Chapter 22

Randomness in Music

[Unpublished notes based on an informal talk given on 9 May 1990.]

John Chowning once asked me to speak for a few minutes to the Stanford Music Affiliates, a group of people representing the companies who were helping to sponsor the research at CCRMA, Stanford's Center for Computer Research in Music and Acoustics. Here is a reconstruction of approximately what I said, based on the slides that I used at the time.

Patterns that are perfectly pure and mathematically exact have a strong aesthetic appeal, as advocated by Pythagoras and Plato and their innumerable intellectual descendants. Yet a bit of irregularity and unpredictability can make a pattern even more beautiful. I was reminded of this fact as I passed by two decorative walls while walking yesterday from my home to my office: One wall, newly built, tries to emulate the regular rectangular pattern of a grid, but it looks sterile and unattractive to my eyes; the other wall consists of natural stones that fit together only approximately yet form a harmonious unity that I find quite pleasing.

An ugly new wall

A delightful old wall

I noticed similar effects when I was experimenting years ago with the design of computer-generated typefaces for the printing of beautiful books.

A design somehow "came to life" when it was not constrained to be rigidly consistent.*

Similar examples abound in the musical realm as well as in the world of visual images. For example, I'm told that people who synthesize music electronically discovered long ago that rhythms are more exciting if they don't go exactly "1, 2, 3, 4" but rather miss the beat very slightly and become what a mathematician might call "$1+\delta_1$, $2+\delta_2$, $3+\delta_3$, $4+\delta_4$." Although the discrepancies δ mount to only a few milliseconds, positive or negative, they enliven the music significantly by comparison with the deadly and monotonous pulsing that you hear when the δ's are entirely zero.

Singers and saxophone players know better than to hit the notes of a melody with perfect pitch.

Furthermore, we can take liberties with the "ideal" notes themselves. In an essay called "Chance in artistic creation," published in 1894,† August Strindberg recounted the following experience:

> A musician whom I knew amused himself by tuning his piano haphazardly, without any rhyme or reason. Afterwards he played Beethoven's *Sonate Pathétique* by heart. It was an unbelievable delight to hear an old piece come back to life. How often had I previously heard this sonata, always the same way, never dreaming that it was capable of being developed further!

And the notion of planned imperfection is not limited to variations in the performance of a given composition; it extends also to the choices of notes that a composer writes down. The main purpose of my talk today is to describe a way by which you could build a simple machine that will produce *random harmonizations of any given melody.*

More precisely, I'll show you how to produce $2^n + 2^{n-1}$ different harmonizations of any n-note melody, all of which are pretty good. A machine can easily generate any one of them, chosen at random, when its user plays the melody on a keyboard with one hand.

The method I shall describe was taught to me in 1969 by David Kraehenbuehl (1923–1997), when I audited his class on Keyboard Harmony at Westminster Choir College. It is extremely simple, although

---

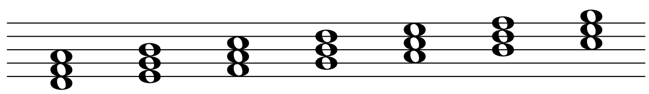
* See, for example, my book *Digital Typography*, pages 57–59, 286–287, 324–325, 386, 391–396; also *The METAFONTbook*, pages 183–185.

† Auguste Strindberg, "Du Hasard dans la production artistique," *La Revue des revues* **11** (15 November 1894), 265–270.

you do need to understand the most elementary aspects of music theory. I shall assume that you are familiar with ordinary music notation.

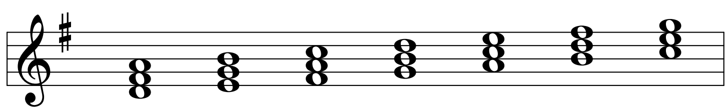
Kraehenbuehl's algorithm produces four-part harmony from a given melody, where the top three parts form "triadic" chords and the bottom part supplies the corresponding bass notes.

A *triad* is a chord that consists of three notes separated by one-note gaps in the scale. Thus the triads are



and others that differ only by being one or more octaves higher or lower. The bottom note of a triad is called its "root," and the other two notes are called its "third" and its "fifth."

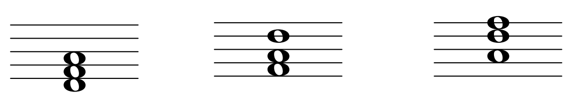
These notions apply to any clef and to any key signature. For example, with the treble clef and in the key of G major, the seven triads



are known by more precise names such as a "D major triad," etc.; but we don't need to concern ourselves with such technical details.

