Listen to the chaotic faucet

Tsung-Lin Hsieh
Department of Physics
University of Illinois at Urbana-Champaign

This project utilizes the chaotic leaky faucet apparatus to generate computer music. Studies have shown that the power spectrum density of human music obeys $1/f^{\beta}$ power law. For the leaky faucet, the time difference between successive drops will exhibit chaotic behavior under specific flow rates, which also obeys $1/f^{\beta}$ power law. The drop signal is imported into a computer which converts it into the pitch and beat of music notes. The sequence of notes is then modified to produce harmony, chord progression and repetition. Listenable music can be created through the process.

Introduction

Human music contains patterns. For instance, harmony is formed by notes whose frequencies have simple integer multiple relation. Other mathematical relations have been used to understand and model music, such as neural topography (Janata et al. 2002) and orbifold space (Tymoczko 2006). In particular, the $1/f$ power law has been found useful to characterize different genres of music (Voss 1978, Hennig et al. 2011, Levitin et al. 2012).

For a time-domain signal $V(t)$, the frequency-domain power spectral density $S_V(\omega)$ is related to the time-domain auto-correlation function $h(t)$ by

$$S_V(\omega) = \int_{-\infty}^{+\infty} h(t)e^{-i\omega t}dt \qquad (1)$$

where

$$h(t) = \int_{-\infty}^{+\infty} V(\tau)^* V(t+\tau)d\tau \qquad (2)$$

$S_V(\omega)$ is an indication of the correlation of $V(t)$, which can be obtained from its Fourier Transform. The $1/f$ power law describes a signal whose power spectral density $S_V(\omega)$, or $S_V(f)$, differ simply by a factor of 2π, obeys the relation

$$S_V(f) \propto 1/f^{\beta} \qquad (3)$$

For white noise, where $V(t)$ has no temporal correlations, $\beta = 0$; for Brownian noise $\beta = 2$, which means that $V(t)$ is strongly correlated.

Voss (1978), Hennig et al. (2011), and Levitin et al. (2012) have shown that in many musical pieces, from classical to rock music, the fluctuation of pitch (frequency), loudness and duration obeys $1/f$ power law. The exponent β ranges from 0.4 to 1.4, depending on the composer and genre. The range of β suggests that human music keeps a balance between predictability ($\beta = 2$) and randomness ($\beta = 0$).

The $1/f$ relation is also observed in several natural phenomena, such as the frequency of earthquakes and the fluctuation of heart beat rate. The chaotic leaky faucet, used in a past UIUC Advanced Physics Lab course, may exhibit chaos in the time difference between successive drops produced because of the standing wave and damping in water (Martien et al. 1985). The time difference follows the $1/f$ power law, and is used as the $1/f$ noise source in this project.

Chaotic faucet setup

Figure 1 shows the chaotic leaky faucet setup. Two water jugs are used to control the flow rate of water. The upper water jug, connected by a siphon to the lower jug, serves as a water reservoir. The end of siphon in the lower jug is attached to a float valve (shown in pink) to keep the water level fixed. Since the drop behavior is highly sensitive to the viscosity, which is related to the temperature of water, the lower jug is covered by a black insulation. The water coming from the reservoir is much fewer than the water inside the lower jug, so the water temperature stays approximately the same.

The bottom of the lower jug is connected to a glass tube, on which a needle valve controlling the flow rate is mounted. The end of the tube (bottom left in Fig. 1) is a nozzle where drops are formed. The nozzle is shielded by plastic funnels to avoid wind perturbation, and a laser beam is set up under the nozzle. When a water drop cuts through the laser beam, the photo-detector at the other end of the beam would receive less intensity and the change is outputted as voltage signal.

The output of the photo-detector is processed by a circuit designed by Professor Steven Errede, and the data is sent to a personal computer with CVI analysis program. To observe the data in real time, the time difference between successive drops is converted to voltage signal and outputted from the DAC1 port of the circuit board. The signal is transmitted to an oscilloscope for visualization (Fig. 2). In addition, the signal is transmitted to a Wavetek function generator and a audio power amplifier, which converts time difference into different frequencies of sound, played by a loud speaker.

