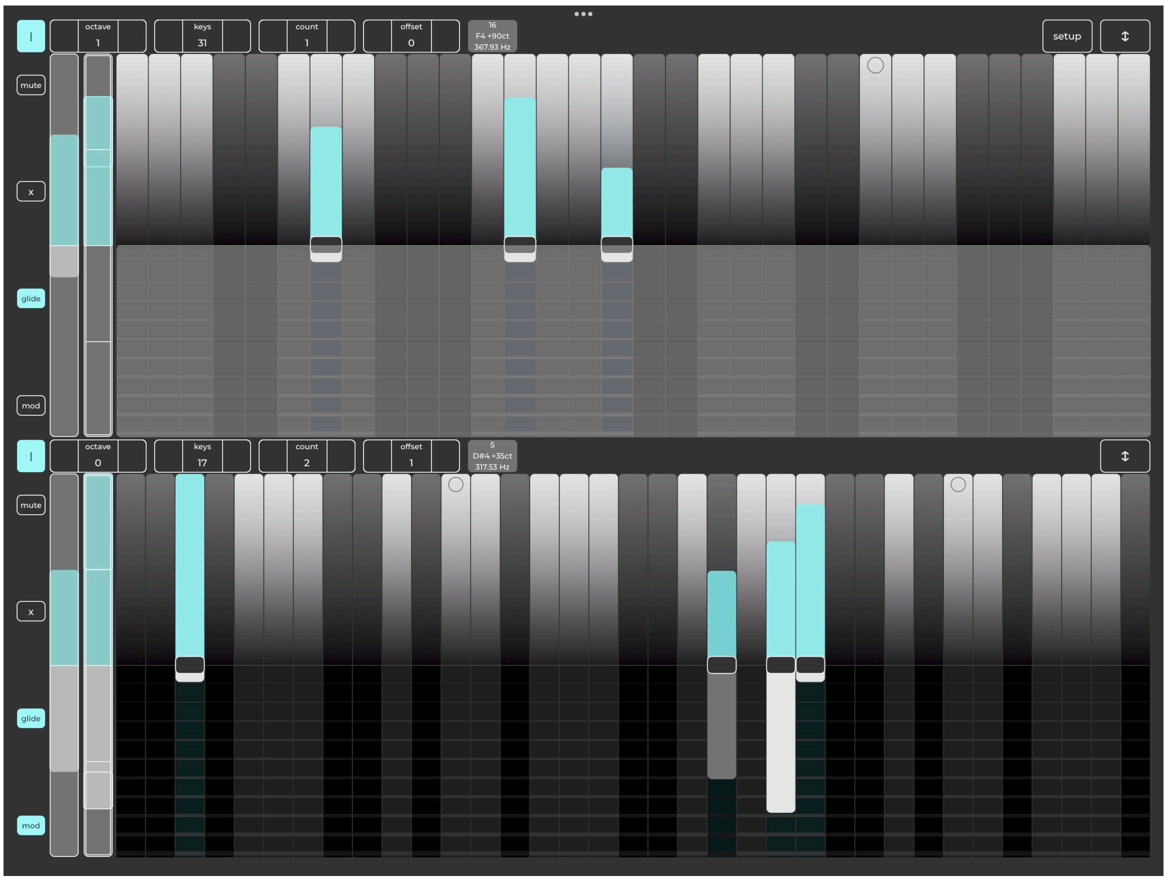
# *sinusedo* **User Guide** 12.21.22

*sinusedo* is a two-keyboard, polyphonic, sine wave synthesizer
that supports EDO (Equal Division of the Octave).

AUv3 and standalone versions are both supported on iPads and iPhones.




**Alex Nadzharov**
http://alexnadzharov.com/apps/sinusedo/

Support: weirdly.named.music.apps@gmail.com

Manual: **David S. Collett**
syntonicomma@gmail.com

# Contents

# Key features

- two independent keyboards
- 4 – 72 keys (notes, voices) *per octave (4-EDO up to 72-EDO)*
- 13-octave range for all EDO (from 51 to 935 separate notes)
- full frequency range from approximately 8 – 34000 Hz for all EDO
- dedicated ring modulator for every note
- maximum polyphony of 144 notes
- 1, 2, 3, or 4 octaves visible at one time
- display with note number (from C), note name + deviation from 12-TET (in cents), frequency (in Hz)
- two sections per keyboard: *upper* for non-sustained notes, gliding, and velocity<br>*lower* for sustained notes and modulation rate
- mute button to stop/start all sound
- panic/delete button to erase all notes
- glide button for smooth sweeps across keyboard frequency range
- modulation on/off button
- slider to set initial velocity and modulation levels
- slider to change velocity and modulation rates for all notes on the keyboard proportionally and to show current velocity and mod rate levels
- button to toggle keyboard visibility
- setup menu with audio out assignment, upper and lower keyboard MIDI input assignment, channel selection, buffer size (64 – 1024), link to this manual
- standalone and AUv3

# What you *need* to know, and what you *don't* *(but may want to)*

If you want to make the developer and author incredibly happy, and if you want to thoroughly understand this program and get the most out of it, you should read and understand at least most of this manual.[1]

The first section describes the interface and all the parameters. Although the interface in *sinusedo* is easy to use and understand, this section explains in more detail how everything works.

The second half of this guide is written for those who would like a deeper understanding about EDO tunings, frequencies, and their interactions. Various topics, some easy computations, footnotes, examples, and nine "experiments" will help you to better hear and appreciate the fascinating sounds of these microtonal tunings and ways that you may want to use them. *sinusedo* is useful not only as a way to produce interesting and complex sine waves with EDO tunings and modulations but also as a great learning tool.

If you'd like to read a primer about EDO tuning, please see the **Appendix** (page 33).

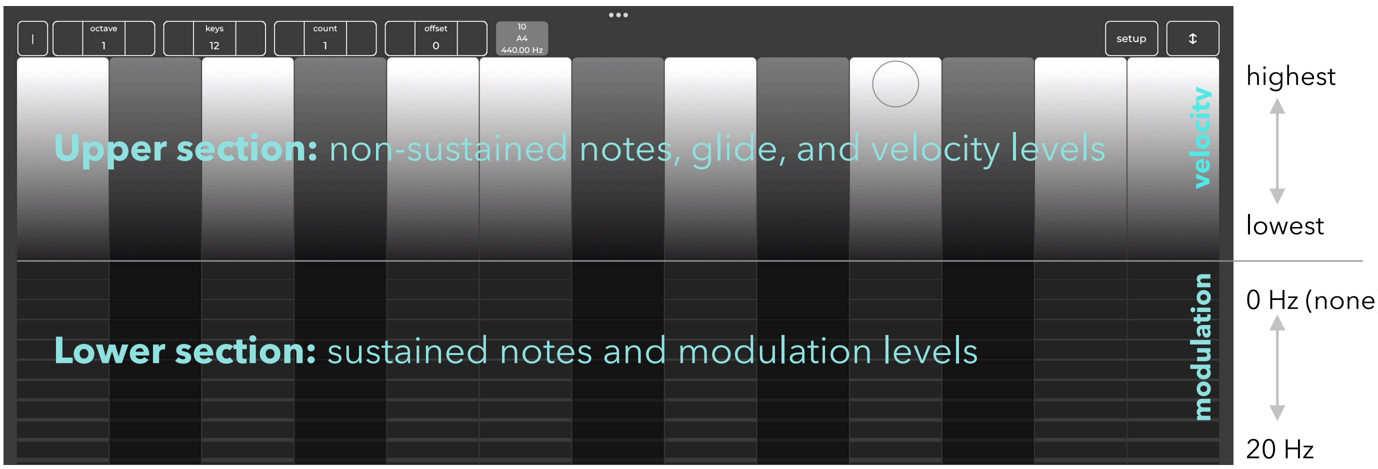
*Above all else, we hope you'll have a great time playing and experimenting with sinusedo.* ***Enjoy!***

---

[1] For those of you who want the *really* short manual (that is, the **TL;DR edition**), here it is:
Tap and glide on the keyboards, change the *keys*, experiment with modulation, have fun.

# Keyboard layout

*sinusedo* has two identical but independent keyboards. The screen layout and parameters for both keyboards function identically in the standalone and AUv3 versions. Only the setup screen is different. Each keyboard is divided into two sections. You can play the keys in either of these..




**Upper section:** Tap or drag in the *upper section* when you want *non-sustained notes*. As you drag across the keys, each one will play according to its exact calculated frequency, which is based on the number of keys you have assigned to that keyboard *(details are below).* The higher you drag, the louder the note.

**Lower section:** Tap on keys in the lower section for *sustained notes*. When a note is sustained, it will have a dark grey rectangle on the center line. Tap it to turn off the note. This lower section controls two parameters: attack time and modulation rate. The lower on a key you tap, the longer the attack time for volume.

In addition, when the **mod** button is ON *(see below),* the lower you tap increases the modulation rate for that key's LFO (from 0 – 20 Hz).