The important thing for our purposes is to consider what happens when individual notes of a triad move up or down by an octave. If we view these chords modulo octave jumps, we see that they make a different shape on the staff when the root tone is moved up an octave so that the third tone becomes lowest; this change gives us the *first inversion* of the triad. And if the root and third are both moved up an octave, leaving the fifth tone lowest, we obtain the *second inversion*:

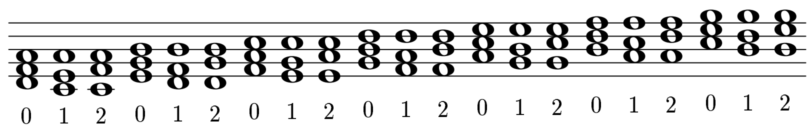
Root position First inversion Second inversion



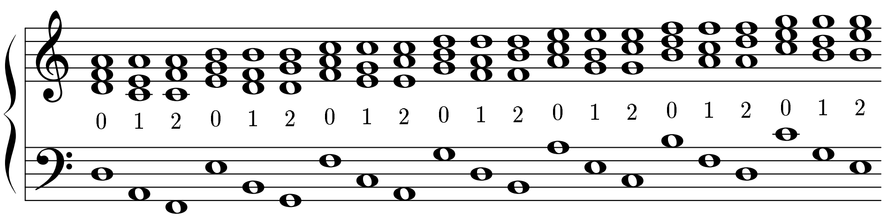
Even though two-note gaps appear between adjacent notes of the first and second inversions, these chords are still regarded as triads, because octave jumps don't change the name of a note on its scale: An A is still an A, etc., and no two notes of an inverted triad have adjacent names.

Music theorists have traditionally studied three-note chords by focussing their attention first on the root of each triad, and next on the bottom note, which identifies the inversion. Kraehenbuehl's innovation was to concentrate rather on the *top* note, because it's the melody note.

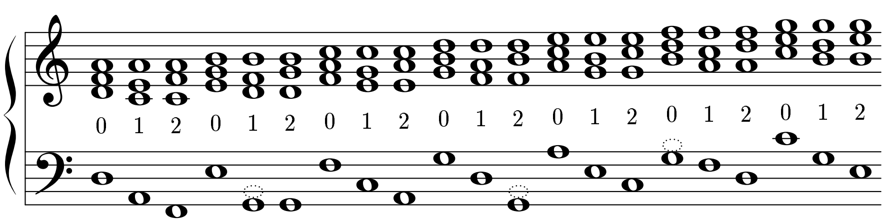
He observed that each melody note in the scale comes at the top of three triadic chords — one in root position (0), one in first inversion (1), and one in second inversion (2):



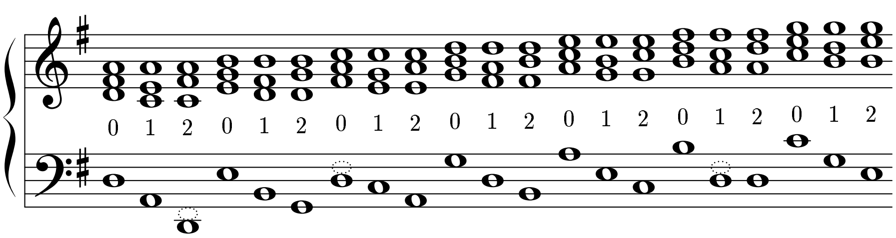
Furthermore, said Kraehenbuehl, there's a natural way to add a fourth part to this three-part harmony by simply repeating the root note, an octave or two lower. For example, in the key of C, we get



as the four-part harmonizations of melody notes A, A, A, B, B, B, ..., G, G, G. This rule almost always works well, but like all good rules it has an exception: When the bass note turns out to be the *leading tone* of the scale (which is one below the tonic), we should change it to the so-called *dominant tone* (which is two notes lower). Thus Kraehenbuehl's correction to the natural rule yields the 21 four-part chords



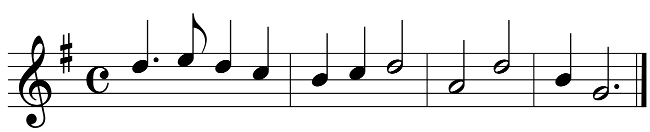
in the key of C, when B is the leading tone; and it yields



in the key of G, because F$^\sharp$ is the leading tone in that case. Notice that when the bass note is corrected by shifting it down from a leading tone in root position, it produces a chord with four separate pitches (the so-called "dominant seventh chord" of its key), so it's no longer a triad.