Procedure

To start the siphon effect, air is pumped into the sealed upper water reservoir. As the upper reservoir stays over-pressured, the water will continuously flow through the

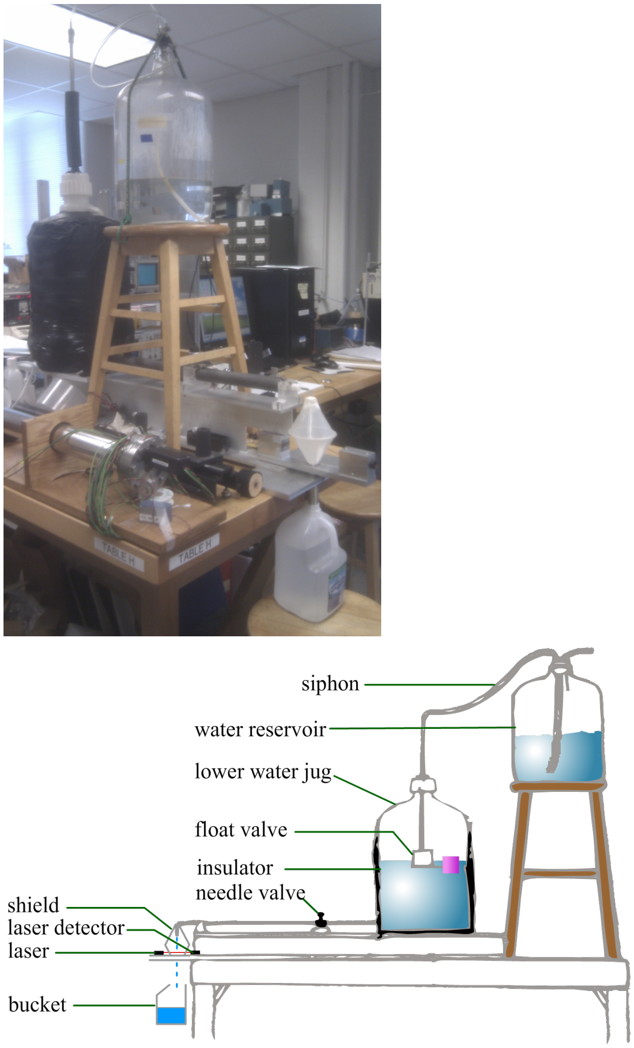

Figure 1. The chaotic faucet setup (top) and the schematic sketch (bottom), adapted from Fregeau (1998).

siphon to the lower jug, which is open to atmospheric pressure. It is critical that the upper reservoir be kept over-pressured until the valve in the lower jug is immersed in water; otherwise air will come into the siphon from the valve. Once the valve is immersed in water and then closed, the upper reservoir will be open to atmospheric pressure. Now the siphon is filled with water and both jugs are at atmospheric pressure.

The needle valve on the glass tube can then be opened to create drops. As water flows out from the lower jug, the water level slightly decreases and the float valve will open to let water come in from the upper reservoir. By the design of the float valve, the end of siphon at the lower jug will remain under water throughout the experiment so that the siphon will continuously take effect. Since the water level is approximately constant, the flow rate will be the same throughout the period of measurement.

After drops are formed, the data acquisition system (laser

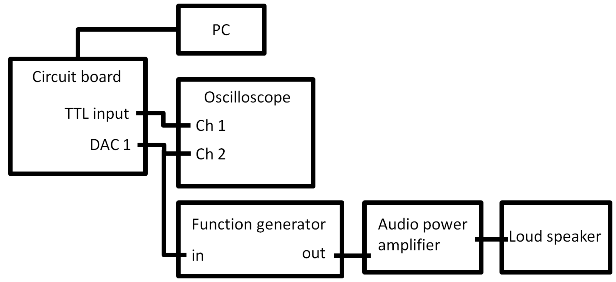

Figure 2. Connection of the data acquisition system.

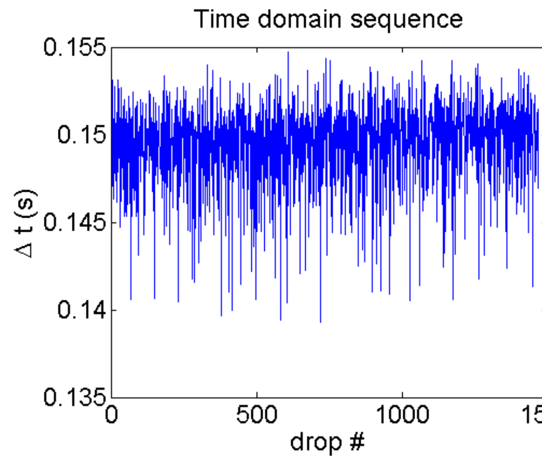

Figure 3. Typical time difference between successive drops when chaotic behavior occurs.