# Control bar

Across the top of the keyboard are the following buttons and parameters:


and


From left-to-right, they are the toggle for additional keyboard functions; octave number (-4 to +4); number of notes per octave (the EDO count); the number of octaves shown; offset to slide keyboard left or right; display for note number, note name, offset in cents, and frequency; setup menu; and toggle for either one or two keyboards.

Each of these functions is described below.

## Single or dual keyboards

At the top right of each keyboard is an arrow. Clicking on either one will close the other keyboard so that one keyboard will be full screen. This larger area allows more exact placement and fine-grained adjustments of note placement, volume levels, modulation levels, and attack levels.

Press the arrow again to return to the two-keyboard layout, which is ideal for performing on dual keyboards that have the same or different number of keys and for comparing the sound when using different EDOs.

## Number of keys (notes) per octave: EDO


The keys parameter allows you to specify the number of notes you want (4–72) *per octave*. To quickly advance down or up, hold down the left or right side of the box. To quickly return to 12-EDO, double tap on the number.

Each of these notes will be spaced evenly within an octave, referred to as EDO (Equal Division of the Octave), TET (Tone Equal Temperament), or simply ET[1]. Although these are initialisms and originally capitalized, it's becoming more common to use the lowercase versions: *tet, et, edo*.

Traditional keyboards are designed with 12 notes per octave. The frequency of notes doubles (or halves) every octave. So, in our standard 12-note chromatic scale (notated as 12-TET, 12-ET, or 12-EDO), the division between each of the 12 notes is equal.

*sinusedo* enables you to break this restriction of only 12 notes and create octaves containing as few as 4 or as many as 72 notes, each note separated by exactly the same division. Each of these divisions is measured in *cents.* Every octave for all EDOs have 1200 cents, equally divided among the notes.

For example, in 12-EDO, there are 100.00 cents between consecutive notes (1200 / 12).
Similarly, in 31-EDO, there are approximately 38.71 cents between each note (1200 / 31).

---

[1] Although TET, ET, and EDO are often used interchangeably with little to no confusion, there is a slight (but important) difference. By their names, TET and ET imply that the number of notes (and therefore the interval size) chosen per octave is a *temperament*; that is, it *tempers* or distributes one or more tiny intervals (called *commas*) that would otherwise be harsh. EDO, on the other hand, simply denotes that the octave has been divided equally into the number of notes specified. It doesn't imply that tempering is of importance.

n-TET or n-ET are subsets of n-EDO. For example, some equal divisions of the octave, such as 12, 19, and 31 notes are considered *meantone temperament*, so they can be referred to as 19-TET, 19-ET, or 19-EDO.

The theory of microtonal (or *xenharmonic*) tuning is enormous and technically complex. *sinusedo* offers you the world of microtonal keyboards for performing without the need for understanding temperaments. Therefore, this manual uses the term *EDO*. Certainly, if you want to study microtonal scales and temperament theory, *sinusedo* is also an excellent tool for learning and experimentation.

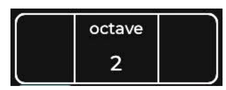
# octave, keys, count, offset

*sinusedo* offers 13 full octaves of keys for all EDO layouts (plus an extra C at the top).
For example, 72-EDO has a total of (72 x 13) – 1 = 935 notes.

So that you can play all of these notes, *sinusedo* uses a combination of four parameters:
**octave**, **keys**, **count**, **offset**.

Every octave in *sinusedo* begins on C. The parameter **octave**, ranging from -4 to +4, specifies which of 9 octaves the *leftmost key* begins. In addition, using the maximum *count* and offset *(see below)*, you can view an additional 4 octaves, bringing the total octaves covered to 13. Note that this octave number is a general indication of which octave range the notes begin on the *sinusedo* keyboard. The actual MIDI octave designation (such as C4 or G#6) is reported in the **info window** *(see page 8)*.

**keys** is the number of notes per octave. This is the EDO. For example, if you want 21-EDO, set keys = 21.
*(See the previous section for details about keys.)*




**count** is the number of octaves (1–4) displayed on the screen at one time. Each of these octaves contains the number of keys you specify, plus an extra C at the top.
For example, if you are using 17-EDO, then 18, 35, 52, or 69 keys can be displayed at one time. *sinusedo* displays all octaves on the screen, which enables you to play individual notes or run your finger across the entire range to hear glissandos of the discrete note pitches or as a continuous sweep of frequencies *(see Glide below)*. Changing the octave lowers or raises the C on which the keyboard begins.


**offset** shifts the keyboard left or right one key at a time, from 0 to $n$-1 keys (where $n$ is the number of keys you've selected). This is useful when you need to see a few more keys that are beyond the view of the screen. You can also increase the range of the keyboard by a full octave.[1]

---

[1] The offset can be set up to $keys - 1$, which will increase the number of notes by *keys*. For example, suppose we are using 17-EDO (keys = 17). When offset = 0, the total number of keys we have is (17 x 12) + 1. However, we can increase the offset up to 16 (17–1), which adds another 16 keys (an extra octave) at the top of the keyboard.

However, the upper notes of every $n$-EDO keyboard are in the frequency range above 32 kHz, far outside our hearing range. Therefore, adding notes even higher than this (which can bump the frequency above 64 kHz) has no purpose.

[A future version of *sinusedo* may apply antialiasing to notes above about 16 kHz by folding the frequency with the sample rate to produce interesting, and sometimes random, inharmonic sounds. This or other techniques would create a musical use for all the infrasound and ultrasound frequencies that are otherwise outside our range of hearing.]

This table shows the octave numbers and the note ranges that the keyboard displays.[1]

Octaves –4 to 3 assume that count = 1. If count > 1, the upper range will expand by 1, 2, or 3 octaves, and setting offset to maximum will add yet one more octave.

| octave # | note names | | | frequency range (in Hz)[1] | | | |
|---|---|---|---|---|---|---|---|
| $4_{\text{Count4 + Offset } n\text{-}1}$ | ~C7 | to | C12 | 4145.90 | to | 66334.46 | |
| $4_{\text{Count4}}$ | C7 | to | C11 | 2093.00 | to | 33488.07 | |
| $4_{\text{Count3}}$ | C7 | to | C10 | 2093.00 | to | 16744.04 | |
| $4_{\text{Count2}}$ | C7 | to | C9 | 2093.00 | to | 8372.02 | |
| $4_{\text{Count1}}$ | C7 | to | C8 | 2093.00 | to | 4186.01 | |
| 3 | C6 | to | C7 | 1046.50 | to | 2093.00 | |
| 2 | C5 | to | C6 | 523.25 | to | 1046.50 | |
| 1 | C4 | to | C5 | 261.63 | to | 523.25 | *contains Middle C4 and A4 (440 Hz)* |
| 0 | C3 | to | C4 | 130.81 | to | 261.63 | |
| -1 | C2 | to | C3 | 65.41 | to | 130.81 | |
| -2 | C1 | to | C2 | 32.70 | to | 65.41 | |
| -3 | C0 | to | C1 | 16.35 | to | 32.70 | |
| -4 | C-1 | to | C0 | 8.18 | to | 16.35 | |

The frequency range of an 88-note piano in traditional 12-TET is from 27.5 Hz (A0) to 4186.0 Hz (C8). So octave -3 to octave $4_{\text{Count1 + Offset 0}}$ covers this entire range.

Every octave, regardless of the number of keys (EDO), will contain an exact tuning for A, which will be an even multiple of 440.00 Hz.

Notice that every A on the keyboard has a small circle at the top of the key. *sinusedo* uses A4 440.00 Hz to compute the frequency for every note of every $n$-EDO. These circles are an excellent navigation aid when you are playing notes or gliding across the keys.

Also note that because all keys are computed based on A 440.00 Hz, no other keys will have the same frequency as they do in 12-TET, except for every other A.[2]

---

[1] These frequencies are for 72-EDO *(keys = 72)*. Every EDO will produce slightly different frequencies for each of these ranges.

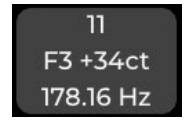
For all octave numbers, when offset = 0, the range can be easily computed by:

$$C_{octave+3} \text{ to } C_{octave+3+count}$$

For example, if octave = 2 and count = 4, then the keyboard will display $C_{2+3}$ to $C_{2+3+4}$ = $C_5$ to $C_9$

[2] Or when keys is an even multiple of 12.

# Info window




This window shows the **key number**, **note name**, **offset (in cents)**, and **frequency** of the last note you pressed.

The number on the first line is the **key number** within the current EDO octave.
The keys in *sinusedo* are always numbered from C to B, starting with key 1. For example, if we set keys=23 (for 23-EDO), then every C is note 1, the next note is 2, then 3, ..., and the last note before the next EDO octave begins (on C) will be 23. In this example, we are on the 11th note of our 23-EDO scale.

The next line displays the closest **note name** (in our traditional notation of C, C#, D, D#, ..., A#, B). That is, *sinusedo* assigns a note name that is *as close as possible* to the equivalent note name we commonly use.

