

# Piano Tuning Method

Zuheng Kang

## CONTENTS

Abstract .....	2
Project Location.....	2
1    Introduction .....	2
2    Technical Knowledge.....	3
2.1    Key Names .....	3
2.2    Key Numbers.....	3
2.3    Functions .....	3
2.4    Tuning Methodology .....	4
3    Piano Tuning Method .....	4
3.1    Traditional Method.....	4
3.1.1    Sampling Piano.....	4
3.1.2    Audio Processing.....	4
3.1.3    Frequency Analysis .....	4
3.1.4    Catchup Overtone.....	5
3.1.5    Inharmonicity Model .....	6
3.1.6    Tuning Curve Optimization Model .....	8
3.1.7    Temperament Model.....	11
3.1.8    Creating Tuning Strategy Table .....	12
3.2    Entropy Tuning Method .....	12
3.2.1    Sampling Piano & Audio Processing .....	13
3.2.2    Construct Spectrum .....	13
3.2.3    Tuning with Entropy Optimizer .....	13
3.2.4    Creating Tuning Strategy Table .....	15
3.2.5    Tune for Songs .....	16
4    Audio Processing & Pure Sound Tuner.....	17
4.1    Tuning .....	17
4.2    Sound Purify.....	17
5    Future Work .....	19
6    Reference.....	19
7    Appendix .....	20

## ABSTRACT

*Since the piano string is considered to be a stick rather than a pure ideal string, it contains stiffness and its overtone will shift in such way that makes piano tuning a difficult work. In this work, two optimization algorithm for the piano tuning method is presented. The traditional tuning algorithm is divided into several models that using various fitting technique model the target piano, and then convert to linear regression problem for optimization. The entropy tuning method is a trial method to tune the piano to minimize the entropy value when all keys are pressed – to achieve a simpler spectrum in pitch domain. In addition, a pure tuner method is invented to get rid of all inharmonic effect of piano sound.*

**Keyword:** *piano tuning, inharmonicity, entropy, audio processing*

## PROJECT LOCATION

*Reference [2]*

---

## 1 INTRODUCTION

Piano tuning is a difficult work since the frequency peaks shift that makes the piano hard to tune. The tuning process will be a task to highly reduce the audible cacophonous. There are several factors we need to consider, which the rule of harmony is.

- The cacophonous created by its base frequency and audible harmonics; a good tuning will largely reduce the inharmonic effects for harmonies (the frequency domain should be simple, which the frequency peaks should merge or coincide).
- The inner music scales related pitch; the odd pitch tuning will result in the weird effect when playing music scales.

Other famous related works are:

- Tunelab (closed source; has a trial version)
- Reyburn CyberTuner (closed source; no trial version)
- Entropy Piano Tuner (open source) [1]

The first two are similar, which represent the old tuning techniques, and my work mostly focuses on this algorithm.

As for Entropy Piano Tuner, it represents the new way of piano tuning. It can also achieve a very good result for tuning a piano, however, this temperament is not a regular 12-equal temperament, but a piano approximation temperament starting from 12-equal temperament, in order to largely eliminate the non-harmonious effect.

- Since the pitch in the piano does not have relatively same pitch interval, some inner scales sound weird.
- Since the piano optimizes all 88 keys harmony, it values overall harmonious – some simpler chord might not sound harmonious.
- It only considers the sound which at the certain striking level of piano keys, which result in the optimization of keys are based only on the given key pressing level. However, it values the average case for piano performance, thus it covers the majority situation of harmony cases.
- The accuracy cannot be too high due to a large amount of calculation, it does not achieve an ideal result.

In my work, I will talk about two piano tuning methods and one audio processing method.

- As for traditional tuning method, since it is closed source, I guessed their tuning method and create a similar solution, and will be shown in this article. Besides, I used more accurate model for inharmonicity coefficients.
- I will reproduce the result for the Entropy Piano Tuning method.

- The tuning for audio and a pure sound tuner is introduced.

In this article, the first part is to introduce the technical knowledge of high-level modeling algorithms. The second part is to introduce my piano modeling and tuning optimization method. Then, followed an audio processing technique. Finally, the future work will be introduced.

## 2 TECHNICAL KNOWLEDGE

### 2.1 KEY NAMES

The leftmost key name is defined as “A0”, where “A” is the note name, 0 is the scale number. “C” is the starting point of one scale. It only allowed sharp in the note, flat is not allowed in this naming format.

*A0, A#0, B0, C1, C#1, ..., B1, C2, ..., B7, C8*

There are 88 keys for standard piano.

### 2.2 KEY NUMBERS

In the real world, the piano key will be labeled with numbers when the piano is open and machine part is shown off.

A0 key is labeled to be 1, and “C8” is 88.

However, in my program, “A0” key is labeled as 0 for easier calculation, which is defined as  $k$ .

### 2.3 FUNCTIONS

Frequency ratio to cents function:

$$\text{Fr}_{\rightarrow c}(\gamma) = 1200 \log_2(\gamma) \quad (2.1)$$

The inverse process is:

$$\text{C}_{\rightarrow \text{fr}}(c) = 2^{\left(\frac{c}{1200}\right)} \quad (2.2)$$

Where cents is from 12-equal-temperament, each half note has 100 divisions, named cents.

Frequency add cents (pitch) function:

$$\text{F}_{+c}(f, c) = f \cdot 2^{\left(\frac{c}{1200}\right)} \quad (2.3)$$

This function returns the frequency that added the pitch (cents)  $c$ .

The ideal frequency for the key  $k$  is:

$$\tilde{f}_k = \tilde{f}_{[A4]} \cdot 2^{\left(\frac{k-48}{12}\right)} \quad (2.4)$$

Where  $\tilde{f}_{[A4]}$  is the international standard pitch for “A4”, usually defined as 440Hz. Another tuning standard will replace this number, 48 is the key number for “A4”.

## 2.4 TUNING METHODOLOGY

Since the minor tuning for each string will rarely affect its stiffness, from Equation (3.3), we assume that the  $B_k$  is the constant.

## 3 PIANO TUNING METHOD

### 3.1 TRADITIONAL METHOD

The traditional tuning method is to match the specific frequency peaks that aimed at largely eliminating the “beat” (pitch differences from two notes; for example, “A3’s” second overtone matches its octave “A4”, which is denoted to be 2:1). Then, use a smooth curve to optimize/minimize all the differences to achieve a relatively good result.

Since the piano sound overtone shift (inharmonicicity) has a very nice relation, it enables us to just sample very few keys and guess all the properties for all piano; then, get the tuning strategy.

#### 3.1.1 Sampling Piano

Before tuning a piano, we need to sample a piano by recording few piano keys sound audios. This process will roughly or precisely measure the inharmonicity of piano strings (which will talk about later), such that we could model the inharmonicity for the targeted piano.

The sampling is suggested to measure keys “C1”, “C2”, “C3”, “C4”, “C5” (and probably “C6”; the user could record more piano keys such as “A1” ~ “A6” for better result). Since the tuning inharmonicity curve is a smooth curve and predictable, thus it is possible to sample fewer notes. The piano key sound should be recorded in a quiet environment, which allows more accuracy for later frequency analysis. In this sampling process, we need to press the key hard in order to get higher harmonic peaks for measurement.

In my program, I use fully or almost fully sampled piano for research purposes.

#### 3.1.2 Audio Processing

Since the real audio may contain the white space at the start or the end, and the sound length varies. I use this method to process my sampled audio:

- Normalize ( $N(x) = x / \max(x)$ ) the audio file into 1, then, find the peak volume of audio, and start from here.
- Slice these audio pieces into tiny partitions, say 0.1 second is one partition. The maximum number of each partition will be its assumed volume at this time point.
- Trim the audio at the volume starting from some large number to a small number – since the piano sound is loud from its beginning and decay by the time. Say from 90% to 2% of the sampled sound audio.

#### 3.1.3 Frequency Analysis

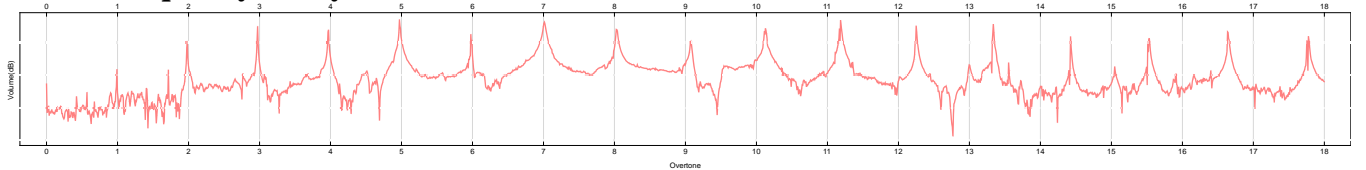


Figure 3-1 “A#0” Key (at Upright Piano Samples) Overtone Plot; Volume at Logarithm Scale

Then, put this audio sample into Fourier analysis (FFT algorithm). Then we get the function  $G_k(f) = \|\text{FFT}(S_k(t))\|_2$  where  $S_k(t)$  is the audio function, and  $G_k(f)$  is the frequency domain function,  $k$  is

piano key number,  $f$  is the frequency variable,  $\|\cdot\|_2$  is the 2-norm of complex numbers. In our work, the frequency domain is converted to the ratio to its ideal fundamental frequency, thus we can see the Figure 3-1, the peaks will always almost lies in the grid by dividing its ideal frequency.

From Figure 3-1, we can see that the higher overtone (right-hand side peaks with larger numbers) shifts higher.

It is a problem to capture all these peaks numbers since some are not clear: the fundamental frequency (at 1), and some have multiple peaks: at 15 ~ 16.

In my work, I use the frequency *Catchup Method* to get octave values for all these peaks.

### 3.1.4 Catchup Overtone

From the characters of these peaks, there are several characters will be considered:

- From left to right, the gap between two peaks is increasing gradually.
- The largest value of this plot is probably some peak of overtone
- The valid peak should be nearly larger than fundamental frequency position: at 1.
- The peak may be broken into several peaks, we need to centralize the targeted position.

From this characteristic, the *Catchup Method* could be built:

- Analyze the frequency samples which roughly larger than 1 (my program is starting from 0.8), get the peak frequency  $f_{k,peak}$  at the key number  $k$  and overtone number  $peak$ .
- Comparing with ideal frequency  $\tilde{f}_k$ . We can then assume that it is  $n = \text{round}(f_{k,peak} / \tilde{f}_k)$  harmonics.

Then, we can know its guessed fundamental frequency is  $\hat{f}_k = f_{k,peak} / n$ . Then, this should be the step size for catchup method.

- The catchup method is forward (going to the right), and the backward (goes to the left). If we are in the forward operation, the next guessed target frequency is  $\hat{f}_{k,peak+1} = f_{k,peak} + f'_k$ , where  $f'_k$  is the assumed gap between two peaks at this position. On the first try, we set this number to  $f'_k = \hat{f}_k$ , and this number will be increasing for more right harmonics. Then, we get data around it (in a relatively small area) for guessed target frequency  $\hat{f}_{k,peak+1} \pm \delta$ . We can find its maximum number these data to be the frequency candidate  $\hat{f}_{k,peak+1}^{candidate}$ , then we get the data of smaller surrounding area  $\hat{f}_{k,peak+1}^{candidate} \pm \delta'$  where  $\delta' \ll \delta$ . Then, we calculate the weighted average for this smaller area, and the result is the actual frequency of this peak  $f_{k,peak+1} = \int_{\hat{f}-\delta'}^{\hat{f}+\delta'} \omega \cdot G(\omega) d\omega$ , where  $\omega$  is proportional to frequency. Then, the assumed gap between two peaks at this step is updated to be  $f'_k = f_{k,peak+1} - f_{k,peak}$ .
- Iterate this method for “forward catchup” to get all higher frequencies.
- If the highest peak is not fundamental frequency, we will perform the backward catchup. Since there are fewer peaks and the overtone shift will be far less than the right, the assumed targeted gap between two peaks is set to be the assumed fundamental frequency  $\hat{f}_k$ .

From this method, we can get an overtone (frequency) list for the key  $k$ . Which is:

$$k \rightarrow \{f_{k,1}, f_{k,2}, \dots\} \quad (3.1)$$

### 3.1.5 Inharmonicity Model

From Figure 3-1, we can see that the overtone will shift higher and higher as the frequency goes higher. This effect is caused by the stiffness of an object, its natural frequency will follow a certain pattern.