beam, audio power amplifier and the CVI program etc.) is turned on. The output frequency of the Wavetek function generator is adjusted such that the time difference signal is transformed into audible sound. The water flow rate is controlled by the needle valve. If the drops are formed periodically, the sound produced by the loud speaker will have the same pitch. If the flow rate is adjusted such that chaotic behavior occurs, the pitch will vary in a way similar to jazz music. An example is shown in "drop.mid," which consists of notes generated according to the drop time difference. The audio effect makes it easy to find the specific flow rates which result in chaos.

Once the chaotic behavior occurs, the needle valve is fixed for at least 5 minutes, producing more than 1,000 time difference data. At the same time, the flow rate is measured by weighting the drops produced in 5 minutes.

Data analysis

Figure 3 is a typical result of the time difference between successive drops versus drop number. The flow rate in this case is 0.400 g/s. The unpredictable variation indicates a chaotic behavior. Despite the variation, the average time difference remains approximately constant, suggesting that the flow rate is indeed constant. Fig. 4 shows the histogram of the time difference. The average is 0.150 (s) and the standard deviation is 0.002 (s).

To see the correlation between current and the next time

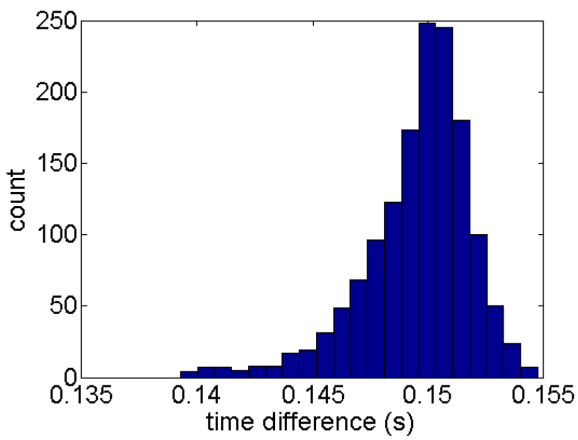

Figure 4. Histogram of the time difference data in Fig. 3.

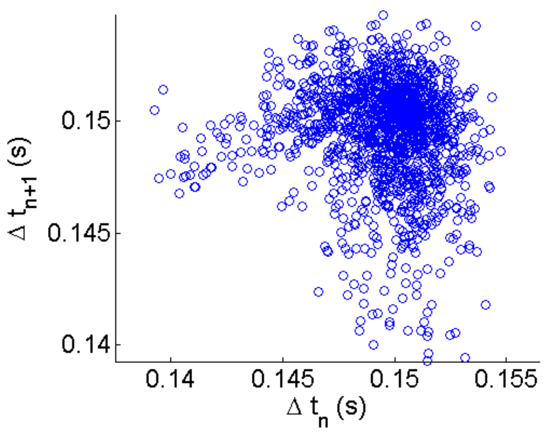

Figure 5. Scatter plot of the time difference data in Fig. 3.

difference, the scatter plot is shown in Fig. 5. The horizontal axis represents the time difference between the $n-1^{th}$and the n^{th} drops, and the vertical axis the $n^{th}$and the $n+1^{th}$ drops. The points are not concentrated on line $x = y$, suggesting that the drops are not formed periodically.

The chaotic time difference data is transformed into frequency domain via Fast Fourier Transform algorithm to compute the power spectral density, shown in Fig. 6. Both x and y axes are in logarithmic scale. A linear relation is observed. The least square fit (red line) indicates that the slope (β) is 1.59, with $r^2 = 0.55$. Typically, β for chaotic drop time difference ranges from 1.2 to 1.8. The data is my primary $1/f^{\beta}$source.

