 1979. pp 12-14

Olson, Harry F. and Belar, Herbert "Aid to Music Employing a Random Probability System" J. of Acoust. Soc. Amer. Sept. 1961. pp. 1163-1170.
Peitgen, Heinz-Otto and Saupe, Dietmar "The Science of Fractal Images" (Springer-Verlag, New York, 1988)

Pinkerton, Richard C. "Information Theory and Melody" Scientific American vol. 194 (1956) pp. 77-86.

Schroeder, M.R. "Number Theory in Science and Communication" (Springer-Verlag pp. 315-340.

Stewart, Ian "The Beat of a Fractal Drum" Nature vol.333, May 1984. p. 206.

Thomsen, Dietrick E. "Making Music--Fractally" Science News, vol. 117, March 22, 1980. pp. 187-190.

Van Der Ziel, A. "On the Noise Spectra of Semi-Conductor Noise and of Flicker Effect" Physica XVI, no.4, April 1950. pp.359-372.

Voss, Richard F. and Clarke, John "'1/f Noise' in Music: Music From 1/f Noise" J. Acoust. Soc. Am., vol. 63 no.1, Jan. 1978. pp. 258-263.

Voss, Richard and Clarke, John "'1/f Noise' in Music and Speech" Nature vol. 258, Nov. 27, 1975. pp.317-318.