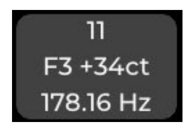
If the frequency of this note (which is always based on A 440.00 Hz) is exactly equal to one of the notes in 12-TET, then only the note name appears. But, in most cases, the note will be flatter (–) or sharper (+) than the nearest note in 12-TET by a certain number of cents.[1]

In this example using 23-EDO, the note pressed is ~178.16 Hz, which is a bit sharper than the F3 in 12-TET, by a difference of approx. +34 cents.[1]

---

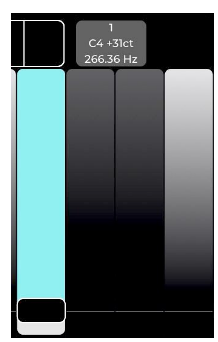
[1] In all tuning systems, we divide an octave into small units called cents, and there are 1200 cents per octave. For example, in 12-TET (our traditional equal temperament scale with 12 notes per octave), the distance from one note to the next note (one semitone up or down) is 100 cents (1200 / 12 = 100). In 31-EDO, there are approximately 38.71 cents between consecutive notes (1200 / 31).

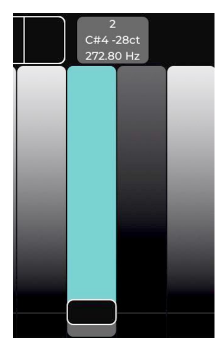
In 12-TET, F3 is ~174.61 Hz. The difference between the frequency of the note we hear on our 23-EDO keyboard and that of 12-TET is ~3.55 Hz (178.16 – 174.61). For a 23-EDO keyboard, this represents a difference of about ~34 cents .



*23-EDO keyboard*

(Note: The final section of this manual, *Making sense (and cents) of frequencies in sinusedo*, explains in more detail the concepts and formulas for cents and frequencies.)




If the difference between frequencies of two adjacent notes is greater than 50 cents, the next note name is used with a negative cents value.

For example, the note on the left (in 29-EDO) is marked C4 **+31 cents**, so it's a bit sharper than C4 in 12-TET. The next note is 272.80 Hz, which is +72 cents sharper than C4. However, because 72 > 50, this note is designated as C#4 **-28 cents** (100 - 72 = 28).

This method allows you to play any note in any EDO and see what the nearest note is in our traditional system, and exactly how far off it is (in cents). Remember that in 12-TET, every semitone equals 100 cents.

So, for example, the note on the above left is 31 cents sharper than a C4 in 12-TET, or about ⅓ of a semitone sharper, and the note on the right is 28 cents flatter than C#4, about ¼ of a semitone flatter.

In order to create a keyboard having from 4 to 72 keys *per octave* and still represent the keyboard using our traditional keyboard layout  (7 white and 5 black notes), the keys must be split into several smaller keys. For example, the keyboard for 23-EDO has two keys for F,  two for G, only one for G#, two for A, etc. *sinusedo* maps the number of keys you requested *as evenly as possible* onto the 7 white and 5 black keys, then distributes the frequencies across them. This allows you to play a full range of EDO scales while still using a traditional keyboard layout. *(See pages 23 and 26 for more details about how the keys are distributed.)*

# Controls panel

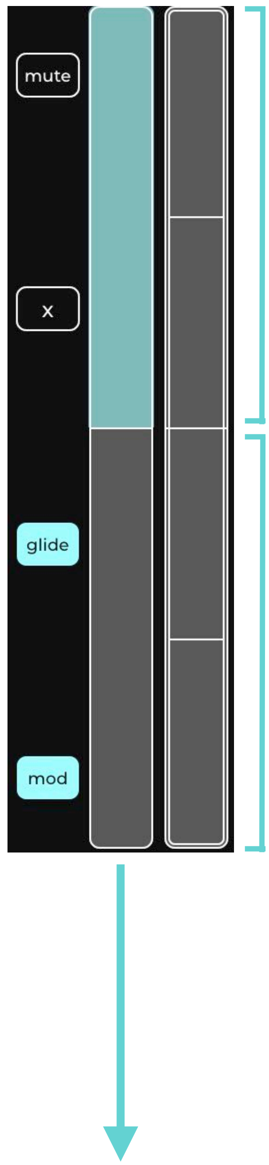
In the upper left corner of each keyboard is an icon with a vertical line.
Press this to open the controls panel, which contains the following functions:

**mute** turns off all sound for this keyboard but doesn't erase the notes. Using this, you can stop and start the sound at any time.

**x** erases all the sustained notes for this keyboard. It also functions as a panic (all MIDI off) button or stuck notes.

**glide** allows smooth sweeps across the frequency spectrum on the keyboard. Double-tap to turn on. When glide is off, sweeping across the upper section will be a glissando with the discrete notes of the scale.

**mod** toggles modulation. When mod is on, dragging the bar downward increases the ring modulator rate from 0 to 20Hz. Using this, you can stop and start modulation at any time.




**global volume adjustment**: After you have entered one or more sustained notes, you can slide down/up in this area to change the volume of all notes proportionally on this keyboard.

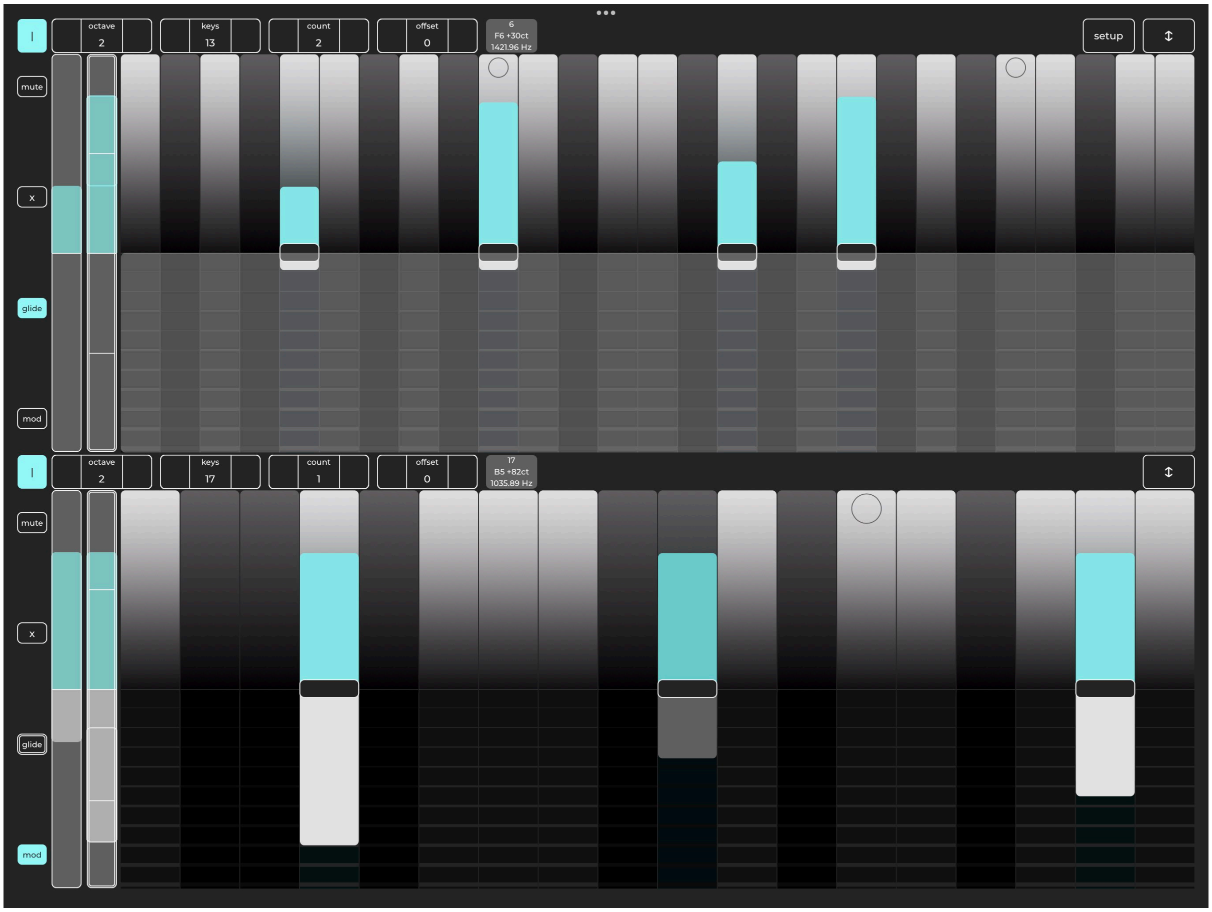
*Tip:* If you find that you can't drag a note's volume all the way to the top or bottom, double-tap the slider to return the white line to the midpoint.

**global modulation adjustment**: After you have entered one or more sustained notes and increased their modulation levels (by dragging down from the note's dark gray rectangle), you can slide down/up in this area to change the levels of all notes proportionally on this keyboard.

**initial velocity and modulation rate levels**:
Before you add any sustained notes, you can drag the upper slider to set an initial velocity. Now when you turn on any sustained notes, they will start at this velocity level.

Similarly, you can drag the lower slider to set an initial level for the modulation rate – the lower you drag, the faster the modulation will be. Now if **mod** is on, when you turn on any sustained notes, they will start with this level of modulation.