From reference [1], we assume that the piano string is a bar with two fixed ends, which approximately follows the partial differential equation:

$$\ddot{y} \propto -y'' - \varepsilon y'''' \quad (3.2)$$

Where  $y$  is the special position of piano string (bar model). The prime is the derivative to the spatial domain, and dots are the derivative to the time domain.

Then, use the modal analysis and solved the natural frequencies of this string are:

$$f_{k,n} \propto n \cdot f_{k,1} \sqrt{1 + B_k \cdot n^2} \Rightarrow f_{k,n} = A_k \cdot n \cdot f_{k,1} \sqrt{1 + B_k \cdot n^2} \quad (3.3)$$

Here we have two unknown variables  $A_k$  and  $B_k$ .

Then, we use this function to fit all frequency results at Equation (3.1). The parameter  $A_k$  is set for not all fundamental frequency is guessing perfectly. We can ignore this number by making sure the fundamental frequency always targets at 1, and focus only on  $B_k$ .

Then, we can get inharmonicity parameter list  $\{\{k, B_k\}\}$ .

From my observation, the logarithm of this number has some beautiful properties with the data  $\{\{k, \ln(s \cdot B_k)\}\}$ , where  $s$  is a scaling parameter (I set to 10000).

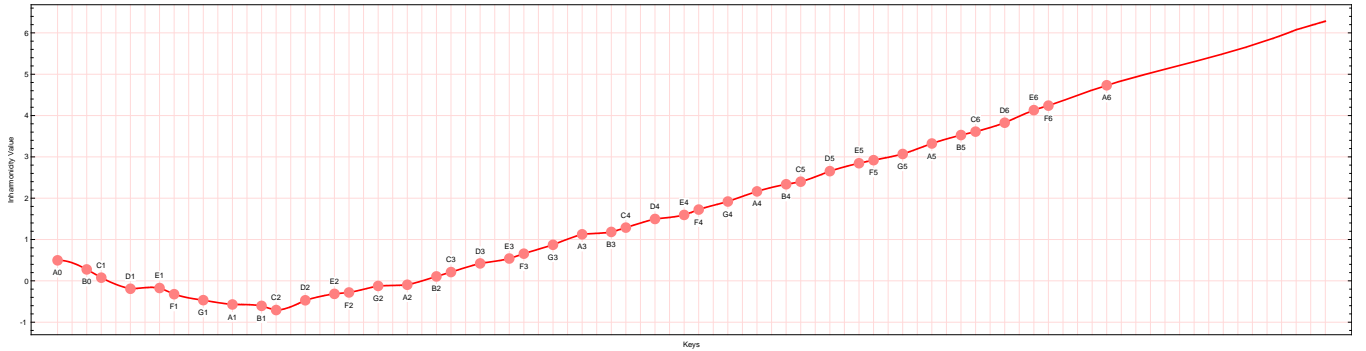
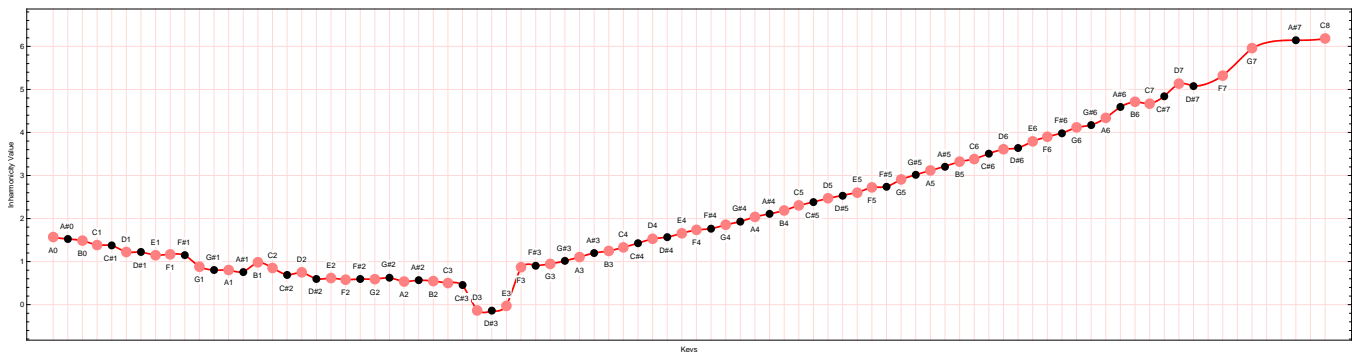


Figure 3-2 Inharmonicity Plot of Grand Piano IH(k)



**Figure 3-3 Inharmonicity Plot of Upright Piano  $IH(k)$**

From Figure 3-2 and Figure 3-3, we can clearly see the line is divided into 2 parts.



**Figure 3-4 Grand Piano String Arrangement**



**Figure 3-5 Upright Piano String Arrangement**

From Figure 3-4 and Figure 3-5, we can clearly see that the string is divided into two parts, with the steel string and copper string (may be covered by silver for highly expensive pianos). The upright piano has more copper strings since the steel string cannot go longer, and the string will become thicker to make the string vibrate slower. From spring vibration formula:

$$\omega = \sqrt{\frac{K}{m}} \quad (3.4)$$

Where  $\omega$  is proportional to frequency,  $m$  is the mass of spring,  $K$  is the stiffness of the spring.

When  $m$  increases,  $K$  increase a little bit,  $\omega$  decreases, then frequency decrease.

Since the piano cannot grow longer, it becomes thick and more like a stick rather than an ideal string. For higher notes strings, it is too short, and the thickness becomes relatively larger compared to its length, thus it is more likely to be a bar.

Thus, from the plot, we can see the inharmonicity increases at two ends, and break at the position of separation of two kinds of strings.

Since the grand concert piano is longer and can have more steel strings, fewer copper strings, thus the break will become a more left side.

The figure of inharmonicity plot also tells us that two separate lines are almost linear. In my model, I used the valid sampled points are modeled with an interpolation function, and the two edges are modeled with a linear function, and its method is shown below.

- We get several samples from one line and fit in a linear form.
- Get its slope, and build a line which passes the right-endpoint (since I will not wish to have a break for the interpolation function), and add some samples for edges situation to sample pool.
- Similar to the left-hand side.
- We use interpolation for these samples of sample pool – “left-hand side + samples + right-hand side”, which is our final model for inharmonicity model function  $\text{IH}(k)$ .

$$\text{IH}(k) = \ln(s \cdot B_k) \quad (3.5)$$

Thus, we can have the modeled parameter  $B_k$  with:

$$B_k = \frac{e^{\text{IH}(k)}}{s} \quad (3.6)$$

Then, the frequencies  $\tau(k, n)$  will be:

$$\tau(k, n) = f_{k,1} \cdot n \cdot \sqrt{\frac{1 + B_k \cdot n^2}{1 + B_k}} \quad (3.7)$$

Where  $f_{k,1}$  is currently unknown but it will be eliminated since it is in frequency ratio form. In this equation, we divide a term  $\sqrt{1 + B_k}$  to make sure the fundamental frequency is  $f_{k,1}$ .

### 3.1.6 Tuning Curve Optimization Model

Similar to Tunelab®, I set the tuning optimization method to separate the lower tones (bass) and higher tones (tenor) into two tuning target optimization method, the separation point  $k_0$  is “C#4/D4”. And the default tuning method for bass is to set 6:3. Since  $6/3=2$  ( $a/b$ ), this frequency ratio is  $\gamma = a/b$ , and its corresponding pitch range is  $\text{Fr}_{\rightarrow c}(\gamma)$  which is 1200, and 1200 is an octave, it means the tone say “A0”’s 6<sup>th</sup> harmonics will largely match its octave’s “A1”’s 3<sup>rd</sup> harmonics.

Here pitch is defined by cents.



The error function  $\varepsilon_k$  is defined as:

$$\begin{aligned}
\varepsilon_k &= \text{Fr}_{\rightarrow c} \left( \frac{\tau(k, a)}{\tau(k + \text{Fr}_{\rightarrow c}(a/b), b)} \right) \\
&= \text{Fr}_{\rightarrow c} \left( \sqrt{\frac{(1 + B_k \cdot a^2) \cdot (1 + B_{k + \text{Fr}_{\rightarrow c}(a/b)})}{(1 + B_{k + \text{Fr}_{\rightarrow c}(a/b)} \cdot b^2) \cdot (1 + B_k)}} \cdot \frac{a}{b} \cdot \left( \frac{f_{k,1}}{f_{k + \text{Fr}_{\rightarrow c}(a/b),1}} \right) \right) \\
&= \text{Fr}_{\rightarrow c} \left( \sqrt{\frac{(1 + B_k \cdot a^2) \cdot (1 + B_{k + \text{Fr}_{\rightarrow c}(a/b)})}{(1 + B_{k + \text{Fr}_{\rightarrow c}(a/b)} \cdot b^2) \cdot (1 + B_k)}} \right)
\end{aligned} \tag{3.8}$$

We can do this for all bass strings.

For tenor strings, the default tuning method is set to 4:1 ( $c/d$ ). But this time we count the higher note as the target to calculate.

$$\varepsilon_k = \text{Fr}_{\rightarrow c} \left( \sqrt{\frac{(1 + B_{k - \text{Fr}_{\rightarrow c}(c/d)} \cdot c^2) \cdot (1 + B_k)}{(1 + B_k \cdot d^2) \cdot (1 + B_{k - \text{Fr}_{\rightarrow c}(c/d)})}} \right) \tag{3.9}$$

The combined expression is:

$$E(k) = \begin{cases} \text{Fr}_{\rightarrow c} \left( \sqrt{\frac{(1 + B_k \cdot a^2) \cdot (1 + B_{k + \text{Fr}_{\rightarrow c}(a/b)})}{(1 + B_{k + \text{Fr}_{\rightarrow c}(a/b)} \cdot b^2) \cdot (1 + B_k)}} \right) & k \leq k_0 \\ \text{Fr}_{\rightarrow c} \left( \sqrt{\frac{(1 + B_{k - \text{Fr}_{\rightarrow c}(c/d)} \cdot c^2) \cdot (1 + B_k)}{(1 + B_k \cdot d^2) \cdot (1 + B_{k - \text{Fr}_{\rightarrow c}(c/d)})}} \right) & k > k_0 \end{cases} \tag{3.10}$$

From this equation, we can see  $E(k)$  is only a value for calculation at given  $k$ .

From this point, we need a function to largely eliminate these errors. The piano tuning curve  $C(k)$  is introduced, it represents the deviation of the actual tuning pitch to the ideal 12-equal temperament pitch.

The optimizer deviation function  $D(k)$  is:

$$D(k) = C(k) - E(k) \tag{3.11}$$

The cost function  $J(k)$  for optimization is:

$$J(k) = \sum_k (D(k))^2 \tag{3.12}$$

Which minimize the square error of these functions.

Here I use polynomial for easier calculation:

$$C(x) = \sum_{i=1}^n \chi_i \cdot x^i \quad (3.13)$$

Since  $C(x)$  will pass the fixed point, which is “A4” pitch at a 440Hz frequency at pitch deviation of 0, thus  $i$  is from 1 and  $x = k - k_{[A4]}$ , where  $k_{[A4]}$  is the key number (index) at “A4”, which is 48.

Thus,  $J(k)$  is the second order multi-variable polynomial function, which is very easy to minimize by linear regression method to calculate the fitting parameter  $\{\chi_i\}$ , and rebuild the functions.

Then, we can bring  $\{\chi_i\}$  to the  $D(k)$  function to calculate its deviations.