Converting to music

Figure 7 is a snippet of MATLAB code which reads the raw drop data. Line 4 eliminates extreme data; namely, points differ from the average by two standard deviations or more. Line 7 randomly selects a segment and returns a sequence of numbers of desired length (n). There are infinite number of ways to convert the drop signal to music. By trial

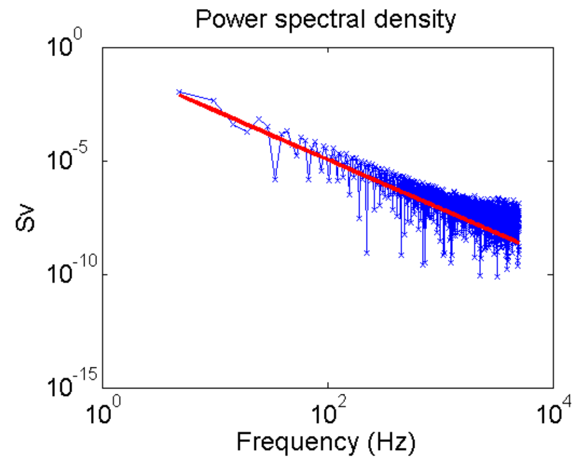

Figure 6. Power spectral density of the time difference data in Fig. 3.

```
1 function y = drop(n,dataname)
2 txt = importdata(dataname,' ',14);
3 timediff = txt.data(:,2);
4 timediff = timediff(find(abs(timediff-mean(
      timediff)) < 2*std(timediff)));
5 L = length(timediff);
6 ini = round(random('unif',1,L-n));
7 y = timediff(ini:ini+n-1)';
```

Figure 7. A snippet of MATLAB code which reads raw drop data.

and error, the most effective procedure I have found is presented below.

Brownian noise perturbation

A major drawback of the music generated directly from drop data (e.g. drop.mid) is that there is no trend or structure, despite a certain degree of correlation. To solve this problem, the drop data is perturbed by a Brownian noise. The Brownian noise is produced by a random walk simulation, and the displacement at each step follows a normal distribution (Fig. 9). For instance, Fig. 8 (top) shows the original drop time difference, and Fig. 8 (middle) shows the Brownian noise. The amplitudes of the two sequences have the same order of magnitude.

The Brownian noise is added to the time difference to get Fig. 8 (bottom). A clear trend can be seen; for example, a climax appears around the 30th drop. The sequence is partitioned evenly into 36 regions and mapped to 36 integers for further processing. In general, the number of regions can range from 12 to 48, depending on the personal preference.

Repetition

Most music compositions have motives, which are recurrent sequences of notes. Fig. 10 shows the code that models this effect. Lines 1 and 2 randomly selects a segment of 8 notes as the motif. Then a replacement of notes is made to repeat the motif. For example, if the first note of a 4-note motif is C, then every time C occurs in the entire sequence,