# Putting it all together




## Upper keyboard:

- **keys** = 13, so we are using 13-EDO = 13 notes per octave. 1200/13 ~ 92.31 cents between each note.
- **octave** = 2, **count** = 2 (two octaves visible): notes C5 to C7. Notice the two circles at A5 and A6.
- **offset** = 0, so leftmost note is C.
- **initial velocity** ~25%
- **mod** = off, so initial mode rate slider cannot be set. Notice that all notes' mod levels are off.
- **glide** = on, so we will hear a smooth sweep across all the frequencies as we drag our finger(s).

## Lower keyboard:

- **keys** = 17, so we are using 17-EDO = 17 notes per octave. 1200/17 ~ 70.59 cents between each note.
- **octave** = 2, **count** = 1 (one octave visible): notes C5 to C6. Notice the circle at A5.
- **offset** = 0, so leftmost note is C.
- **initial velocity** ~75%
- **mod** = on and we adjusted each of our 3 notes to have a different modulation rate (from 0–20 Hz).
- **glide** = off, so we will hear a glissando with discrete notes as we drag our finger(s).

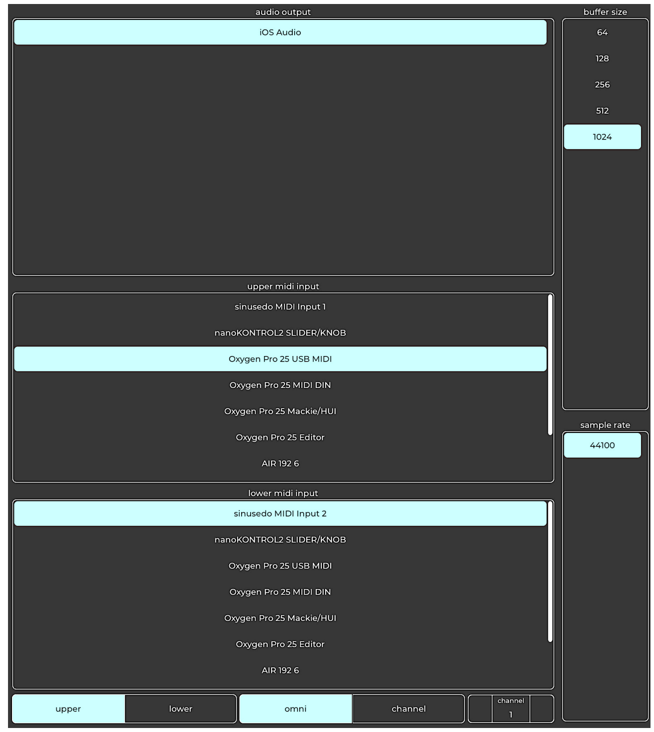
---

[1] Note that the high C on every octave is labeled incorrectly. The octave number should be increased by one. For example, in the lower keyboard example above, the highest C is written as C5. It is actually C6. *This will be corrected in an update.* Also note that some keyboards start and end with C# instead of C. *This will also be corrected in an update.*

# Setup menu

The **setup** screen for the *standalone version* contains:

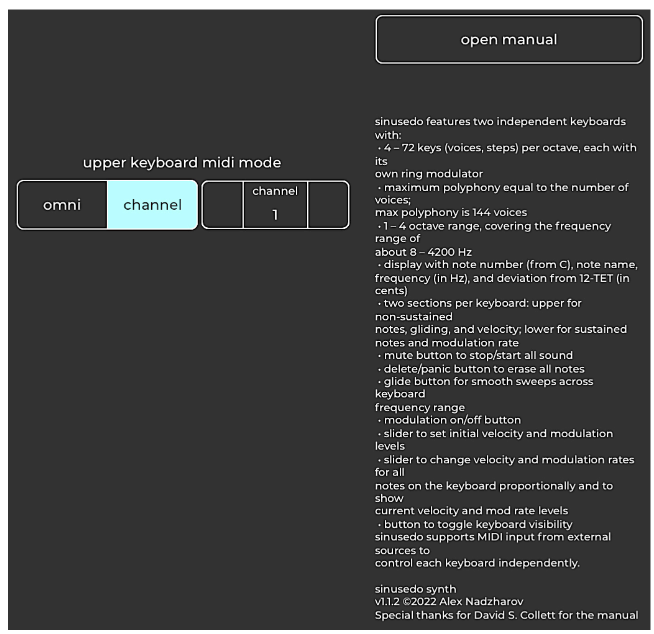
- audio output
- MIDI input source for the upper and lower keyboards
- channel numbers for the upper and lower keyboards (or OMNI [1])
- buffer size (64–1024)
- sample rate (44.1–192 kHz)
- contact information
- *sinusedo* feature list
- link to this manual




The **setup** screen for the *AUv3 version* contains:

- channel numbers for the upper keyboard only (or OMNI [1])
- contact information
- *sinusedo* feature list
- link to this manual

The audio output, MIDI input, buffer size, and sample rate are set within the AUv3 host.




---

[1] OMNI mode can be used to play notes and chords on a standard external MIDI keyboard and have them interpreted in your chosen n-EDO tuning. See *Experiment 8* for information.

# Making sense (and cents) of frequencies in *sinusedo*

The information below isn't about *how* to use *sinusedo*. That was covered above.

Instead, this section explains how two or more frequencies interact. Having a better understanding of frequencies, cents, beats, just noticeable difference, and other aspects will help you to better appreciate the amazing interaction of sound waves and get the most out of *sinusedo*.

Acoustics, tuning systems, and the psychophysiology of sound are all extremely complex subjects. But, for most of us, the information below will be enough to give you a good understanding of frequencies.

*sinusedo* is not only an excellent app for producing notes, chords, arpeggios, and sweeps in different EDOs, but also an extremely useful way to learn more about waveforms and their interactions.

## A few formulas *(for those of you who like math)*

[1] $r = \sqrt[k]{2} \; = \; 2^{\frac{1}{k}}$

[2] $f_2 = f_1 r \; = \; f_1$

[3] $f_n = f_1 r_k^n \; = \; f_1 \sqrt[n/k]{2} = f_1 2^{\frac{n}{k}}$

[4] $c_{i,i+1} = 1200/k$

[5] $C_{f_2 f_1} = |1200 \times log_2(f_2/f_1)|$

[6] $r_k^n = \sqrt[n/k]{2} = 2^{\frac{n}{k}}$

[7] $B_{f_2 f_1} = |f_2 - f_1|$

[8] $DT_{f_2 f_1} = (f_2 + f_1)/2$

For 12-tet, these formulas are useful:

[$a$] Piano note $=$ MIDI note $- 20$

where piano notes are 1–88, MIDI notes are 21–108

with frequencies from 27.500 – 4186.000 Hz

[$b$] $f_N = \sqrt[12]{2}^{(N-49)} 440 \; = \; 2^{\frac{N-49}{12}} 440$

[$c$] $f_M = \sqrt[12]{2}^{(M-69)} 440 \; = \; 2^{\frac{M-69}{12}} 440$

[$d$] $N_f = \lfloor 12 \times log_2(f/440) + 49 + 0.5 \rfloor$

[$e$] $M_f = \lfloor 12 \times log_2(f/440) + 69 + 0.5 \rfloor$

Approximations:

[$f$] $JND \approx 0.005 f$ (in cents)

[$g$] Upper Freq $\approx 20925 - (age \times 166)$

### Agh! Do I really need these?

***Short answer:*** No.

***Better answer:*** In order to calculate frequencies, cents, and beats, these basic formulas are necessary.

*However*, if you want to avoid the math, the following discussion and hands-on experiments with *sinusedo* will still give you a high-level understanding and appreciation of frequency interactions.

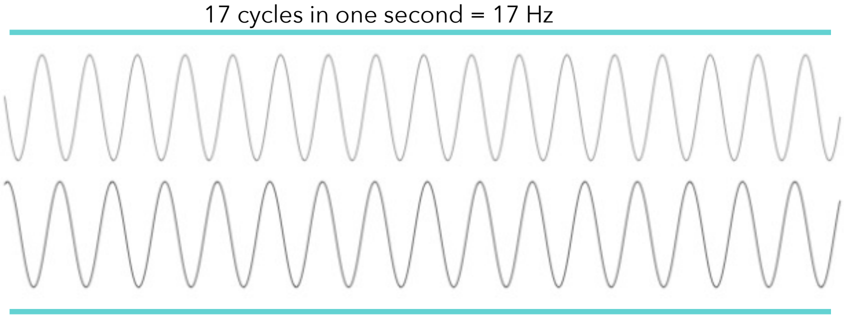
r = ratio between adjacent notes
k = number of steps (notes/keys) per octave
$f_1$ and $f_2$ = frequencies of two simultaneously played notes
$f_N$ and $f_M$ = frequencies of piano note N or MIDI note M
$N_f$ and $M_f$ = closest piano note # and MIDI note #
c, C = cents
B = beats
DT = difference tone
JND = just noticeable difference
Upper Freq = approx highest frequency we can hear based on age

## Measuring frequencies

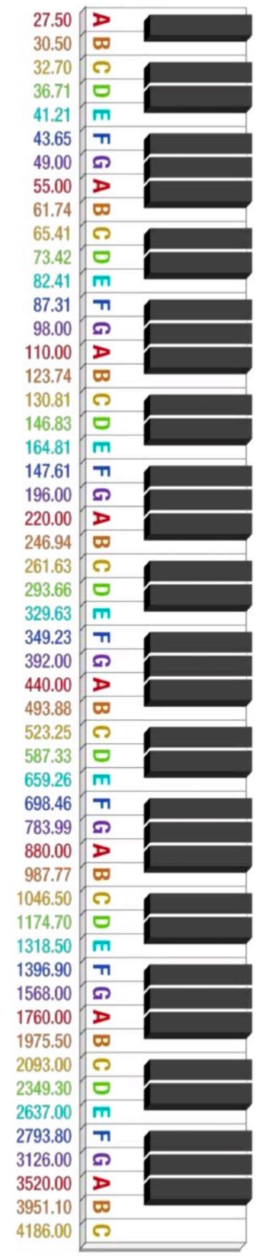
We measure the pitch of a sound (how high or low) using hertz (Hz), which is simply the number of cycles the wave completes in one second.