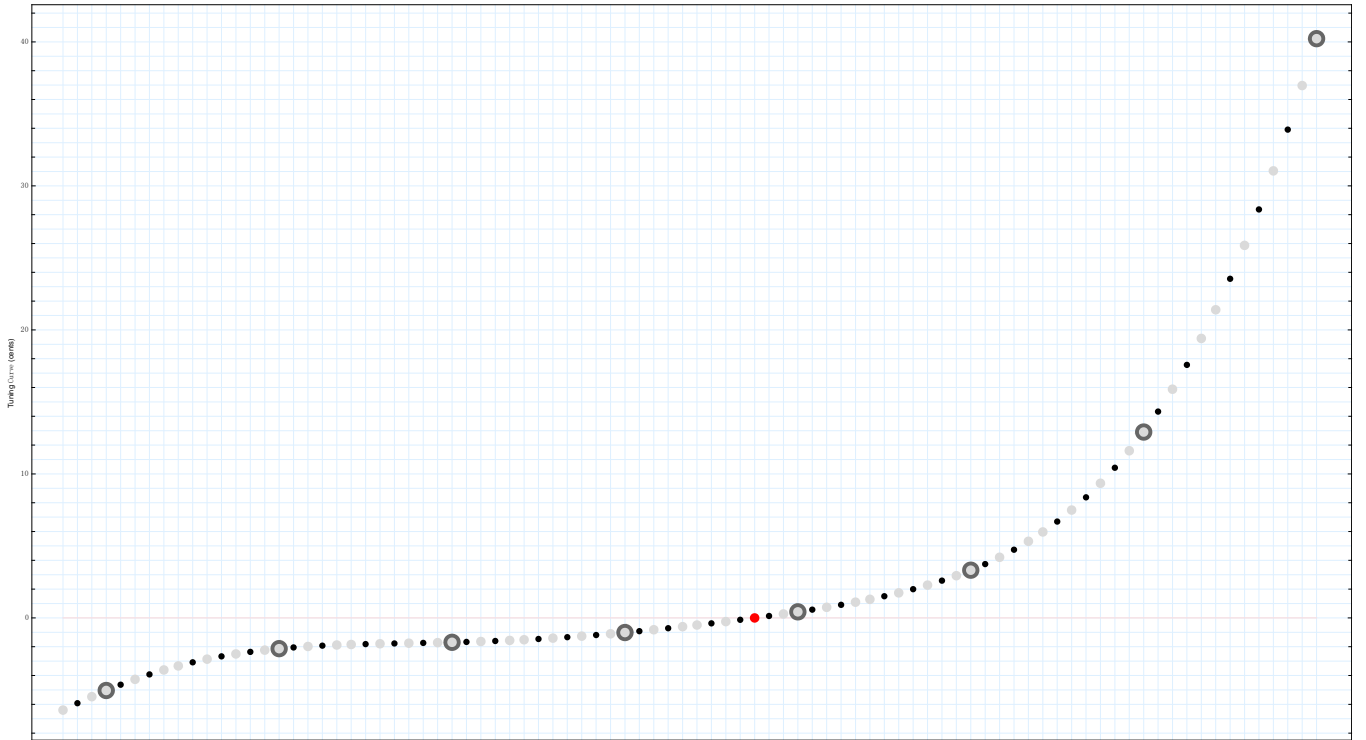


Figure 3-6  $C(k)$  for Grand Piano

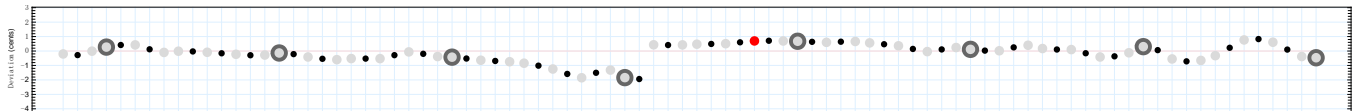


Figure 3-7  $D(k)$  for Grand Piano

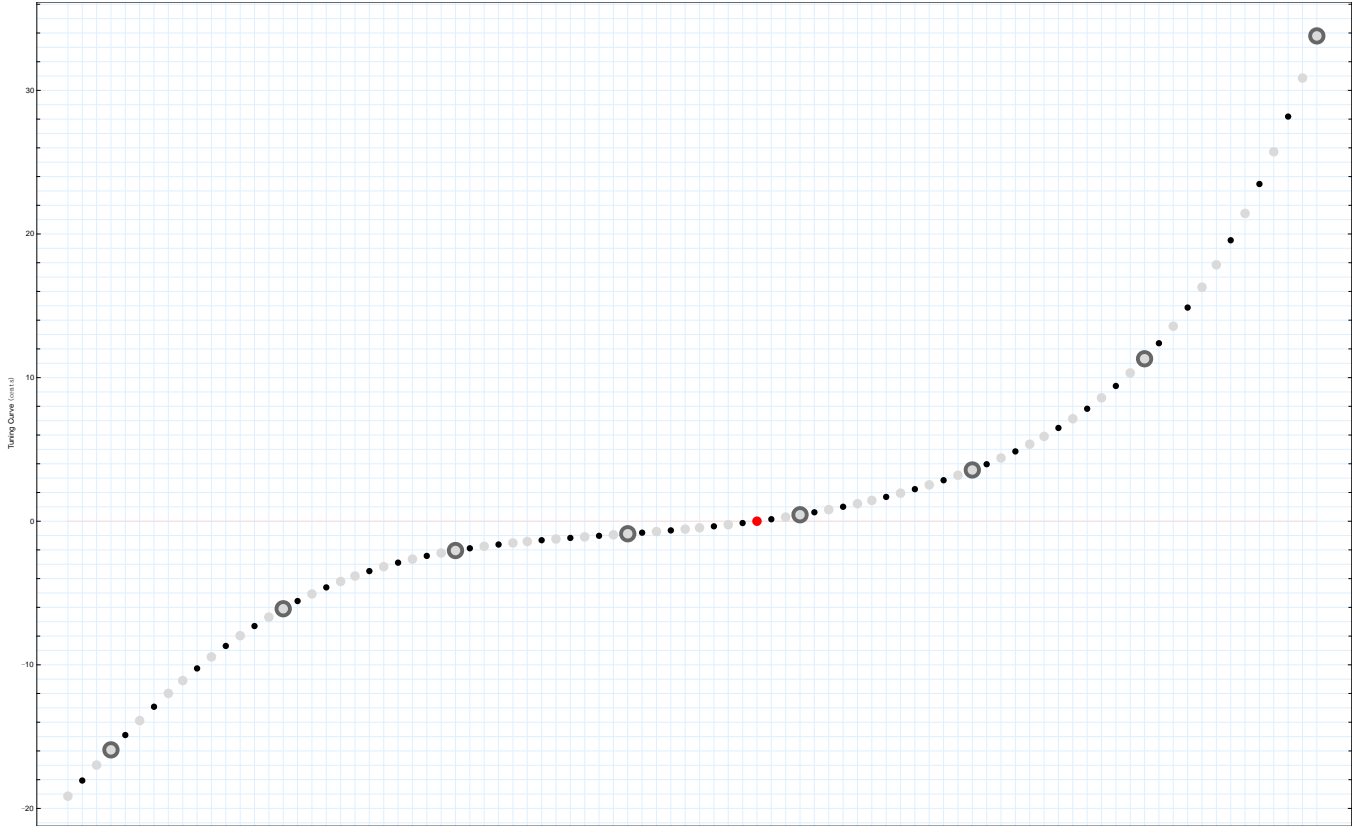


Figure 3-8  $C(k)$  for Upright Piano

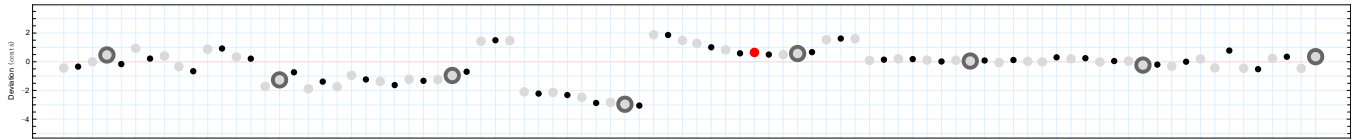


Figure 3-9  $D(k)$  for Upright Piano

The result of two pianos is shown above. The horizontal axis is the key number and the vertical axis of the pitch interval with its ideal frequencies represented by cents.

From this tuning method, we can see that the bass tuning will consider the deviations from the tenor part, and vice versa. The effect is inner related. Thus, this tuning method is theoretically to optimize almost the whole piano keys tuning.

### 3.1.7 Temperament Model

With the development of music, various temperaments appear and create the unique flavor of music. The temperament model is using the pitch deviation tables of different temperament (the unit is cent). We can then create the non-12 equal temperament tuning strategy. The temperament function is defined to be  $T(k)$ .

The tuning table such as “Bach - Bradley Lehman” is:

C	C#	D	D#	E	F	F#	G	G#	A	A#	B
5.87	3.91	1.96	3.91	-1.96	7.82	1.96	3.91	3.81	A 0	3.91	0

Table 3-1 Table for “Bach - Bradley Lehman” Temperament

Where A note will always be 0 since A is the reference frequency and will always keep to 440 Hz (if is standard situation).

This table shows the situation of “C” major.

The other major tuning will follow the rotation of the table. For example: if tuning “D” major, the “D” will rotate to current “D”  $\rightarrow$  “C” place, which is rotating left 2 times. However, we will make sure “A” note will always be 0, then, we can subtract the number at “B”  $\rightarrow$  “A” to make it possible.

Then, add these pitch errors to all the notes of tuning, the modified tuning curve is:

$$C'(k) = C(k) + T(k) \quad (3.14)$$

### 3.1.8 Creating Tuning Strategy Table

The final tuning strategy  $\tau(k, n)$  (unit: Hz) is:

$$f_{k,1} = F_{+c}(\tilde{f}_k, C'(k)) \quad (3.15)$$

$$\begin{aligned} \tau(k, n) &= f_{k,1} \cdot n \cdot \sqrt{\frac{1 + B_k \cdot n^2}{1 + B_k}} f \\ &= F_{+c}(\tilde{f}_k, C'(k)) \cdot n \cdot \sqrt{\frac{1 + B_k \cdot n^2}{1 + B_k}} \\ &= F_{+c}(\tilde{f}_k, C'(k)) \cdot n \cdot \sqrt{\frac{s + e^{IH(k)} \cdot n^2}{s + e^{IH(k)}}} \end{aligned} \quad (3.16)$$

From Equation (3.16), we can see only  $C(\cdot)$  and  $IH(\cdot)$  function is modeled function, other functions are basic mathematics functions.

From the modeling, we can get a strategy of piano tuning, then we can convert this strategy into a tuning table, which shows all the frequency of fundamental and its overtone frequencies, and corresponding deviation to ideal frequencies represented by cents.

The grand and upright piano tuning strategy is shown in Figure 7-1 and Figure 7-2.

The red font is the frequencies recommended for the devices to tune.

## 3.2 ENTROPY TUNING METHOD

The entropy tuning method is not to model the exact value of frequencies or pitches, it simulates the condition that simultaneously presses down all piano keys, and uses entropy method as the cost function to largely merge the peaks at pitch domain to create sharper and simpler sound for piano, which optimizes the piano sound. The method is extremely simple, however, it is really computational intensive.

Why simulate pressing all keys? We need to know the philosophy of piano in behind. To deal with all kinds of complicated situations, let us assume several cases. Whether the chord is harmonious is to check the transient pitch domain. In the other word, several notes at certain short time period will contact with each other, and we need to make sure this sound is harmonious. However, the contact cases of notes at all time for all songs are too complicated, and the key pressing level varies all the time. What if assuming that all notes have equal probability to contact, and the key pressing level when playing each small piece of music on average is the same – some pieces are loud, some

are small but they usually approximately on the same level when playing the piano. As for the key pressing level that could change the sound quality, we suggest the sample sound will be played in medium level.

### 3.2.1 Sampling Piano & Audio Processing

In the entropy piano tuning method, sampling every piano key is necessary. Another requirement is similar to a traditional method. The audio processing is also similar to a traditional method.

### 3.2.2 Construct Spectrum

Since the human ear is sensitive to the pitch (“pitch” is equivalent to the logarithm of a frequency component for approximation: ignore the nonlinear effect of ear structures) within the hearing range (20Hz ~ 10000Hz is reasonable for optimizing algorithm). Thus, the model should be built by putting equal significance to the pitch scale. Traditionally, the pitch is represented as music note. If we evaluate the “pitch” content/data by equally sampling from the pitch scale of the spectrum, it puts the equal importance to the pitch scale – the logarithm scale of frequencies. In my experiment, I put 0.1 cents as the precision.

Then, we have the converted the spectrum into pitch domain  $I(\kappa)$ , to resample the data with the key number:

$$I(\kappa) = \left\| G(f_\kappa) \right\|^\beta \Big|_{\kappa \rightarrow 12 \cdot \log_2 \left( \frac{f_\kappa}{f_{[A0]}} \right)}, \beta \rightarrow 2 \quad (3.17)$$

Where for each key  $k$  we will have 1000 samples in total, each sample’s pitch denote as  $\kappa$ . Namely, each sample will represent 0.1 cents. Since the audio is also the limited samples, I use the interpolation function to resample the data.