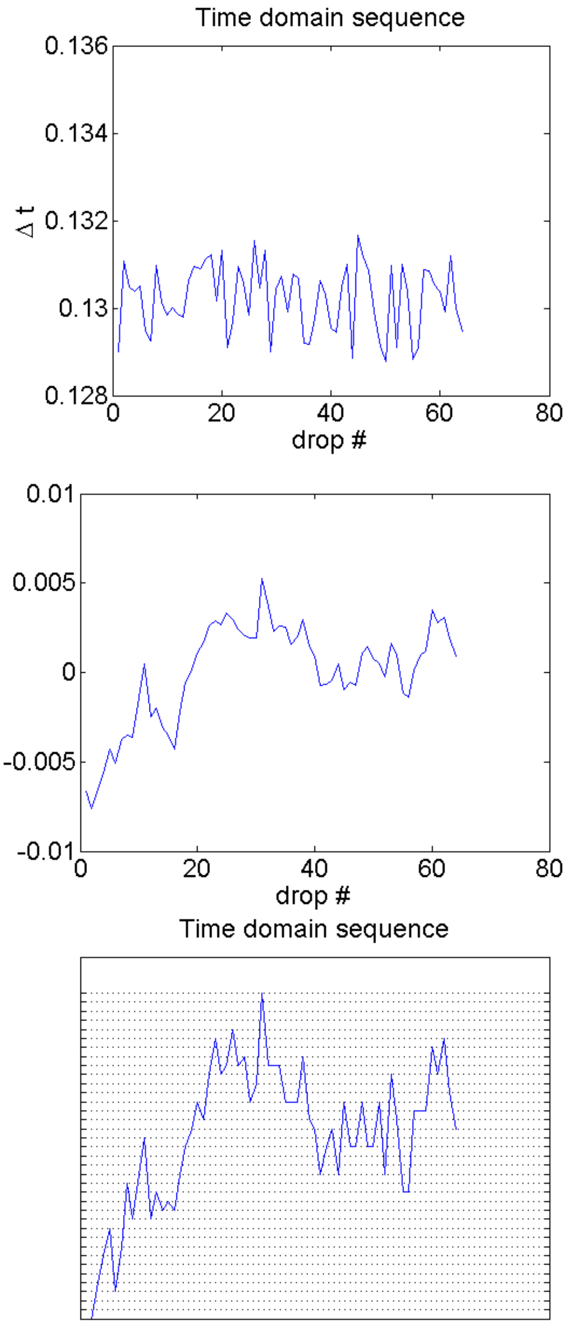

Figure 8. An example of drop time difference (top), the Brownian noise used for perturbation (middle), and the sum of two sequences partitioned into 36 regions (bottom).

```
function y = brown(n)
y = zeros(1,n);
for ii = 2:n
    y(ii) = y(ii-1) + random('norm',0,1);
end
```

Figure 9. A snippet of MATLAB code which generates a Brownian noise.

```
s = round(random('unif',round(n/4),round(n
    /2)));
motif = notes(s:s+7);
idx = findstr(notes,[motif(1)]);
for ii = 1:length(idx)
    notes = [notes(1:idx(ii)) motif(2:end)
        notes(idx(ii)+length(motif):end)];
end
```

Figure 10. A snippet of MATLAB code which adds repetition to the sequence.

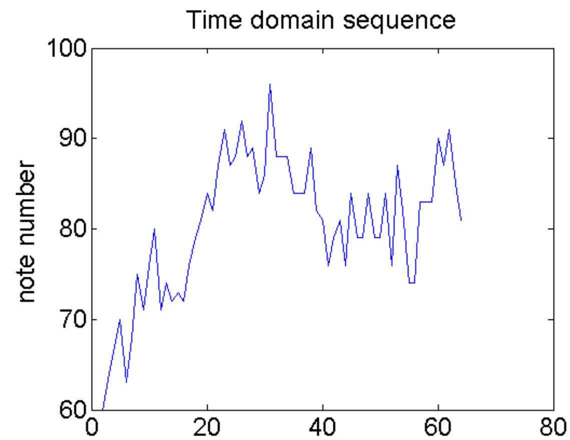

Figure 11. Sequence of notes produced by drop data. The horizontal axis is time and the vertical axis is the number of key on the piano.

the note and the following 3 notes will be replaced by the motif. The length of the entire sequence stays the same after the process.

There will be quite a few repetition after this step. However, some of the motives will be changed by the following processes and others will remain unchanged, which leads to a combination of predictability and randomness.

Chord progression

Chord progression can be found in many music compositions. In this step, the processed drop data is mapped into musical notes (Fig. 11). The notes are then adjusted to follow the chord progression (Fig. 12), so that the music sounds more structured. The following two chord progressions turn out to be effective in my music generator: I-IV-V-I and I-V-vi-iii-IV-I-ii-V.

To illustrate how the algorithm works, an example of I-IV-V-I in C major is shown in Fig. 13. The first bar is in I chord, which means that every note in this bar will be changed to either C, E, or G, whichever closest to the original pitch. The notes in the next bar will be changed to either F, A, or C, etc. To account for the freedom in music, each note will have a 30% chance to remain unchanged. Again, the percentage can be adjusted according personal preference.

```
function [y2,progression] = tochord(y,
    notesperbar)
triad = [0 4 7];
triad = [triad-12 triad triad+12 triad+24
    triad+36];
progression = [1 4 5 1]-1; % I IV V I
pp = 0; nn = 0;
y2 = y;
while nn < length(y) - notesperbar
    for nn = nn + [1:notesperbar]
        notes = progression(pp+1) + triad;
        if rand(1) > 0.3
            y2(nn) = notes(dsearchn(notes',y
                (nn)));
        else
            y2(nn) = y(nn);
        end
    end
    pp = mod(pp + 1,length(progression));
end
```

Figure 12. A snippet of code which adjusts notes to follow chord progression.

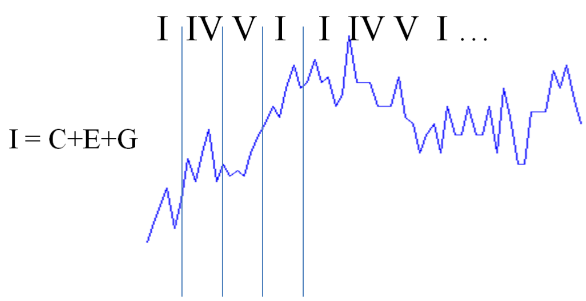

Figure 13. Illustration of chord progression adjustment.