Suppose the horizontal aqua lines below represent one second. The top sine wave completes 17 cycles in one second (17 Hz), whereas the bottom wave completes 15 cycles (15 Hz), so the frequency (pitch) of the bottom wave is lower.

17 cycles in one second = 17 Hz



15 cycles in one second = 15 Hz




## Frequency range of human hearing

The theoretical frequency range that humans can hear is from about 20 – 20,000 Hz for "perfect" (and young!) ears. However, most sounds above about 10,000 Hz are uncomfortable, and the majority of daily sounds ranges from about 250 to 6000 Hz. For comparison, the range of a standard 88-note piano is from 27.50 to 4186.00 Hz.

As we grow older, our ability to hear high frequencies diminishes. The formula [ *g* ] above gives a rough estimate of the highest frequency we can hear based on our age. This is assuming, of course, that we have no other ear damage. For example, a person who is 16 years old will have an upper limit in the range of 18,000 Hz, whereas the upper range of a person who is 60 is reduced to about 11,000 Hz.

**Experiment 1**: In *sinusedo*, open a single keyboard, set octave=**1**, keys=**12**, count=**1**, mod=**off**. The lowest (left) note is C4 (middle C). To the right, you'll see that one of the keys has a circle at the top, indicating that it's an A (A4 in octave 1). Turn the note on by pressing anywhere in the lower section.

In the small window at the top, you'll see that this is key 10 (ten notes above C), A4, and 440.00 Hz, which simply means that the sine wave completes 440 cycles every second.

Now add the adjacent higher note, which is note 11 (from C), A#4, at 466.16 Hz. Note that *sinusedo* rounds off the exact frequencies *(see below for exact calculations)*.

Listen to the sound. You should hear two important concepts: extremely fast "fluttering" **beats** and a newly created **difference tone** *(see below)*.

## Difference tone

When two tones that are at least 20 Hz apart are played simultaneously (at equal loudness intensities of at least 50dB), the two tones merge to create a new tone that is exactly the average. See formula [ 8 ] above.

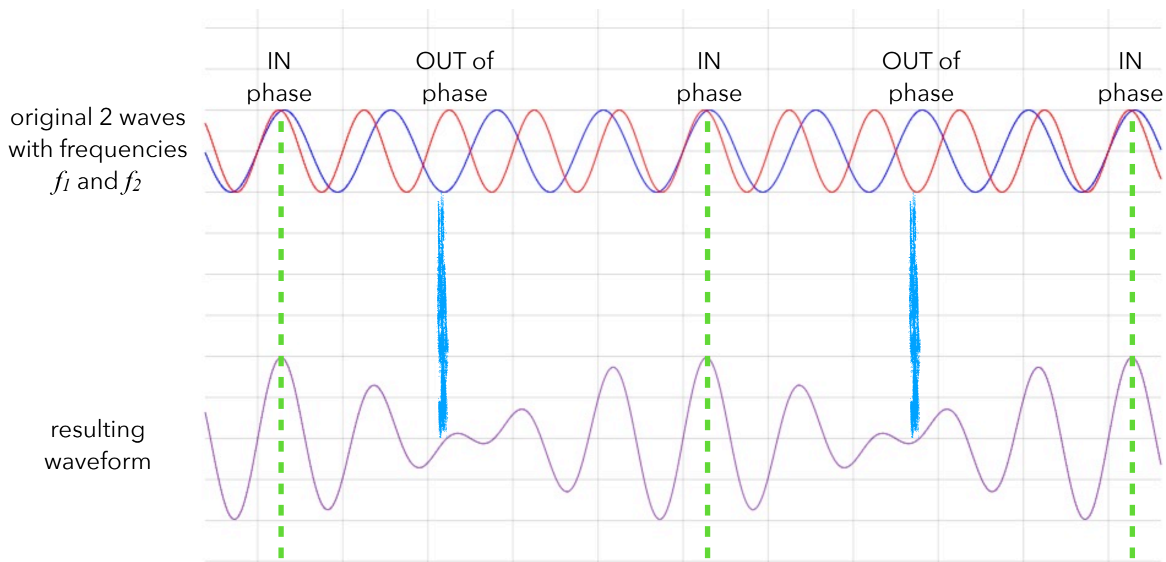
For our example above, $f_1$ = 400.00 Hz *(400 cycles per second)* and $f_2$ = 466.16 Hz,
so the frequency that we actually hear is (400.00 + 466.16) / 2 ~ 433.08 Hz

If the two notes have different volumes (loudness intensities), the difference tone becomes skewed toward the one with the higher volume.

*What if the two tones were closer than 20Hz?*

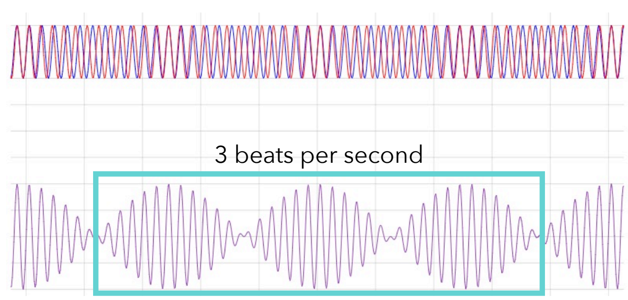
## Beats

When two notes with different frequencies are played simultaneously, their waves fall in and out of phase:




When the difference in frequency between the two sounds $f_1$ and $f_2$ is small (under about 20 Hz), then we hear a "beating" effect caused by the waves falling in and out of phase as shown above. We hear only one pitch but with the volume increasing and decreasing; that is, we hear a *tremolo*. Because the difference of the frequencies is small, we don't hear the difference tone. The smaller the frequency difference, the more noticeable the beats.

When we increase the time so that we can see more waves, the beating becomes more apparent:



**How many beats will we hear?** Using formula [ 7 ] above, the number of beats per second is simply the difference in frequencies of the two notes.

> **Experiment 2**: In *sinusedo*, open a single keyboard, press the 'x' button if any notes are playing, then set octave=**0**, keys=**72**, count=**1**, mod=**off**.
>
> Turn on the A3 (marked with a small circle at the top of the key). This is the 55th key above C and has a frequency of exactly 220.00 Hz. (Note: All of the A notes in *sinusedo* are tuned to exact multiples of 440.00Hz, and all other notes for all EDO are calculated from these frequencies.)
>
> Now turn on the adjacent note. This is labeled A3+16ct *(see below)* with a frequency of 222.13 Hz.
>
> You should hear the noticeable beats (the pulsing). These are caused by the two waves interacting and interfering with the other wave as shown on the previous page. The exact number of beats you hear is simply the absolute difference between the two frequencies, formula [ 7 ]:
>
> $$B = | 220.00 - 222.13 | = 2.13 \text{ Hz} \quad \text{(about 2 beats per second)}$$
>
> Now turn off the A3+16ct note (the second note you added), and then turn on the note to its immediate right, which is A3+33ct at 224.28. Now the number of beats you hear is approximately 4 beats per second.
>
> Continue doing this, each time moving the note higher. The noticeable beats become a very fast flutter, and somewhere between about 240 to 250 Hz (which is 20–30 Hz above our original note), the beating (and fluttering) will have disappeared, and you will now hear a second tone. This difference in frequency is called the JND.

## Just Noticeable Difference (JND)

JND is the difference in two frequencies at which we can distinguish that there are two notes rather than a single pitch with beating.

This isn't an exact number because it depends on the loudness (in decibels), whether or not the waveform is pure (such as sine) or complex (such as an instrument or voice), the frequency range of the notes, and the acuity of the ears. Also, the JND is smaller if both tones are played simultaneously rather than one after the other, primarily because we can hear the beating.

From about 100–1000 Hz, the JND is 1-2 Hz. This means that, for example, if we play two notes *(one after the other, not simultaneously)* under 1000 Hz separated by only 1 or 2 Hz (such as 400 and 401 Hz or 402 Hz), we can start to hear a second tone. We can still hear the beating of 1 or 2 beats per second if they are played simultaneously, but we can also hear a slight change of pitch.