In this model, I use the square of the spectrum  $\beta = 2$ . The reason is that: although human ear sensitive to the sound pressure level is based on the logarithm of magnitude of sound, unit could be decibel (dB), however, the human ear also has the auditory mask, which masks small peaks around it, thus we should value more on major peaks, and ignore minor one. From the paper [1], and my trial and error, the power of 2 is actually achieved a very ideal result. I also tried other numbers for  $\beta$ , when  $\beta = 1$ , the sound is messy at all;  $\beta = 2$  is perfect;  $\beta$  is larger, the simpler sound will hear more harmonious, however, the complicated chord may not hear well since the algorithm may value more on merging major peaks of the spectrum and ignore the little ones. If people need to play more simple chord songs, they may try larger numbers of  $\beta$ , if need to play more messy types of songs like Impressionist or Jazz, I suggest they will use smaller  $\beta$ . On average, 2 is a great number for  $\beta$ .

Since for each key sound, the first peak of the spectrum should start from its fundamental frequency, thus, we will set it 0 to ignore these noise.

### 3.2.3 Tuning with Entropy Optimizer

The tuning process from a programming point of view is to move left or right of the array  $I(\cdot)$  as minor tuning process with  $+c$  cent shift.

$$I_k(\kappa - c) = \left\| G(f_{\kappa - c}) \right\|^\beta \quad (3.18)$$

The entropy function is defined as:

$$\text{Entropy}(x) = -x \cdot \log(x) \quad (3.19)$$

The entropy for a function is defined as:

$$\begin{aligned}\text{Entropy}(\phi(x)) &= \int_{-\infty}^{+\infty} (-\phi(x) \cdot \log(\phi(x))) dx \\ &= \sum_x (-\phi(x) \cdot \log(\phi(x)))\end{aligned}\quad (3.20)$$

Where  $\phi(\cdot)$  is the density function:

$$\begin{aligned}1 &= \int_{-\infty}^{+\infty} \phi(x) dx \\ &= \sum_x \phi(x)\end{aligned}\quad (3.21)$$

### 3.2.3.1 How to calculate the entropy value for the optimizer.

Since the algorithm optimize the case that all sound volume is equal, however, the sampling time is different, we will make a standard case to simulate all keys are pressed in an equal key pressing level. In my program, I use density function  $\bar{I}_k(\kappa)$  to simulate the equal key pressing level for each piano key sound in pitch domain:

$$\bar{I}_k(\kappa) = \frac{I_k(\kappa)}{\sum_{\kappa} (I_k(\kappa))} \quad (3.22)$$

When press all piano keys, the total volume  $V(\kappa)$  for each key pitch shift  $+c_k$  cents for tuning is:

$$V(\kappa) = \sum_k (\bar{I}_k(\kappa - c_k)) \quad (3.23)$$

The density function for this function is:

$$\bar{V}(\kappa) = \frac{V(\kappa)}{\sum_{\kappa} (V(\kappa))} \quad (3.24)$$

Then, the cost function value  $J$  (entropy value for function  $\bar{V}(\kappa)$ ) is:

$$J = \sum_{\kappa} (-\bar{V}(\kappa) \cdot \log(\bar{V}(\kappa))) \quad (3.25)$$

### 3.2.3.2 Steps to calculate a tuning strategy

In my program, there are several steps to dig out the good strategy for tuning.

- Step 1: Calculate the traditional tuning strategy which is a simpler version of the Traditional Tuning strategy, to be the initial starting point for entropy minimizer to begin. In this algorithm, no inharmonicity model is built, but just uses the captured frequency to optimize.
- Step 2: Randomly change tuning for one key for  $c_k$  cents, and check its entropy value. If the entropy value is smaller than last time, we keep this tuning strategy, otherwise, drop. Where the changing pitch is defined as a random number between 0 to some small number  $p$ . We will try both sides of tuning by adding and subtracting the pitches. The “A4” key never changes, since it is a standard pitch.
- Step 3: We do “step 2” experiment for all keys and all directions as one round of experiments. Each time we count the times of successfully tuned until we cannot find one round with no improvement.

- Step 4: We stop the algorithm with the test for  $p$  precision. Then we shrink the  $p$  and more accurate spectrum data (more data), and calculate “Step 2” and “Step 3”
- Step 5: Calculate tuning strategy and get the report.

In this process, “Step 1” is because the algorithm has many local minimums; although some local minimum can achieve similar simple and sharp harmony, it performs badly in simpler harmonies, such as an octave. A traditional tuning method can roughly optimize major overtones, the best result for entropy minimizer should be around the traditional tuning strategy.

In “Step 2”, although there should be more improvement during this step, however from a probability point of view, when it stops, the result is good enough for this precision. It could also use the parallel algorithm. In my program, I modeled several CPUs (not GPU program this time: GPU should calculate array sum much faster) with one shared memory to modify the result altogether. Although all CPUs will affect the overall result, however, if we can understand it will stop at the point that several CPUs could not find improvement, the effect is the same.

In “Step 4”, my program uses 3 round with 1, 0.5 and 0.2 cent boundaries as step size for entropy minimizers. Since there are many local minimums, and we need to achieve a smooth tuning strategy for not creating weird music scale sound, we cannot set the step size to be really large. Thus, 1 cent boundary is a good point to start. The next two round are precise tuning, the accuracy will be increased to 0.1 cent, which is desirable.

In “Step 5”, the frequency peak frequencies  $f_{k,n}$  are also captured by “catchup method”, but without weighted average.

### 3.2.4 Creating Tuning Strategy Table

The method to get the frequency components of each key sound is simple:

$$\tau'(k, n) = f_{k,n} \cdot C_{\rightarrow \text{fr}}(c_k) \quad (3.26)$$

However, this process is problematic. Since the whole process is based on pitch shift with a certain precision, the “A4” standard frequency will not be the fixed number. Here we need to eliminate this tuning error by introducing a correction factor  $\mathcal{E}_{[A4]}$ :

$$\mathcal{E}_{[A4]} = \frac{\tau'([A4], 1)}{\tilde{f}_{[A4]}} \quad (3.27)$$

Thus, the tuning strategy  $\tau(k, n)$  is modified to be:

$$\tau(k, n) = f_{k,n} \cdot C_{\rightarrow \text{fr}}(c_k) \cdot \mathcal{E}_{[A4]} \quad (3.28)$$

To build the tuning curve, the pitch deviation to the ideal frequency function  $C(k)$  is shown:

$$C(k) = \text{Fr}_{\rightarrow c} \left( \frac{\tau(k, n)}{\tilde{f}_k} \right) \quad (3.29)$$

The tuning strategy is shown in Figure 7-3.

The tuning curve is shown in Figure 3-10, the spectrum of the optimized result is shown in Figure 3-11:

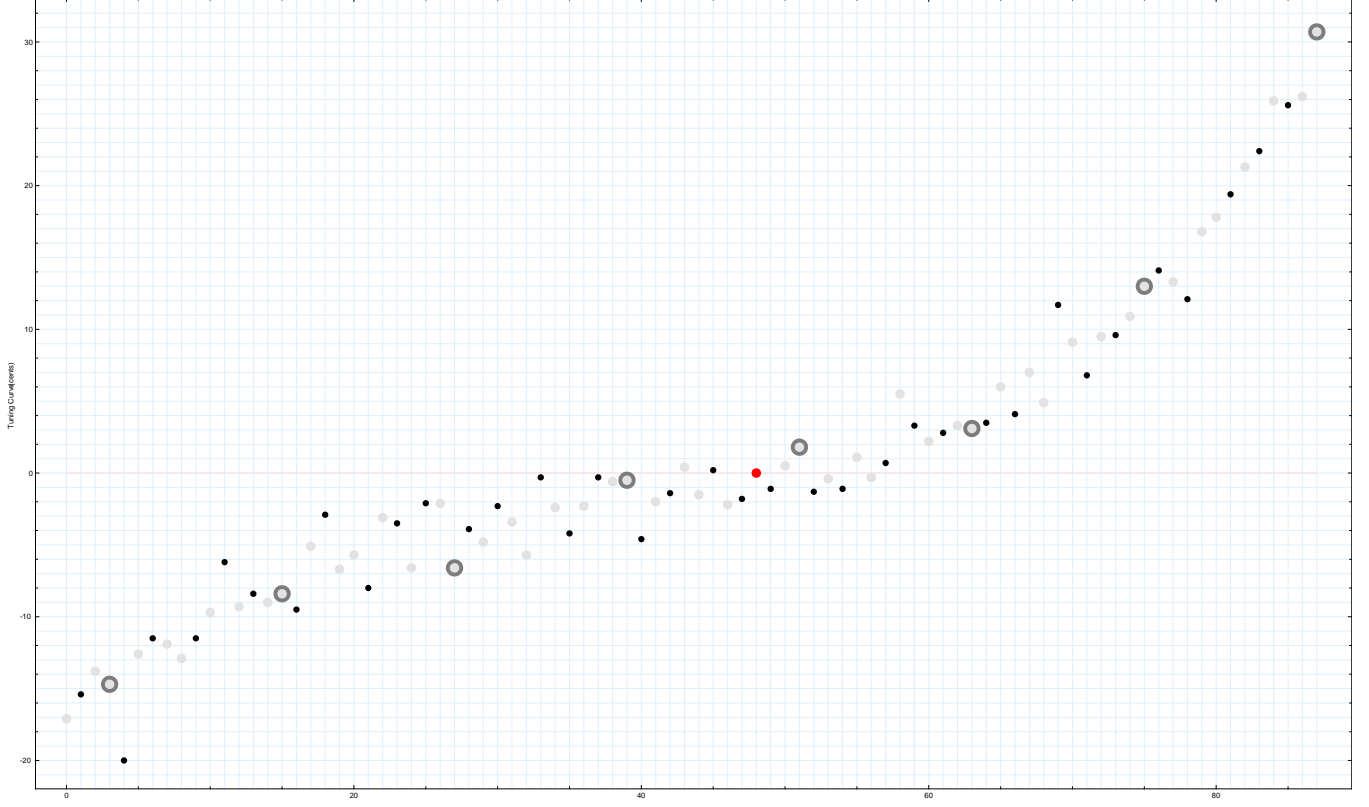


Figure 3-10 Tuning Curve for Upright Piano Optimized by Entropy Minimizer

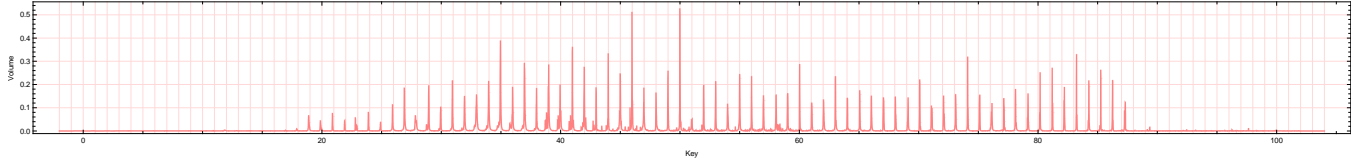


Figure 3-11 Spectrum for Optimized Result

From Figure 3-11, we could see the spectrum are largely merged. From the sound quality point of view, the harmony will sound sharp and clear.

### 3.2.5 Tune for Songs

In the real world, some of the piano keys have not been used, especially for the simpler tonal music. Since I have mentioned the previous entropy minimizer is not quite suitable for simpler harmony music due to some of the simple harmony like octave sometimes will not sound perfect, we should ignore the keys that have not been used. Thus, I add another coefficient for the entropy minimizer.

We will put the bias  $\text{Bias}_k$  that will ignore the key  $k$  which have not been used.

$$\text{Bias}_k = \begin{cases} 1 & k \in \text{used} \\ \varepsilon_{\text{Bias}} & k \notin \text{used} \end{cases} \quad (3.30)$$

Where  $\varepsilon_{\text{Bias}}$  is a very small number – to make sure the key which is not used could be tuned by the entropy minimizer. If the bias for one key is 0, there is no spectrum for entropy minimizer for this key, and the algorithm



will stop tuning for this key. However, if we put a very small number as weight on this key, it still can be tuned to a correct place – it just tuned, but does not affect the tuning for other keys.