OK, now we know three good ways to harmonize any given melody note in any given key. Kraehenbuehl completed his method by pointing out that the same principles apply to melodies with any number of notes, *provided only that we never use the same inversion twice in a row.* If one note has been harmonized with, say, the first inversion, the next note should be harmonized with either the root position or the second inversion; and so on. With this procedure there are three choices for the first chord, and two choices for every chord thereafter.

Let's test his algorithm by trying it out on a familiar melody:

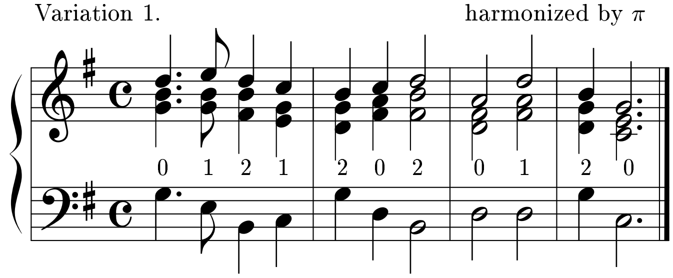


"London Bridge is falling down, my fair lady" has eleven notes, so Kraehenbuehl has given us $3\cdot2\cdot2\cdot2\cdot2\cdot2\cdot2\cdot2\cdot2\cdot2\cdot2=3072$ ways to harmonize it. The binary representations of three fundamental constants,

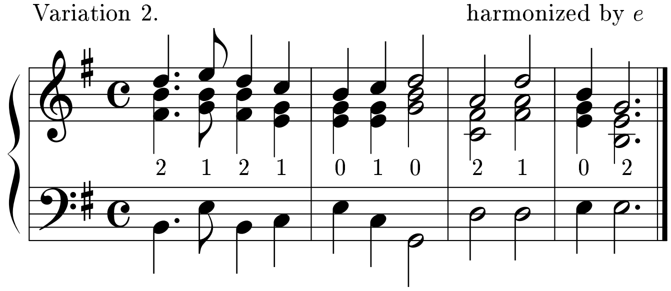
$$\pi = 3 + (.00100100001111110110101010001\ldots)_2,$$
$$e = 2 + (.10110111111000010101000101100\ldots)_2,$$
$$\phi = 1 + (.10011110001101110111100110111\ldots)_2,$$

serve to define three more-or-less random sequences of suitable inversions, if we prefer mathematical guidance to coin-flipping. Namely, we can use the integer part of the constant to specify the first chord, then we can change the number of inversions by $+1$ or -1 (modulo 3) for each successive binary digit that is 0 or 1, respectively. This procedure gives us three new harmonizations of that classic British theme:

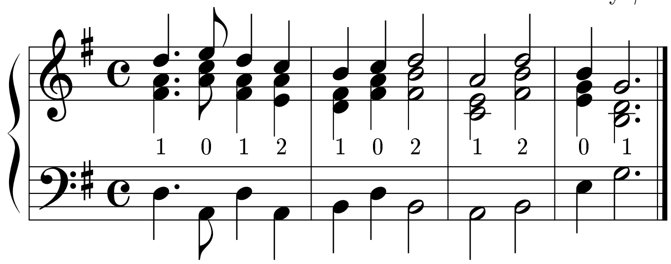
Variation 1. harmonized by π



Variation 2. harmonized by e

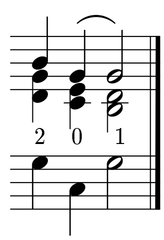


Variation 3. harmonized by ϕ

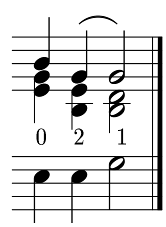


Amazing. Kraehenbuehl's algorithm seems far too simple to be correct, yet it really works!

Of course there's a little glitch at the end, because we have only one chance in three of ending on a chord that's stable and "resolved." No problem: In such a case we can just repeat the last melody note and force the desired chord. With this extension, variations 1 and 2 will end nicely with



and



.

I can hardly wait for somebody to build me a keyboard that will perform such harmonizations automatically. After all, it's basically just a small matter of programming.