From 1000–4000 Hz, the JND is approximately 0.5% (0.005) times the frequency, formula [ $f$ ] above. This is approximately 10 Hz. For example, if we play two notes consecutively, one at 2000 Hz and the next one at 2003 Hz, we cannot tell that this is two pitches. But if we play two notes at 2000 and 2010 Hz (or higher), we should be able to hear these as different frequencies, even though we still hear fast beating.

**Experiment 3**: In *sinusedo*, open **two** keyboards, press the 'x' buttons if any notes are playing.
On the upper keyboard, set octave=**3**, keys=**72**, count=**1**, mod=**off**.
On the lower keyboard, set octave=**3**, keys=**66**, count=**1**, mod=**off**.

On the upper keyboard, turn on G#6+66ct, 1726.44 Hz (this is two notes below A6).
On the lower keyboard, turn on G#6+63ct, 1723.42 Hz (also two notes below A6).

Try turning on only one of them, turn it off, then turn the other one on. The two pitches sound identical. When they are both on, you hear approximately 3 beats per second (1726.44 – 1723.42), formula [ 7 ].

There's only a 3 Hz difference between the notes, but because the JND for this range is about 10 Hz, our ears can't interpret that this is a different note.

Keep the note on the upper keyboard, and on the lower keyboard, turn on G#6+45ct, 1705.41 Hz (three notes below A6). Most people can hear that this pitch is slightly lower than the first one. Try playing them together, then try playing them individually to compare. This time, the difference in pitch is 21Hz, so we can hear 2 notes.

Try setting one keyboard to 72 keys and the other to 71 keys. Experiment playing notes in different ranges of the keyboard that are close in frequencies. Play the keys individually, and play them together. For different ranges on the keyboard, what is the difference in Hz for you to hear two separate frequencies?

## Making sense of cents

In order to accurately measure small changes in frequency between two notes, the octave is divided into 1200 cents (divisions). So, whether you are using 12-TET, 23-EDO, or 72-EDO, there are always 1200 cents per octave.

To calculate the number of cents between consecutive notes for any EDO with $k$ keys, use formula [ 4 ]. For example, for 23-EDO, the number of cents between consecutive notes is 52.1739130435..., or approximately 52.17 cents (1200/23). For 72-EDO, there are approx. 16.67 cents between notes.

Notice that even with 72-EDO, we may be able to hear an extremely slight difference in pitch between two consecutive notes.

We can calculate the number of cents between any two frequencies using formula [ 5 ]:

Suppose our two notes have frequencies 593.78 and 664.42 Hz. The number of cents between them is: $| 1200 \times \log_2(593.78 / 664.42) | \sim 194.60$ cents. We can also easily compute approximately how many keys apart this will be in any EDO with $k$ keys by simply dividing the number of cents by the number of keys. For example, in 37-EDO, 194.60 / 37 ~ 5.3 keys apart (which will be either 5 or 6 keys apart).

In 17-EDO, 194.60 / 17 ~ 11.45 keys apart (so 11 or 12 keys).

## Calculating frequencies

For many calculations with frequencies, we need to know the ratio between them. For example, the ratio of an octave is always exactly 2:1. So if our frequency is 440.00 Hz, then an octave below is 220.00 Hz, and an octave above is 880.00 Hz.

To calculate the ratio between two consecutive notes, use formulas [ 1 ] and [ 2 ].
For example, suppose we are using 19-EDO. The ratio between each note is $\sqrt[19]{2} = 2^{\frac{1}{19}} = 1.03715504445$
So, if our first note is 220.00 Hz, then the next note is 220.00 x 1.03715504445 = 228.174109779 Hz.
And the next note would be 228.174109779 x 1.03715504445 = 236.65192897 Hz.

Obviously, this would be tedious if we wanted to know, for example, the number of cents from our note (220.00 Hz) to the 11th note above it using 19-EDO. Formula [ 3 ] makes this easy:

$$220.00 \sqrt[\frac{11}{19}]{2} = 220.00 \times 2^{\frac{11}{19}} \approx 328.63 \text{ Hz}$$

Finally, suppose you want to know the ratio between a note and another one $n$ keys higher using $k$-EDO. Formula [ 6 ] gives you that ratio. For example, for the above example of 220.00 Hz, what is the ratio of the 11th note above 220.00 Hz in 19-EDO? $r$ = 2^(11/19) = 1.49375896165

(We can verify this by multiplying 220.00 x 1.49375896165 = 328.63 Hz.)

## MIDI and piano note numbers and their frequencies in 12-TET

Formulas [ $a$ ] through [ $e$ ] will be useful if you know the MIDI note number or the piano key number and need to calculate the exact frequency.

The 88 piano keys are numbered from 1 for A0 (27.5 Hz) to 88 for C8 (4186.0).
The piano note number is the MIDI note number – 20. This is formula [ $a$ ].
(Conversely, the MIDI note number is the piano note number + 20.)

If you know the MIDI note number (M) or the piano note number (N), formulas [ $b$ ] and [ $c$ ] will tell you the exact frequency of the note.

Conversely, if you know any frequency between 27 and 4186 Hz, formulas [ $d$ ] and [ $e$ ] will tell you the closest MIDI note (M) or piano note (N) to that frequency.

For example, suppose you have a note with frequency 661.50 Hz and want to know the closest MIDI note number. Use formula [ $e$ ] :

$$M = \lfloor 12 \times log_2(661.50/440) + 69 + 0.5 \rfloor = 76$$

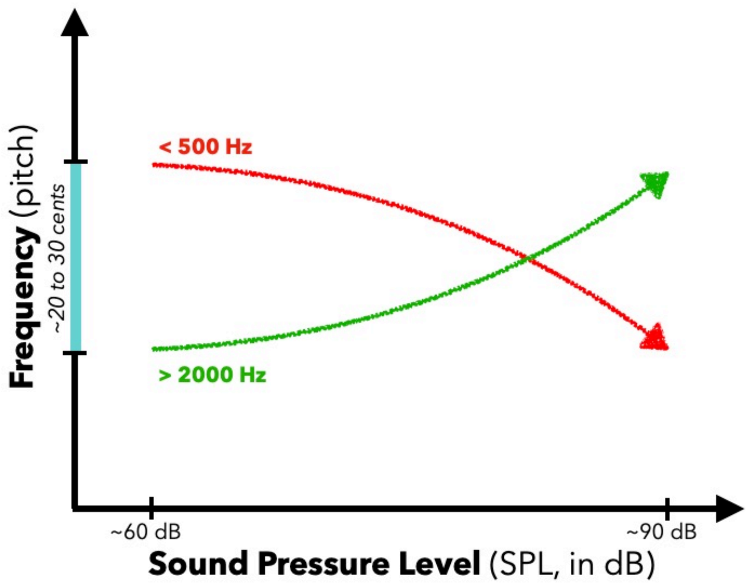
MIDI note 76 is E5 with frequency 659.26 Hz.

$\lfloor n \rfloor$ is the *floor function* and simply means the largest integer less than or equal to $n$. For example, $\lfloor 3.9324 \rfloor = 3$

## Loudness and Pitch

Loudness is a *subjective* measure of sound pressure level (SPL, measured in *dB*). The loudness (or volume, measured in *phons*) that we perceive a sound to be depends on the actual SPL, the pitch, the duration, and the complexity of the waveforms (in particular, which harmonics are most prevalent).

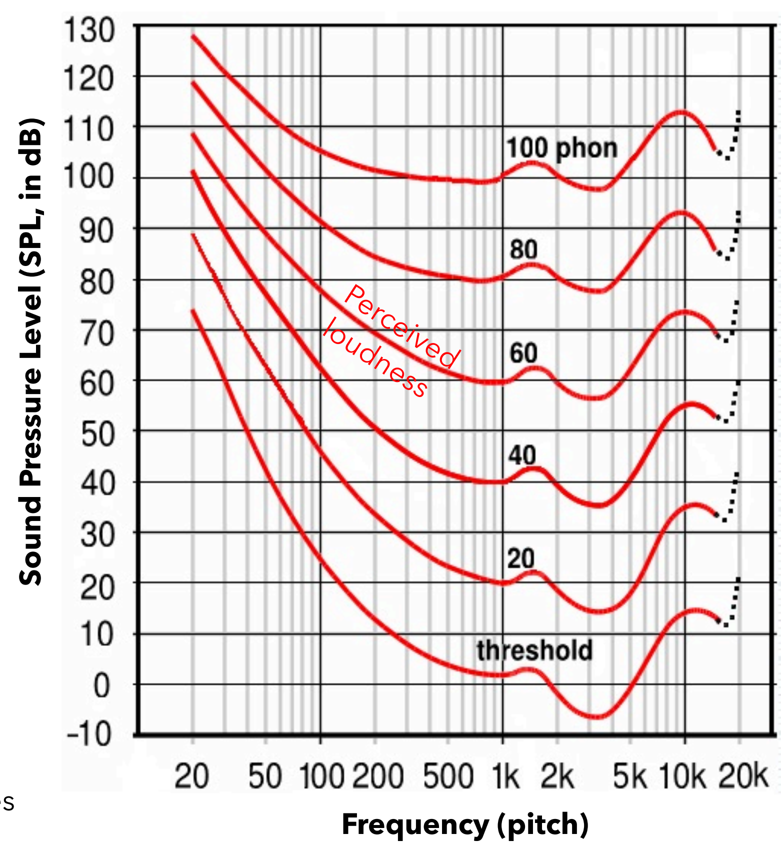
As the SPL changes for a sound, our perception of pitch changes across the spectrum of frequencies, even when the *actual* pitch remains constant. Similarly, as the frequency (pitch) changes across the spectrum, our perception of loudness changes, even when the actual SPL is held constant.