Then, we will put the bias on the entropy minimizer algorithm and modify the Equation (3.25):

$$J = \sum_{\kappa} \left( -\text{Bias}_{\kappa} \cdot \bar{V}(\kappa) \cdot \log(\bar{V}(\kappa)) \right) \quad (3.31)$$

Then, we use the method above to minimize this entropy function and get the tuning strategy.

From the example of one tonal music from Mozart Piano Sonata No 11 A major K 331 – Movement 1 (Figure 3-12), we could see only the middle range and several low range keys are used.

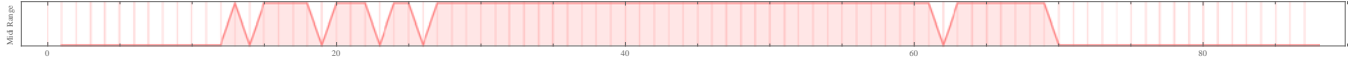


Figure 3-12 Song Key Used Cases

The optimized spectrum is shown in Figure 3-13.

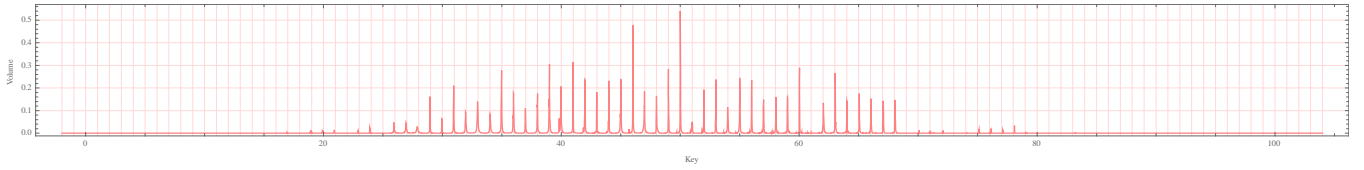


Figure 3-13 Optimized Spectrum

From this example, we can see and hear, the sound will be more optimized whenever in simple and complicated harmonies.

## 4 AUDIO PROCESSING & PURE SOUND TUNER

### 4.1 TUNING

Tuning process in an audio is to create samples for the virtual instrument so that we can hear the tuning result before tuning process to make a decision whether to adopt or drop this tuning strategy.

The sound function  $S(t)$  tunes in order to add pitch  $c$  cents:

$$S_{+c}(t) = S\left(t \cdot 2^{\left(\frac{c}{1200}\right)}\right) \quad (4.1)$$

The  $S(t)$  function is modeled as an interpolation function.

### 4.2 SOUND PURIFY

This audio processing technique is invented by myself. It removes the inharmonic effect of piano sound.

Since the inharmonicity model has been built, it is possible to use the audio processing technique to shrink the harmonics in order to remove the inharmonicity.

If the key  $k$  sound with the inharmonicity coefficient  $IH(k)$  and tuned to the fundamental frequency to be the frequency (ideal frequency)  $\tilde{f}_k$ ; the  $f_k$  is the fundamental frequency.

We firstly get the FFT of the audio sample with  $\Gamma_k(f)$  of complex number samples:

$$\Gamma_k(f) = \text{FFT}(S_k(t)) \quad (4.2)$$

Since the FFT is creating an almost symmetry data from the middle, we can extract this data into 4 parts: the real head data  $\Gamma_k^{(0)}(f)$ , the imaginary head data  $\Gamma_k^{(1)}(f)$ , the real tail reverse data  $\Gamma_k^{(2)}(f)$  and the tail imaginary reverse data  $\Gamma_k^{(3)}(f)$ . Four of them looks similar, however, it contains all the details of the sound. Since it samples the piano keys, the spectrum is pretty obvious. At its high frequencies, it is almost 0, and it is almost out of hearing range, thus if we need to compress the frequency domain, as for higher frequencies, we could regard it to be 0. For each component we write it as  $\Gamma_k^{(m)}(f)$ , where  $m$  is from 0 to 3 (4 cases),  $i$  is the unit imaginary number.

$$\Gamma_k(f) = \left\{ \Gamma_k^{(0)}(f), \text{rev}(\Gamma_k^{(2)}(f)) \right\} + \left\{ \Gamma_k^{(1)}(f), \text{rev}(\Gamma_k^{(3)}(f)) \right\} \cdot i \quad (4.3)$$

From Equation (3.6) and Equation (3.7), we could get the compression functions, which is  $\tau(k, n)$ . Here the overtone is continuous, which is  $f / f_k$ , rather than  $n$ . Thus, we have the compressed frequency scaler  $\tilde{f}_k$  and its pitch component  $\tilde{\Gamma}_k^{(m)}(f)$ :

$$\tilde{f}_k = \tilde{f}_k \cdot \tau\left(k, \frac{f}{f_k}\right) \quad (4.4)$$

$$\tilde{\Gamma}_k^{(m)}(f) = \begin{cases} \Gamma_k^{(m)}(\tilde{f}_k) & \tilde{f}_k \in \text{defined} \\ 0 & \tilde{f}_k \notin \text{defined} \end{cases} \quad (4.5)$$

Where  $\Gamma_k^{(m)}(f)$  and  $\tilde{\Gamma}_k^{(m)}(f)$  will be the same size as samples.

Use the interpolation function to stretch, and do this for four functions; then, combine them in an original way, and use inverse Fourier function to restore the audio  $\tilde{S}_k(t)$ .

$$\tilde{\Gamma}_k(f) = \left\{ \tilde{\Gamma}_k^{(0)}(f), \text{rev}(\tilde{\Gamma}_k^{(2)}(f)) \right\} + \left\{ \tilde{\Gamma}_k^{(1)}(f), \text{rev}(\tilde{\Gamma}_k^{(3)}(f)) \right\} \cdot i \quad (4.6)$$

$$\tilde{S}_k(t) = \text{Re}(\text{invFFT}(\tilde{\Gamma}_k(f))) \quad (4.7)$$

Where  $i$  is an imaginary number,  $\text{invFFT}(\cdot)$  is the inverse FFT,  $\text{Re}(\cdot)$  is to get the real part of a number or array,  $\text{rev}(\cdot)$  is the reverse of an array.

Then, do this for 2 channels and create the audio as Pure Sound Tuner result.

From this function, it needs 3 data: the audio data  $S_k(t)$ , the inharmonicity coefficient  $\text{IH}(k)$ , and its fundamental frequency  $f_k$  (which could be captured by audio data).

## 5 FUTURE WORK

Over-pull tuning is implemented in some tuning apps, and I do not know its method. Since I am still lacking of research in this area, I will leave it as future work to think about. I know this effect is caused by the experimental result of the percentage that the tuning pins will loosen and drop the pitch, it should have the correction coefficient for the tuner will make up the errors of this effect by over pull to tune the frequency higher than its actual one.

## 6 REFERENCE

[1] Hinrichsen, Haye. "Entropy-based tuning of musical instruments." *Revista brasileira de Ensino de Física* 34.2 (2012): 1-8.

[2] Github for Piano Tuning Project [[https://github.com/RobertBoganKang/piano\\_tuning](https://github.com/RobertBoganKang/piano_tuning)]