Rhythm variation

So far, adjustments are made only on the frequency (melody). The rhythm is also changed in this step simply by removing notes according to the $1/f$ noise of drop time difference data. However, it is harder to control the rhythm fluctuation. The rhythm determined merely by the $1/f$ noise sounds awkward; therefore, only a little rhythm variation is added.

Harmony

To make the music sound "thicker," harmonies are formed for some notes by playing an octave-lower key at the same time. At the beginning of each bar, the root of the chord progression is also played at the same time. The loudness of the harmonics is adjusted such that there will not be a sudden increase in loudness when multiple keys are played at a time.

Finally, the sequence of notes is converted to a midi file with piano timbre, using the MATLAB script developed by Schutte (2012). A typical power spectral density of the processed sequence is shown in Fig. 6. The computer generated music is indeed a $1/f^{\beta}$ noise with β ranges from 0.8 to 1.5.

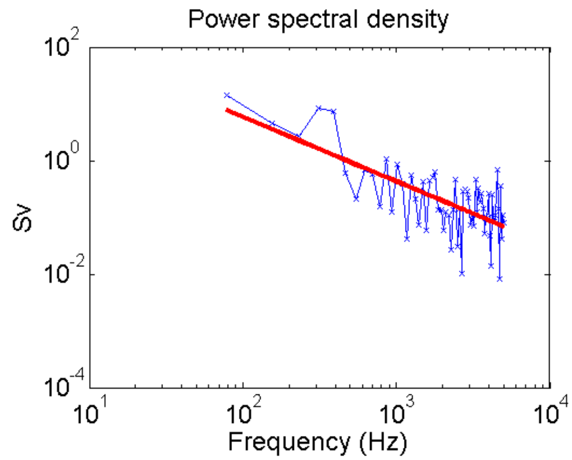

Figure 14. Power spectral density of a processed note sequence. The red line is the least square fit with slope 1.1 and $r^2 = 0.50$.

Discussion

Examples of my computer generated music are "piano1.mid," "piano2.mid" and "piano3.mid." A new music piece can be generated simply by pressing a key to run the code. Because of the random processes, the music generated each time is unique. Some parameters can be changed manually, such as the length of music or the probability that a note will be adjusted. The processed music is hopefully more enjoyable than the music generated directly from the raw drop time difference data. The fact that it is possible to create listenable music with a computer suggests that there are patterns in music people enjoy.

Creativity is a key element in music. After the program is written, it requires little creativity to produce music, even with more sophisticated algorithm. Computer generated music may provide motivations for composers, but they can never replace the role of human composed music.

Acknowledgment

I would like to thank Professor Steven Errede for his guidance in the project and for helping me fix the chaotic faucet setup. I would like to thank the TAs, Tom Satrom and Darby Hewitt, and other students for their help and for bearing with the noise I produced when looking for a chaotic flow rate.

References

Fregeau, J. (1998). Investigating the chaotic faucet. *Physics 303 Final Project*.

Hennig, H., Fleischmann, R., Fredebohm, A., Hagmayer, Y., Nagler, J., Witt, A., . . . Geisel, T. (2011). The nature and perception of fluctuations in human musical rhythms. *PLoS ONE*.

Janata, P., Birk, J. L., Van Horn, J. D., Leman, M., Tillmann, B., & Bharucha, J. J. (2002). The cortical topography of tonal structures underlying western music. *Science*, *298*(5601), 2167-2170. doi: 10.1126/science.1076262

Levitin, D. J., Chordia, P., & Menon, V. (2012). Musical rhythm spectra from bach to joplin obey a 1/f power law. *Proceedings of the National Academy of Sciences*. doi: 10.1073/pnas.1113828109

Martien, P., Pope, S., Scott, P., & Shaw, R. (1985). The chaotic behavior of the leaky faucet. *Physics Letters A*, *110*(7¡V8), 399 - 404. doi: 10.1016/0375-9601(85)90065-9

Schutte, K. (2012). Matlab and midi. Retrieved from `http://www.kenschutte.com/midi`

Tymoczko, D. (2006). The geometry of musical chords. *Science*, *313*(5783), 72-74. doi: 10.1126/science.1126287

Voss, C. (1978). "1/f noise" in music: Music from 1/f noise. *J. Acoust. Soc. Am.*, *63*(1), 258-263.