For sounds above approximately 1000 Hz (the green arrow), as the SPL (in dB) increases, we perceive the pitch to be increasing, even though the actual frequency remains constant.

Conversely, for sounds below approximately 500 Hz, as the SPL increases, we perceive the pitch to be decreasing.

The amount that we perceive the pitch to be changing is small, about 20–30 cents, which is about 1/5 to 1/3 of a semitone. However, this is a big enough change that we hear a lower or higher pitch.




The chart on the right (called an *equal-loudness contour*) shows how the SPL must be lower or higher to maintain the same perceived loudness levels. You can see that for any loudness level (designated by the red lines), when the sound is less than about 500 or greater than about 2000 Hz, the lower or higher the actual frequency, the higher the actual SPL must be for us to perceive that the loudness hasn't changed.

Some amplifiers have a "loudness" button that boosts low and high frequencies to offset the perceived change of pitch at the low and high ends; without this adjustment, the mid frequencies can dominate the sound.

### Experiment 4: **Hearing pitch change with increased dB**

1. Set octave=0, keys=12, count=1, offset=20, mod=off.
2. Turn your volume down *very* low.
3. Turn ON the following note: A3 (220.00 Hz, marked with a circle at the top of the key).
4. Listen to the pitch.
5. Turn OFF the note, then turn the volume up quite a bit.
6. Turn the same note ON again. You should hear that the perceived pitch has *dropped*.
7. Turn OFF the note, then turn the volume down again to *very* low (almost off).
8. Set octave=4, then turn ON A7 (3520 Hz, also marked with a circle at the top of the key).
9. Listen to the pitch.
10. Turn OFF the note, then turn the volume up – but only a small amount *(to avoid hurting your ears!)*.
11. Turn the same note ON again. You should hear that the perceived pitch is now *higher*!
12. The higher the note, the more apparent this loudness-to-pitch phenomenon becomes *(but be careful not to turn up the volume too high for these high frequencies!)*
13. The pitch shift you perceived in both cases is small – about 20–30 cents (1/5 to 1/3 of a semitone), but certainly big enough to hear.

This simple experiment shows that the sound level (dB) for pitches outside the 500–2000 Hz range need to be adjusted in order for the listener to perceive that the pitch is the same.

Note that you are listening to pure sine waves in *sinusedo*. When using more complex waveforms, such as those generated from instruments or voice, the perceived change in pitch is smaller (about 15–20 cents), but still noticeable. Also, the pitch of the harmonics will affect whether we perceive the pitch to be lower or higher. For example, for *any* pitched note that has a predominance of harmonics above 2000Hz, the perceived pitch shift will be upward.

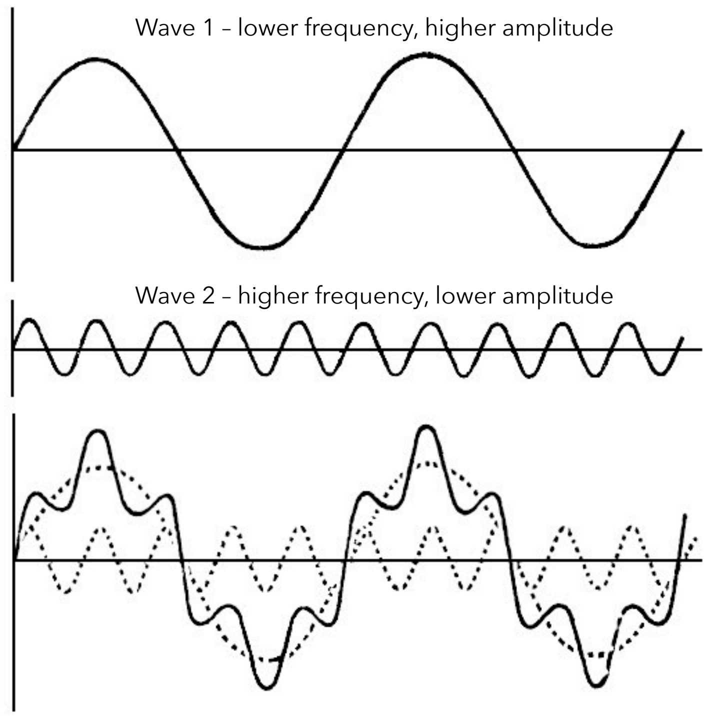
## Modulation and wave interference

As shown above, when two notes with different frequencies are played together, a new wave is created that has the sum of their frequencies at every point across the time spectrum. If both waves increase in amplitude or frequency at the same time, the resulting wave at that point in time is increased. The opposite is true if both wave decrease simultaneously. If one wave increases while the other decreases, the amplitude or frequency is reduced.

This additive effect when more and more simultaneously sounding waves having different amplitudes, frequencies, and waveforms (such as sine, sawtooth, triangle, pulse, etc.) can create any possible sound.

In fact, this is the principle of *additive synthesis*.

Suppose we have the following two waveforms, each with different frequencies and amplitudes (volumes).




Resulting wave is a combination of Wave 1 and Wave 2, obtained by adding the wave's y values for every point x on the time axis.

In *sinusedo*, you can create extremely complex waveforms constructed using only sine waves by varying the multiple frequencies and amplitudes.

The **mod** button enables you to add a ring LFO to each key individually. So you could have up to 144 LFOs running at the same time, each with a different modulation rate.

In addition, when two or more notes are less than about 20 Hz apart, they will produce beats. These beats also affect the waves to produce even more complex waveforms.

## Channels and external MIDI keyboard(s)

*sinusedo* allows you to use one or more external MIDI keyboard(s). However, because the number of notes per octave isn't the same as in our traditional 12-TET keyboards, there are several things to know.

MIDI supports up to 16 MIDI channels, and each channel can carry 128 notes (values).

Children can hear in the range of approximately 20 to 20000 Hz. By 18 years old, the upper range is only about 18000 Hz; by 50 years old about 12000 Hz; and by 70 only about 9000 Hz.

The frequency range in *sinusedo* is approximately 8 – 34000 Hz. This range is, of course, far wider than humans can hear, but this full range is kept in order to make the keyboard consistent for all EDO from 4 to 72 keys.

*sinusedo* offers 13 full octaves for all EDO layouts (12 using the **octave** and **count** buttons, and one extra octave using the **offset** button set to maximum). These octaves are numbered C-1, C0, C1, C2, ..., C11 (and the extra top note at C12 using offset).

The total number of notes in an EDO scale depends on the number of notes per octave (**keys**):

$$\text{number of notes for } k\text{-edo} = 13k - 1$$

For example:

$$\begin{aligned}
\text{9edo:} &\quad \text{number of notes } = (13 \text{ x } 9) \ \ - 1 = 116 \\
\text{10edo:} &\quad \text{number of notes } = (13 \text{ x } 10) - 1 = 129 \\
\text{19edo:} &\quad \text{number of notes } = (13 \text{ x } 19) - 1 = 246 \\
\text{31edo:} &\quad \text{number of notes } = (13 \text{ x } 31) - 1 = 402 \\
\text{72edo:} &\quad \text{number of notes } = (13 \text{ x } 72) - 1 = 935
\end{aligned}$$

Every MIDI channel can have, at most, 128 notes (values). For $k$-edo, $4 \leq k \leq 9$, a single MIDI channel will cover all the notes. For example, for 9-edo, there are 9 notes per octave, so there will be a maximum of 116 notes to cover all 13 octaves in sinusedo.

### How do we hear the all the notes when k ≥ 10?

We need to assign a different MIDI channel number to each group of 128 values:

$$\text{number of MIDI channels required } = \left\lceil \frac{k}{128} \right\rceil$$

*(where ⌈ ⌉ means to round up to the nearest whole integer)*

Looking at the above examples, for 10-EDO and 19-EDO, we will need 2 MIDI channels; for 31-EDO, we will need 4 channels; and for 72-EDO, we will need 8 MIDI channels.

Obviously, using multiple MIDI keyboards, each set to a different channel, isn't practical or even possible for most people...

**So, what's the best way to do this?**

For now[1], there are two methods:

1. Set your MIDI keyboard to the MIDI channel that will cover the range you want *(see below)*; or
2. Use a separate app to re-map incoming MIDI data to any channel that you choose

**Which MIDI channel should I use?**

As described above, every MIDI channel can have a maximum of 128 values (notes).
So, for any $k$-EDO, we need to decide *which* 128 notes out of all the possible $13k - 1$ notes we want to use on our external keyboard (of course, on the keyboard, we can use only 88 of them).