# 7 APPENDIX

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16																
A0	27.3985	6.330	54.8092	0.010	82.244	0.330	109.715	-4.400	137.235	3.340	164.815	-1.040	192.467	-0.200	220.203	1.60	248.036	-0.370	275.977	-1.140	304.037	0.780	332.228	-11.000	360.563	-13.770	389.052	-16.130	417.706	-21.270	446.537	-25.040
A#0	29.0357	5.920	58.0836	0.560	87.1556	-0.860	116.264	-1.120	145.42	-0.040	174.637	-1.720	203.927	-1.180	233.1	-1.030	262.77	-0.860	292.347	-0.510	322.044	0.390	351.873	-11.110	381.844	-14.050	411.97	-17.220	442.261	-20.610	472.729	-24.220
B0	30.7704	-5.460	61.5517	-1.510	92.3548	-0.640	123.191	0.330	154.07	-0.010	185.005	-1.680	216.004	-0.950	247.08	-0.370	278.243	-2.710	309.503	-0.630	340.872	0.760	372.359	-0.000	403.976	-11.610	435.733	-14.310	467.64	-17.210	499.708	-20.360
C1	32.6082	-0.530	65.2258	-4.780	97.8625	-4.360	130.528	0.370	163.231	-0.020	195.981	-2.060	228.789	-1.070	261.663	-0.250	294.612	-1.670	327.646	-0.250	360.775	5.0	394.007	-0.910	427.352	-0.980	460.819	-11.220	494.417	-13.610	528.155	-16.160
C#1	34.5552	-6.930	69.1189	-4.420	103.7	-0.060	138.306	0.590	172.947	-2.910	207.631	-2.130	242.366	-1.260	277.161	0.130	312.024	-1.080	346.964	-0.430	381.989	0.320	417.108	-5.550	452.328	-7.320	487.659	-9.220	523.108	-11.260	558.684	-13.440
D1	36.6178	-4.260	73.2437	-0.070	109.886	-0.370	146.553	0.36	183.252	-2.720	219.992	-2.010	256.781	-1.180	293.627	0.220	330.539	-0.870	367.523	0.090	404.588	0.430	441.743	-4.890	478.994	-6.480	516.351	-8.260	553.821	-10.040	591.412	-12.0
E1	38.8028	-3.920	77.6145	3.720	116.444	0.300	155.3	2.830	194.191	-2.340	233.127	-1.610	272.116	0.760	311.166	-0.220	350.287	-1.340	389.488	-0.580	428.777	0.350	468.162	-5.450	507.653	-7.080	547.257	-8.840	586.985	-10.720	626.843	-12.730
F1	41.1175	-3.610	82.2444	3.410	123.39	0.090	164.563	0.620	205.774	2.040	247.031	-0.330	288.343	-0.480	329.721	0.490	371.172	-0.460	412.707	-0.830	454.334	1.190	496.063	-5.670	537.903	-7.290	579.862	-9.030	621.949	-10.890	664.175	-13.880
F#1	43.6896	-0.330	87.1476	-0.160	130.743	0.380	174.363	2.480	218.018	-1.380	261.715	-1.390	305.463	-0.830	349.27	0.210	393.145	-1.160	437.096	-0.220	481.132	3.44	525.261	-4.890	569.491	-6.080	613.83	-7.680	658.287	-9.190	702.871	-10.910
G1	46.167	0.080	92.3423	2.920	138.534	-0.670	184.75	-2.310	231.184		277.29	-1.280	323.63	-0.610	370.028	0.160	416.491	-1.030	463.028	-0.0	509.647	3.080	556.356	-4.250	603.163	-5.630	650.076	-6.910	697.104	-8.380	744.254	-9.950
G#1	48.9185	-2.860	97.8452	-2.710	146.789	-0.470	195.757	0.130	244.758	-1.690	293.8	-1.150	342.892	-0.520	392.041	0.210	441.257	-1.030	490.546	-0.950	539.918	2.970	589.38	-4.080	638.941	-5.290	688.608	-6.590	738.39	-7.950	788.295	-9.480
A1	51.8331	-2.660	103.675	2.530	155.532	2.30	207.415	1.970	259.331	-1.560	311.287	-1.060	363.294	-0.460	415.358	0.230	467.488	-1.0	519.691	-1.870	571.977	2.830	624.354	-3.880	676.828	-5.020	729.409	-6.250	782.105	-7.490	834.924	-8.670
A#1	54.9206	-2.50	109.391	-2.260	164.795	-2.140	219.766	-1.840	274.77	-1.440	329.816	-0.960	384.912	-0.360	440.067	0.270	495.288	-1.010	550.584	-1.840	605.963	2.760	661.434	-3.760	717.005	-4.850	772.683	-6.030	828.478	-7.290	884.397	-8.630
B1	58.1912	-2.350	116.819	-2.220	174.609	-2.0	232.853	-1.760	291.133	-1.360	349.456	-0.820	407.833	-0.250	466.271	0.40	524.78	-1.150	583.368	-1.980	642.044	2.890	700.817	-3.890	759.695	-4.980	818.688	-6.150	877.903	-7.410	937.049	-8.750
C2	61.6557	-2.220	123.321	-2.10	185.004	-2.000	246.714	-1.590	308.46	-1.210	370.252	-0.740	432.099	-0.190	494.009	0.450	555.991	-1.170	618.055	-1.970	680.21	-2.890	742.464	-3.920	804.827	-4.930	867.307	-6.030	929.913	-7.250	992.654	-8.550
C#2	65.3258	-2.100	130.66	-2.010	196.012	-1.920	261.39	-1.550	326.803	-1.210	392.259	-0.710	457.767	-0.200	523.337	0.290	588.975	-0.940	654.692	-1.670	720.495	2.470	786.393	-3.520	852.385	-4.590	918.61	-5.620	984.745	-6.820	1051.11	-7.610
C2	69.2137	-2.040	138.437	-1.920	207.681	-1.710	276.954	-1.420	346.268	-1.050	415.631	-0.530	485.053	-0.050	554.546	0.270	624.118	-1.270	693.779	-2.060	763.539	2.930	833.409	-3.880	903.398	-4.810	973.515	-5.820	1043.77	-7.210	1114.17	-8.480
D2	73.3322	-1.860	146.677	-1.820	220.046	-1.590	293.453	-1.250	366.908	-0.810	440.426	-0.210	514.018	-0.360	587.696	-1.080	661.472	-1.910	735.36	-2.820	809.371	3.840	883.517	-4.950	957.81	-5.960	1032.26	-7.460	1106.89	-8.860	1181.7	-10.340
D#2	76.6953	-1.620	155.405	-1.780	243.143	-1.50	319.024	-1.120	388.763	-0.660	466.672	-0.060	544.668	-0.830	622.764	-1.420	700.373	-2.320	779.311	-3.330	857.792	4.440	936.429	-5.650	1015.82	-6.670	1095.07	-7.840	1173.42	-9.010	1252.82	-11.130
E2	82.3174	-1.880	164.651	-1.710	247.017	-1.420	329.431	-1.030	411.91	-0.520	494.47	-0.140	577.126	-0.840	659.895	-1.600	742.794	-2.640	825.837	-3.710	909.04	-4.80	992.421	-6.190	1075.99	-7.580	1159.77	-9.10	1243.78	-10.290	1328.02	-12.040
F2	87.214	-1.840	174.446	-1.660	261.713	-1.370	349.033	-0.860	436.424	-0.430	523.904	-0.210	611.49	-0.570	699.2	-1.650	787.05	-2.640	875.06	-3.690	963.246	5.170	1051.63	-6.560	1140.22	-8.030	1229.04	-9.520	1318.1	-11.20	1407.43	-12.950
F#2	92.4016	-1.810	184.823	-1.620	277.286	-1.340	369.809	-0.860	462.413	-0.390	555.19	-0.40	647.946	-1.230	740.914	-2.240	834.044	-3.240	927.356	-4.430	1020.87	-5.760	1114.6	-7.26	1208.58	-8.760	1302.82	-10.040	1397.33	-12.260	1492.15	-14.180
G2	97.8974	-1.720	195.818	-1.500	293.785	-1.240	391.823	-0.780	489.953	-0.410	588.199	-0.470	686.586	-1.50	785.134	-2.530	883.869	-3.600	982.812	-4.690	1081.99	-6.420	1181.42	-7.880	1281.12	-9.080	1381.13	-11.560	1481.46	-13.050	1582.15	-15.550
G#2	103.72	-1.770	207.465	-1.560	311.259	-1.210	415.129	-0.720	519.098	-0.110	623.191	-0.660	727.434	-1.550	831.851	-2.590	936.466	-3.760	1041.3	-5.070	1146.39	-6.510	1251.75	-8.090	1357.4	-9.610	1463.38	-11.650	1569.7	-13.620	1676.38	-15.740
A2	109.889	-1.750	219.804	-1.540	329.773	-1.030	439.822	-0.690	549.979	-0.060	660.27	-0.710	770.721	-1.620	881.359	-2.680	992.21	-3.870	1103.3	-5.190	1214.66	-6.660	1326.31	-8.250	1438.28	-10.0	1550.8	-11.660	1663.28	-13.590	1776.36	-16.020
A#2	116.424	-1.730	232.679	-1.50	349.393	-1.110	466.008	-0.670	582.744	-0.120	699.635	-0.670	816.713	-1.570	934.008	-2.620	1051.55	-4.430	1169.37	-5.880	1287.5	-7.490	1405.98	-9.250	1524.82	-11.150	1644.06	-13.260	1763.73	-15.40	1883.86	-18.050
B2	123.349	-1.710	246.734	-1.450	370.193	-0.920	493.673	-0.420	617.481	-0.360	741.382	-1.310	865.505	-2.430	989.885	-3.710	1114.56	-5.170	1239.560	-6.880	1364.93	-8.590	1490.71	-10.550	1616.92	-12.880	1743.03	-14.970	1870.79	-17.620	1998.53	-20.040
C3	130.685	-1.680	261.413	-1.44	392.229	-0.830	523.174	-0.250	654.293	-0.610	785.629	-1.010	917.225	-1.910	1048.132	-3.160	1187.37	-4.580	1314	-7.760	1447.07	-9.750	1580.6	-11.920	1714.65	-13.780	1849.26	-16.830	1984.47	-18.650	2120.31	-22.450
C#3	138.458	-1.680	276.967	-1.340	415.579	-0.910	554.345	-0.360	693.316	-0.360	832.544	-0.980	972.079	-1.840	1111.97	-3.060	1252.27	-4.880	1393.03	-6.870	1534.3	-11.090	1676.13	-13.520	1818.56	-16.140	1961.65	-18.970	2105.45	-22.0	2250	-25.290
D3	146.694	-1.630	293.447	-1.280	440.321	-0.690	587.376	-0.140	734.67	-1.340	882.265	-0.510	1030.22	-1.420	1178.59	-2.620	1324.47	-3.780	1476.83	-5.020	1626.81	-12.450	1777.45	-15.130	1928.78	-18.020	2080.21	-21.140	2233.84	-24.490	2387.66	-28.030
D#3	155.42	-1.60	310.906	-1.220	465.527	-0.620	623.249	-0.270	778.439	-1.390	934.864	-2.170	1091.69	-3.360	1248.98	-4.220	1406.81	-5.910	1565.23	-7.660	1724.32	-13.290	1884.13	-16.040	2044.74	-18.090	2206.2	-22.370	2368.58	-25.670	2531.94	-28.610
E3	164.665	-1.560	329.403	-1.160	494.299	-0.480	659.42	-0.440	824.84	-1.630	990.646	-3.090	1156.4	-4.910	1323.69	-6.780	1491.07	-9.020	1659.14	-11.520	1827.95	-14.270	1997.59	-17.270	2168.12	-20.520	2339.62	-24.020	2512.16	-27.760	2685.8	-31.740
F3	174.461	-1.510	343.013	-1.060	523.476	-0.310	698.75	-0.730	874.116	-2.080	1049.93	-3.720	1226.29	-5.050	1403.28	-7.870	1589.08	-10.390	1759.49	-13.190	1938.89	-16.270	2119.27	-19.640	2300.72	-23.290	2483.31	-26.710	2667.14	-31.40	2852.28	-35.050
F#3	184.841	-1.460	369.789	-0.960	554.95	-0.120	740.432	-1.040	926.34	-2.540	1112.78	-4.360	1299.86	-5.510	1487.68	-6.960	1676															

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16																
A0	27.1975	12.500	54.4302	11.7335	109.142	132.691	102.219	164.415	6.36	192.348	128.9	220.524	1.120	248.975	102.96	277.736	17.740	306.837	24.890	336.31	102.790	366.185	41.590	396.494	60.930	427.264	18.890	458.525	21.46			
A#0	28.8329	18.020	57.7015	16.980	86.6414	15.20	115.688	12.710	144.877	9.020	174.243	6.80	203.821	1.070	233.644	4.180	263.747	10.080	294.163	16.620	324.925	23.810	356.063	31.870	387.611	40.4	419.597	48.880	452.053	48.850	485.006	48.40
B0	30.5664	18.080	61.1693	16.940	91.845	14.230	122.63	11.830	153.56	8.750	184.671	6.090	215.998	0.810	247.576	4.450	279.44	10.140	311.624	16.460	344.162	23.300	377.085	30.970	410.426	38.020	444.216	47.690	478.487	48.890	513.267	46.650
B1	32.4038	18.020	64.8424	14.930	97.3508	13.440	129.964	11.270	162.715	8.490	195.64	4.110	228.773	1.130	262.146	4.540	295.793	8.930	329.746	14.370	364.039	20.900	398.702	27.420	433.768	34.780	469.265	47.690	505.224	51.040	541.674	50.910
C#1	34.3511	14.880	68.7389	10.380	103.2	12.420	137.771	10.270	172.488	7.510	207.388	4.160	242.506	0.90	277.877	3.340	313.536	4.940	349.519	15.130	385.858	21.370	422.587	28.140	459.739	35.450	497.347	43.280	535.44	51.80	574.051	50.410
D#1	36.4148	13.080	72.8629	10.090	109.378	11.770	145.992	9.930	182.74	7.590	219.654	4.880	256.766	1.280	294.108	4.620	331.714	7.020	369.613	11.910	407.839	17.280	446.42	23.130	485.388	28.440	524.737	35.210	564.602	43.410	604.906	51.610
E1	38.6017	12.820	77.2388	10.120	115.947	9.360	154.761	8.950	193.716	6.590	232.847	3.70	272.188	0.90	311.775	3.610	351.641	6.010	391.819	12.970	432.342	18.290	473.244	24.150	514.556	30.470	556.309	37.240	598.536	46.460	641.265	52.510
F1	40.9191	11.980	81.8729	11.250	122.896	10.020	164.024	8.910	205.29	6.120	246.729	3.440	288.375	0.290	330.263	3.340	372.424	7.420	414.893	11.970	457.701	16.970	500.882	22.410	544.467	28.200	588.487	34.500	632.973	41.260	677.955	48.420
F#1	43.3745	11.14	86.7867	10.340	130.274	8.900	173.875	7.340	217.625	5.14	261.563	2.370	305.725	0.850	350.147	4.850	394.866	8.730	439.919	13.370	485.339	18.470	531.162	24.030	577.422	30.020	624.152	36.450	671.387	43.370	719.159	50.570
G1	45.9762	10.250	91.9917	9.510	138.085	8.260	184.296	6.570	230.663	4.370	277.225	1.880	324.02	1.480	371.086	5.110	418.461	9.20	466.182	13.780	514.286	17.870	562.809	24.220	611.787	30.110	661.254	36.420	711.246	43.370	761.797	50.290
G#1	48.7328	9.440	97.4972	8.880	146.325	7.350	195.247	6.460	244.295	4.980	293.501	2.390	342.896	0.50	392.509	2.270	442.373	6.410	492.517	9.380	542.972	12.740	593.768	18.920	644.933	21.450	696.497	28.280	748.489	31.510	800.937	37.030
A1	51.6533	8.680	103.337	8.160	155.084	7.30	206.923	6.090	258.885	4.540	311.002	2.840	363.304	0.410	415.821	2.160	468.583	5.090	521.62	8.290	574.963	11.880	628.639	18.730	682.679	19.920	737.11	24.430	791.962	29.250	847.262	34.370
A#1	54.7473	7.870	109.528	7.450	164.374	6.580	219.319	5.380	274.395	4.810	329.636	1.970	385.074	0.940	440.74	3.910	496.667	6.830	552.888	9.070	609.432	12.640	666.332	18.540	723.619	20.750	781.323	28.280	839.474	36.110	898.101	38.250
B1	58.0252	7.30	116.084	6.80	174.209	5.880	232.433	4.820	290.79	3.340	349.313	1.620	408.035	0.490	466.987	3.080	526.202	6.930	585.714	8.920	645.552	12.320	705.749	18.030	766.337	20.080	827.346	24.360	888.806	28.970	950.748	33.880
B#1	61.4978	6.570	123.034	6.050	184.671	5.010	246.434	3.580	308.374	1.690	370.535	0.980	432.96	0.380	495.692	3.330	558.775	6.910	622.25	13.880	686.16	17.940	750.546	23.570	815.449	27.580	880.91	32.970	946.969	38.710	1013.66	44.810
C2	65.1765	6.000	130.394	5.550	196.694	4.040	261.116	3.570	326.703	1.740	392.494	0.250	458.53	0.260	524.852	2.580	591.488	6.340	658.509	11.730	725.924	15.470	793.782	19.540	862.121	23.940	930.979	29.670	1000.399	37.710	1070.60	44.210
C#2	69.0735	5.550	138.184	5.090	207.368	3.320	276.664	3.340	346.106	1.850	415.734	0.160	485.581	1.830	555.686	1.130	626.085	6.720	696.812	10.610	767.903	12.790	839.395	18.270	913.321	20.020	983.716	24.070	1056.610	31.040	1130.05	32.980
D2	73.2017	5.060	146.445	4.570	219.772	2.740	293.224	2.590	366.843	1.120	440.669	0.080	514.745	0.280	589.111	0.250	663.807	0.910	738.875	11.080	814.353	14.470	890.282	18.160	966.701	21.760	1045.65	26.900	1121.16	31.040	1199.28	35.920
D#2	77.5479	4.610	155.188	4.040	232.876	1.490	310.678	2.50	388.631	1.230	466.772	0.370	545.14	0.230	623.771	1.420	702.702	6.580	781.97	9.320	861.611	12.130	941.663	16.360	1022.16	18.730	1103.14	22.420	1184.63	28.370	1266.68	35.570
E2	82.2074	4.190	164.456	3.780	246.786	1.040	329.24	1.030	411.857	0.740	494.679	0.840	577.745	0.210	661.097	1.480	744.775	7.290	828.817	9.950	913.264	12.020	998.156	16.160	1083.53	19.870	1169.43	23.450	1255.88	27.480	1342.93	31.780
F2	87.1147	3.810	174.271	3.40	261.511	0.710	348.877	1.740	436.409	0.560	524.149	1.020	612.139	0.410	700.42	4.880	789.032	7.190	878.016	8.780	967.413	12.640	1057.26	15.760	1147.16	18.140	1239.92	22.770	1329.88	28.860	1421.97	35.730
F#2	92.3131	3.470	184.671	3.050	277.12	0.340	369.703	1.360	462.466	0.490	555.454	1.450	648.712	0.370	748.538	3.060	836.211	7.790	930.541	10.370	1025.32	13.260	1120.58	16.450	1216.37	19.880	1312.74	23.980	1409.72	27.540	1507.36	31.740
G2	97.8198	3.160	195.687	2.740	293.649	0.490	391.754	1.060	490.048	0.190	588.579	1.730	687.393	0.840	786.508	2.620	886.001	7.980	986.004	10.860	1086.42	13.490	1187.35	16.850	1288.83	20.600	1380.65	24.070	1493.66	27.870	1597.09	31.850
A#2	103.653	2.880	207.358	2.450	331.168	1.720	415.133	0.710	519.307	0.100	623.741	0.180	728.486	0.090	833.593	0.210	939.114	0.850	1045.1	11.370	1151.6	14.380	1258.66	17.820	1366.34	21.760	1474.68	24.070	1583.73	28.080	1693.53	33.360
G#2	109.832	2.590	219.716	2.340	329.7	1.570	439.836	0.840	550.173	0.050	660.764	0.210	771.656	0.370	882.9	0.70	994.546	2.780	1106.64	10.420	1219.24	13.170	1332.38	16.170	1446.12	19.610	1560.51	22.910	1675.59	28.860	1791.4	35.250
A#2	116.378	2.470	232.812	2.0	349.356	1.310	466.066	0.900	582.997	0.070	700.204	0.280	817.741	0.410	935.662	0.190	1054.02	0.490	1172.88	11.060	1292.27	13.890	1412.27	16.980	1532.92	20.020	1654.28	23.020	1776.38	27.780	1899.3	31.870
B2	123.313	2.210	246.683	1.810	370.167	1.140	523.824	0.32	617.709	0.6	741.88	0.470	866.393	0.240	991.305	0.26	1116.87	0.450	1242.55	10.870	1368.99	13.740	1496.06	16.760	1623.8	20.030	1752.27	23.660	1881.52	27.320	2011.61	31.330
C3	130.659	2.040	261.375	1.650	392.208	0.710	593.214	0.120	654.452	0.030	785.979	0.240	917.853	0.040	1050.13	0.8	1182.67	0.810	1316.12	10.550	1449.95	13.20	1584.41	16.090	1719.55	19.220	1855.43	22.590	1992.1	28.260	2129.62	30.030
C#3	138.44	1.880	276.94	1.510	415.557	0.360	554.35	0.040	693.378	0.080	832.7	0.40	972.373	0.090	1112.46	1.810	1253.7	0.880	1394.08	10.180	1535.74	12.710	1678.03	15.480	1821.02	18.480	1964.76	21.710	2109.43	25.160	2254.75	28.840
D3	146.684	1.780	293.403	1.540	440.191	1.20	597.082	0.730	734.11	0.120	881.311	0.830	1028.72	0.150	1176.36	0.250	1329.22	0.860	1472.52	9.340	1621.09	9.350	1770.47	9.890	1919.97	12.560	2083.65	16.710	2219.37	20.920	2370.25	25.370
D#3	155.418	1.620	310.872	1.420	468.399	0.980	622.034	0.610	777.816	0.40	933.778	0.740	1089.96	0.020	1246.39	0.830	1403.12	0.370	1560.17	1.040	1717.58	6.450	1875.38	9.780	2033.62	8.860	2192.33	11.450	2351.54	13.370	2511.29	15.430
E3	164.67	1.570	329.383	1.280	494.181	0.910	659.109	0.380	824.208	0.380	989.522	1.120	1155.09	0.210	1396.39	0.320	1487.18	0.450	1653.78	0.920	1820.8	7.480	1988.29	9.20	2156.3	11.080	2324.86	14.940	2494.16	16.20	2663.79	17.490
F3	174.472	1.410	349.065	1.060	523.863	0.670	699.006	0.370	874.595	0.030	1050.74	0.500	1227.56	0.440	1405.15	0.180	1583.63	0.230	1763.1	18.740	1943.67	20.540	2125.45	24.060	2388.54	26.970	2593.08	33.080	2679.07	38.120	2866.7	44.590
F#3	184.856	1.320	369.834	0.740	555.058	0.210	74																									

	1	2	3	4	5	6	7	8	9	10	11	12
A0	27.4413 +0.3%	54.6261 +0.9%	82.0674 +1.5%	109.765 +2.1%	137.463 +2.8%	165.417 +3.4%	193.884 +4.0%	222.095 +4.6%	251.075 +5.2%	280.055 +5.8%	309.804 +6.4%	339.297 +6.9%
A#0	28.8409 +0.3%	57.6818 +0.7%	86.5226 +1.1%	115.579 +1.5%	144.635 +1.9%	173.906 +2.3%	204.038 +2.7%	233.525 +3.1%	264.088 +3.5%	294.65 +3.9%	325.428 +4.3%	356.422 +4.7%
B0	30.6368 +0.4%	61.2735 +0.8%	91.9103 +1.2%	122.746 +1.6%	153.98 +2.0%	185.213 +2.4%	216.845 +2.8%	248.675 +3.2%	280.903 +3.6%	313.33 +4.0%	345.758 +4.4%	378.981 +4.8%
C1	32.4352 +0.4%	64.8703 +0.8%	97.4898 +1.2%	130.109 +1.6%	163.097 +2.0%	196.454 +2.4%	229.626 +2.8%	263.72 +3.2%	297.445 +3.6%	331.539 +4.0%	365.633 +4.4%	400.648 +4.8%
C#1	34.1767 +0.5%	68.502 +0.9%	102.679 +1.3%	137.301 +1.7%	171.924 +2.1%	207.14 +2.5%	242.209 +2.9%	277.723 +3.3%	313.385 +3.7%	349.494 +4.1%	385.751 +4.5%	422.305 +4.9%
D1	36.5828 +0.6%	73.0243 +1.0%	109.748 +1.4%	146.614 +1.8%	183.338 +2.2%	220.768 +2.6%	258.339 +3.0%	296.052 +3.4%	334.189 +3.8%	372.184 +4.2%	410.744 +4.6%	449.304 +5.0%
D#1	38.6613 +0.6%	77.3226 +1.0%	115.984 +1.4%	155.044 +1.8%	194.303 +2.2%	233.761 +2.6%	273.419 +3.0%	313.276 +3.4%	353.532 +3.8%	393.787 +4.2%	434.441 +4.6%	475.295 +5.0%
E1	40.8872 +0.7%	81.9168 +1.1%	122.946 +1.5%	164.404 +1.9%	205.718 +2.3%	247.46 +2.7%	289.06 +3.1%	331.371 +3.5%	373.826 +3.9%	416.85 +4.3%	459.304 +4.7%	502.613 +5.1%
F1	43.2485 +0.7%	86.6554 +1.1%	130.062 +1.5%	173.786 +1.9%	217.668 +2.3%	261.867 +2.7%	306.225 +3.1%	350.899 +3.5%	395.415 +3.9%	440.723 +4.3%	486.031 +4.7%	531.972 +5.1%
F#1	45.9597 +0.8%	91.9195 +1.2%	138.012 +1.6%	184.372 +2.0%	230.731 +2.4%	277.623 +2.8%	324.516 +3.2%	371.808 +3.6%	419.366 +4.0%	467.191 +4.4%	515.282 +4.8%	564.039 +5.2%
G1	48.8379 +0.8%	97.5044 +1.2%	146.514 +1.6%	195.694 +2.0%	244.703 +2.4%	293.713 +2.8%	343.236 +3.2%	392.931 +3.6%	442.968 +4.0%	493.005 +4.4%	543.557 +4.8%	594.451 +5.2%
G#1	51.5976 +0.9%	103.379 +1.3%	154.976 +1.7%	206.941 +2.1%	258.722 +2.5%	310.687 +2.9%	363.019 +3.3%	415.351 +3.7%	465.664 +4.1%	520.199 +4.5%	574.367 +4.9%	627.985 +5.3%
A1	54.5567 +0.9%	109.325 +1.3%	164.093 +1.7%	218.861 +2.1%	273.841 +2.5%	329.032 +2.9%	384.435 +3.3%	440.049 +3.7%	495.874 +4.1%	551.7 +4.5%	608.371 +4.9%	665.254 +5.3%
A#1	57.9676 +1.0%	115.935 +1.4%	174.057 +1.8%	232.487 +2.2%	290.609 +2.6%	349.039 +3.0%	407.623 +3.4%	466.515 +3.8%	525.562 +4.2%	585.38 +4.6%	645.352 +5.0%	705.632 +5.4%
B1	61.2533 +1.0%	122.864 +1.4%	184.474 +1.8%	246.442 +2.2%	308.41 +2.6%	370.556 +3.0%	432.524 +3.4%	495.027 +3.8%	558.423 +4.2%	621.641 +4.6%	685.573 +5.0%	749.862 +5.4%
C2	65.2473 +1.1%	130.233 +1.5%	195.742 +1.9%	261.261 +2.3%	326.78 +2.7%	392.571 +3.1%	458.906 +3.5%	524.969 +3.9%	591.576 +4.3%	658.454 +4.7%	726.148 +5.1%	794.386 +5.5%
C#2	69.1917 +1.1%	137.766 +1.5%	207.043 +1.9%	276.321 +2.3%	345.441 +2.7%	415.034 +3.1%	484.785 +3.5%	554.693 +3.9%	625.233 +4.3%	695.931 +4.7%	766.629 +5.1%	838.589 +5.5%
D2	73.2518 +1.2%	146.432 +1.6%	220.086 +2.0%	293.301 +2.4%	367.394 +2.8%	440.829 +3.2%	515.36 +3.6%	589.453 +4.0%	664.641 +4.4%	739.83 +4.8%	815.457 +5.2%	890.865 +5.6%
D#2	77.3853 +1.2%	155.26 +1.6%	232.523 +2.0%	310.642 +2.4%	387.905 +2.8%	465.779 +3.2%	544.876 +3.6%	622.995 +4.0%	702.336 +4.4%	780.822 +4.8%	860.407 +5.2%	940.82 +5.6%
E2	82.0508 +1.3%	164.102 +1.7%	246.583 +2.1%	328.921 +2.5%	411.402 +2.9%	493.739 +3.3%	576.794 +3.7%	660.136 +4.1%	744.052 +4.5%	827.537 +4.9%	912.313 +5.3%	997.09 +5.7%
F2	86.8265 +1.3%	173.957 +1.7%	261.544 +2.1%	348.827 +2.5%	436.261 +2.9%	523.088 +3.3%	610.979 +3.7%	699.326 +4.1%	785.088 +4.5%	876.78 +4.9%	966.04 +5.3%	1056.21 +5.7%
F#2	91.9477 +1.3%	184.156 +1.7%	276.623 +2.1%	368.831 +2.5%	461.298 +2.9%	554.025 +3.3%	647.012 +3.7%	740.518 +4.1%	834.284 +4.5%	928.57 +4.9%	1022.86 +5.3%	1117.66 +5.7%
G2	97.5987 +1.4%	195.746 +1.8%	293.482 +2.2%	391.629 +2.6%	489.227 +3.0%	587.511 +3.4%	686.481 +3.8%	785.862 +4.2%	885.379 +4.6%	985.171 +5.0%	1085.51 +5.4%	1186.26 +5.8%
G#2	103.234 +1.4%	207.236 +1.8%	311.046 +2.2%	414.856 +2.6%	518.283 +3.0%	622.86 +3.4%	727.821 +3.8%	832.207 +4.2%	938.128 +4.6%	1043.66 +5.0%	1150.16 +5.4%	1257.04 +5.8%
A2	109.361 +1.5%	219.307 +1.9%	329.643 +2.3%	439.59 +2.7%	549.928 +3.1%	659.677 +3.5%	770.403 +3.9%	882.298 +4.3%	993.024 +4.7%	1104.92 +5.1%	1218.18 +5.5%	1330.47 +5.9%
A#2	115.706 +1.5%	232.675 +1.9%	348.855 +2.3%	465.351 +2.7%	582.478 +3.1%	698.658 +3.5%	816.258 +3.9%	933.859 +4.3%	1051.93 +4.7%	1170.95 +5.1%	1289.19 +5.5%	1409.63 +5.9%
B2	123.074 +1.6%	246.49 +1.9%	369.563 +2.3%	493.321 +2.7%	617.078 +3.1%	741.007 +3.5%	865.448 +3.9%	989.89 +4.3%	1115.19 +4.7%	1240.99 +5.1%	1367.32 +5.5%	1493.81 +5.9%
C3	130.589 +1.6%	261.693 +2.0%	392.692 +2.4%	523.081 +2.8%	654.384 +3.2%	785.992 +3.6%	917.6 +4.0%	1050.43 +4.4%	1182.64 +4.8%	1316.08 +5.2%	1449.82 +5.6%	1583.78 +6.0%
C#3	137.462 +1.6%	276.483 +2.0%	414.88 +2.4%	553.278 +2.8%	691.363 +3.2%	830.072 +3.6%	969.405 +4.0%	1109.67 +4.4%	1249.32 +4.8%	1389.9 +5.2%	1531.1 +5.6%	1672.92 +6.0%
D3	146.502 +1.7%	293.308 +2.1%	440.42 +2.5%	587.531 +2.9%	734.642 +3.3%	881.753 +3.7%	1029.17 +4.1%	1176.28 +4.5%	1324.3 +4.9%	1472.94 +5.3%	1621.47 +5.7%	1770.51 +6.1%
D#3	155.143 +1.7%	310.646 +2.1%	466.039 +2.5%	621.57 +2.9%	777.239 +3.3%	932.908 +3.7%	1088.85 +4.1%	1245.08 +4.5%	1401.58 +4.9%	1558.49 +5.3%	1714.99 +5.7%	1873.02 +6.1%
E3	164.035 +1.8%	329.759 +2.2%	493.475 +2.6%	658.528 +3.0%	823.244 +3.4%	988.297 +3.8%	1153.69 +4.2%	1319.08 +4.6%	1484.47 +5.0%	1650.53 +5.4%	1811.54 +5.8%	1985.02 +6.2%
F3	173.983 +1.8%	348.892 +2.2%	523.8 +2.6%	699.71 +3.0%	874.542 +3.4%	1050.38 +3.8%	1227.6 +4.2%	1404.82 +4.6%	1583.43 +5.0%	1762.5 +5.4%	1942.2 +5.8%	2124.35 +6.2%
F#3	183.896 +1.9%	369.662 +2.3%	554.181 +2.7%	739.323 +3.1%	925.713 +3.5%	1112.1 +3.9%	1299.11 +4.3%	1486.75 +4.7%	1675.63 +5.1%	1865.76 +5.5%	2057.14 +5.9%	2249.14 +6.3%
G3	195.775 +1.9%	391.55 +2.7%	587.727 +3.1%	784.308 +3.5%	981.694 +3.9%	1179.48 +4.3%	1378.08 +4.7%	1577.48 +5.1%	1778.09 +5.5%	1979.9 +5.9%	2183.33 +6.3%	2387.57 +6.7%
G#3	206.993 +2.0%	413.985 +2.8%	621.475 +3.2%	829.643 +3.6%	1038.94 +4.0%	1247.43 +4.4%	1457.9 +4.8%	1669.38 +5.2%	1881.34 +5.6%	2094.8 +6.0%	2311.25 +6.4%	2527.2 +6.8%
A3	219.729 +2.0%	439.795 +3.2%	660.198 +3.6%	880.602 +4.0%	1102.69 +4.4%	1325.12 +4.8%	1548.56 +5.2%	1773.35 +5.6%	1999.16 +6.0%	2226.65 +6.4%	2456.17 +6.8%	2687.37 +7.2%
A#3	232.696 +2.0%	465.911 +3.6%	699.127 +4.0%	933.122 +4.4%	1167.38 +4.8%	1403.71 +5.2%	1640.57 +5.6%	1878.72 +6.0%	2119.48 +6.4%	2360.75 +6.8%	2604.63 +7.2%	2850.99 +7.6%
B3	246.266 +2.1%	493.394 +3.9%	740.951 +4.3%	988.94 +4.7%	1237.79 +5.1%	1487.93 +5.5%	1738.5 +5.9%	1992.52 +6.3%	2246.1 +6.7%	2503.13 +7.1%	2761.88 +7.5%	3023.22 +7.9%
C4	261.469 +2.1%	523.505 +4.3%	786.111 +4.7%	1049.28 +5.1%	1313.6 +5.5%	1579.04 +5.9%	1846.19 +6.3%	2115.05 +6.7%	2386.18 +7.1%	2660.16 +7.5%	2935.84 +7.9%	3213.79 +8.3%
C#4	276.399 +2.2%	553.749 +4.6%	831.104 +5.0%	1109.42 +5.4%	1389.65 +5.8%	1670.85 +6.2%	1953.96 +6.6%	2238.99 +7.0%	2528.82 +7.4%	2816.73 +7.8%	3110.4 +8.2%	3405.99 +8.6%
D4	293.461 +2.2%	587.402 +4.6%	881.823 +5.0%	1177.21 +5.4%	1474.03 +5.8%	1773.25 +6.2%	2073.92 +6.6%	2376.5 +7.0%	2683.89 +7.4%	2992.24 +7.8%	3304.92 +8.2%	3619.03 +8.6%
D#4	310.756 +2.3%	621.512 +4.9%	933.128 +5.3%	1245.17 +5.7%	1559.37 +6.1%	1875.71 +6.5%	2193.35 +6.9%	2515.28 +7.3%	2841.94 +7.7%	3168.6 +8.1%	3500.84 +8.5%	3834.38 +8.9%
E4	329.696 +2.3%	656.116 +5.3%	989.137 +5.7%	1321.47 +6.1%	1655.65 +6.5%	1991.21 +6.9%	2329.09 +7.3%	2671.13 +7.7%	3016.87 +8.1%	3364.91 +8.5%	3718.51 +8.9%	4077.18 +9.3%
F4	348.817 +2.4%	698.29 +5.7%	1048.42 +6.1%	1400.52 +6.5%	1754.92 +6.9%	2110.97 +7.3%	2470.62 +7.7%	2832.58 +8.1%	3203.07 +8.5%	3571.92 +8.9%	3947.34 +9.3%	4330.65 +9.7%
F#4	369.811 +2.5%	739.622 +6.1%	1110.68 +6.5%	1483.61 +6.9%	1859.03 +7.3%	2237.58 +7.7%	2617.37 +8.1%	3002.77 +8.5%	3396.28 +8.9%	3786.67 +9.3%	4187.04 +9.7%	4590.52 +10.1%
G4	391.691 +2.5%	784.275 +6.5%	1178.05 +6.9%	1574.2 +7.3%	1972.14 +7.7%	2373.64 +8.1%	2779.31 +8.5%	3187.66 +8.9%	3606.71 +9.3%	4026.36 +9.7%	4454.64 +10.1%	4885.58 +10.5%
G#4	414.798 +2.6%	829.597 +6.9%	1246.18 +7.3%	1664.53 +7.7%	2086.45 +8.1%	2511.93 +8.5%	2942.75 +8.9%	3371.79 +9.3%	3822.2 +9.7%	4267.26 +10.1%	4719.44 +10.5%	5178.75 +10.9%
A4	440.74 +2.6%	881.137 +7.3%	1324.55 +7.7%	1769.1 +8.1%	2219.33 +8.5%	2671.83 +8.9%	3131.16 +9.3%	3595.04 +9.7%	4069.15 +10.1%	4547.8 +10.5%	5035.56 +10.9%	5525.58 +11.3%
A#4	465.705 +2.7%	932.19 +7.7%	1401.01 +8.1%	1872.18 +8.5%	2348.03 +8.9%	2828.55 +9.3%	3316.88 +9.7%	3803.65 +10.1%	4312.26 +10.5%	4819.3 +10.9%	5340.39 +11.3%	5863.04 +11.7%
B4	493.727 +2.8%	987.974 +8.1%	1484.82 +8.5%	1985.31 +8.9%	2490.49 +9.3%	3000.86 +9.7%	3518.52 +10.1%	4039.82 +10.5%	4578.81 +10.9%	5122.48 +11.3%	5678.64 +11.7%	6244.69 +12.1%
C5	524.062 +2.8%	1048.12 +8.5%	1577.04 +8.9%	2108.74 +9.3%	2645.3 +9.7%	3189.49 +10.1%	3739.93 +10.5%	4301.47 +10.9%	4872.73 +11.3%	5456.49 +11.7%	6045.1 +12.1%	6660.09 +12.5%
C#5	553.989 +2.9%	1110.06 +8.9%	1668.2 +9.3%	2231.55 +9.7%	2801.13 +10.1%	3376.94 +10.5%	3963.15 +10.9%	4556.64 +11.3%	5169.87 +11.7%	5793.5 +12.1%	6428.56 +12.5%	7072.98 +12.9%
D5	587.907 +2.9%	1176.73 +9.3%	1769.9 +9.7%	2367.86 +10.1%	2973.51 +10.5%	3588.75 +10.9%	4212.63 +11.3%	4849.95 +11.7%	5500.7 +12.1%	6164.89 +12.5%	6844.44 +12.9%	7544.15 +13.3%
D#5	621.43 +3.0%	1247.02 +9.7%	1875.39 +10.1%	2507.91 +10.5%	3151.54 +10.9%	3804.87 +11.3%	4467.92 +11.7%	5142.06 +12.1%	5838.39 +12.5%	6543.05 +12.9%	7274.06 +13.3%	8005.31 +13.7%
E5	659.347 +3.0%	1318.69 +10.1%	1986.36 +10.5%	2660.27 +10.9%	3342.5 +11.3%	4035.12 +11.7%	4738.15 +12.1%	5466.13 +12.5%	6200.36 +12.9%	6963.71 +13.3%	7735.37 +13.7%	8545.15 +14.1%
F5	698.14 +3.0%	1400.02 +10.5%	2104.39 +10.9%	2821.23 +11.3%	3545.56 +11.7%	4282.34 +12.1%	5036.58 +12.5%	5817. +12.9%	6607.4 +13.3%	7418.99 +13.7%	8262.99 +14.1%	9105.31 +14.5%
F#5	739.78 +3.1%	1481.48 +10.9%	2230.84 +11.3%	2990.74 +11.7%	3762.14 +12.1%	4549.84 +12.5%	5351.9 +12.9%	6177.93 +13.3%	7025.03 +13.7%	7900.89 +14.1%	8755.66 +14.5%	9628.99 +14.9%
G5	786.154 +3.1%	1576.47 +11.3%	2373.74 +11.7%	3179.34 +12.1%	4003. +12.5%	4840.54 +12.9%	5701.7 +13.3%	6590.64 +13.7%	7505.97 +14.1%	8437.96 +14.5%	9419.96 +14.9%	10437.99 +15.3%
G#5	831.792 +3.2%	1666.36 +11.7%	2510.62 +12.1%	3367.37 +12.5%	4239.37 +12.9%	5133.54 +13.3%	6049.9 +13.7%	6996.76 +14.1%	7975.5 +14.5%	8984.74 +14.9%	10033.33 +	