Fortunately, the range of frequencies we are able to hear is far less than 13 octaves!
And the number of frequencies that we would normally use is even less.

$$\text{number of available octaves using one keyboard} = \frac{128}{k}$$

For example, suppose you have decided to use 23-EDO. The number of octaves (23 notes per octave) available using a single MIDI keyboard is 128/23 ~ 5.6 octaves
or approximately 5 octaves + 13 notes.[2]

The table on the right shows the approximate frequency range of six keyboards ranging from 12-EDO to 72-EDO. The aqua highlighting shows the frequency range that you are most likely to use and its associated channel number.

For example, if you decide to use an EDO with about 20 or fewer notes per octave, channel 1 would cover most of the notes you need.

From about EDO 21 to 35, use channel 2.
From EDO 36 to 60, use channel 3 or 4.
Above that, use channel 4 or 5.

If you find the notes are too low or high, you can always adjust the channel number up or down by one.

In *sinusedo*, click **setup** then choose your MIDI controller for the *upper midi input*.[3] Set the MIDI channel to the one you feel covers the frequency range you wish, as shown in the table.

| EDO | From (in Hz) | To (in Hz) | Channel # |
|---|---|---|---|
| 12 | 8 | 12544 | 1 |
| | 13290 | >34000 | 2 |
| 17 | 8 | 1495 | 1 |
| | 1557 | >34000 | 2 |
| 23 | 8 | 378 | 1 |
| | 390 | 17919 | 2 |
| | 18467 | >34000 | 3 |
| 31 | 8 | 141 | 1 |
| | 144 | 2461 | 2 |
| | 2561 | >34000 | 3 |
| 53 | 8 | 43 | 1 |
| | 44 | 231 | 2 |
| | 235 | 1236 | 3 |
| | 1253 | 6594 | 4 |
| | 6681 | >34000 | 5 |
| 72 | 8 | 27.8 | 1 |
| | 28.0 | 95.2 | 2 |
| | 96 | 327 | 3 |
| | 330 | 1120 | 4 |
| | 1130 | 3839 | 5 |
| | 3876 | 13162 | 6 |
| | 13290 | >34000 | 7 |

---

[1] The next version of *sinusedo* will offer a method of playing all $13k - 1$ notes for $k$-EDO using a single keyboard.

[2] More precisely: $\text{number of available octaves using one keyboard}$

$$= \left\lfloor \frac{128}{k} \right\rfloor \text{octaves} + \left\lfloor k\left(\frac{128}{k} - \left\lfloor \frac{128}{k} \right\rfloor\right) \right\rfloor \text{notes}$$

[3] Although *sinusedo* won't prevent you from assigning the same channel and MIDI controller to both keyboards, this will create unpredictable (and usually undesirable) results.

## How is a traditional keyboard mapped to a *k*-EDO keyboard?

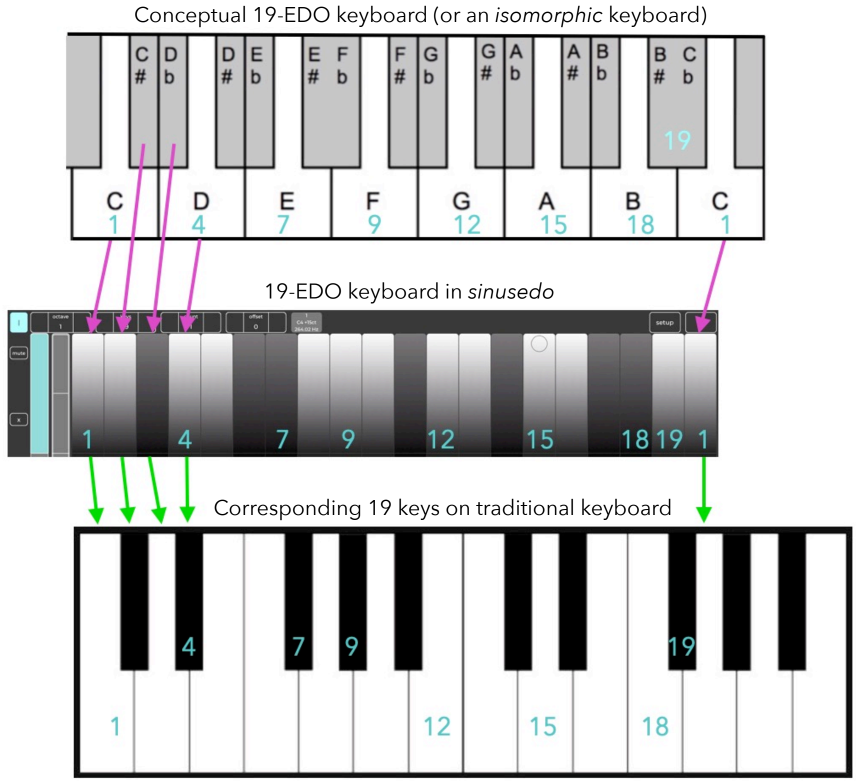
Obviously, if we set *sinusedo* to be 12-TET on one of the keyboards, then we would play the corresponding keys on our external keyboard. However, what if your *sinusedo* keyboard has fewer or more than 12 keys? It's simple! Starting with C, the keys are in a one-to-one corresponding order going up chromatically.

For example, suppose we set the upper keyboard in *sinusedo* to be 19-EDO. This particular EDO is popular because it's easy to represent all 19 notes using only ♯ and ♭.

In 19-EDO, C♯ is not the same as D♭. Same with D♯ and E♭. However, E♯/F♭ as well as B♯/C♭ are each a single note.

The keyboards on the right show how the keys of 19-EDO map directly onto the first 19 keys of *sinusedo* and a traditional keyboard.

As you can see, if we wanted to play the 19-EDO equivalent to a C major chord, we would need to play keys 1, 7, and 12. An octave on the MIDI keyboard is from key 1–20.




Playing traditional harmonies, such as a C major chord, using a non-12 EDO layout is possible and can often result in more pure major 3rds or 5ths than in 12-TET. In addition, you'll discover some really incredible chords that you could never produce in 12-TET.

As an example, 31-TET is popular because many intervals are closer to *just intonation* than in 12-TET, and the tuning is close to *meantone temperament*.[1] But more importantly, 31-TET offers some intervals and chords that are unique to our "12-TET ears." *(See the next experiment.)*

However, the goal of using various EDO isn't to replicate the harmonies and frequencies of 12-TET (or even of *just* intervals). Instead, your goal should be to discover new harmonic relationships built from combinations of notes to create sounds that are otherwise impossible to achieve using 12-TET.

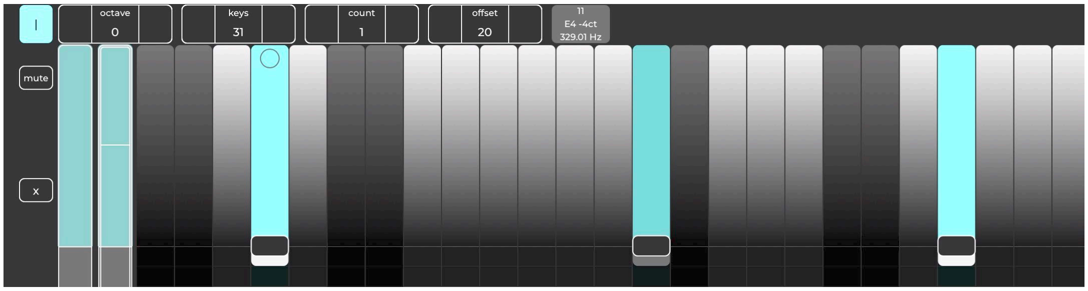
---

[1] In *just (pure) intonation*, intervals are tuned in whole-number ratios, such as 3:2 for a perfect 5th, 4:3 for a major 3rd, etc. By themselves, pure intervals sound beautiful because of the simple waveform interactions. However, just intonation fails as a tuning system because of *commas* – small intervals that cause some intervals to be extremely dissonant, which prevents us from playing in all major and minor keys because some 3rds and 5ths sound too harsh.

*Tempering* is the process of distributing the comma(s) among the other notes (in a sense, to break up one large problem into many smaller ones). 31-TET is a good approximation of *quarter-comma (1/4-comma) meantone temperament*.

**Experiment 5: Listening to 31-TET**

Set **octave**=0, **keys**=3, **count**=1, **offset**=20, **mod**=off.
Turn on the following notes: **A3** (220.00 Hz), **C#4** -13ct (275.12 Hz), and **E4** -4ct (329.01 Hz)




This is close to a **major** chord in 12-TET. Listen how pure this sounds. The reason is this major 3rd (A to C#) has the ratio of 275.12 / 220.00 = 1.251, and a *just intonation major third* is 5:4 = 1.25, nearly exact.

Similarly, this perfect 5th A to E has a ratio of 329.01 / 220.00 = 1.496, and a *just intonation perfect 5th* is 3:2 = 1.5.

Counting our first note (A) as 1, these are notes **1, 11, 19**.

Turn off the middle note and turn on note 9: **1, 9, 19**.
This is close to a **minor** chord in 12-TET but a bit brighter.

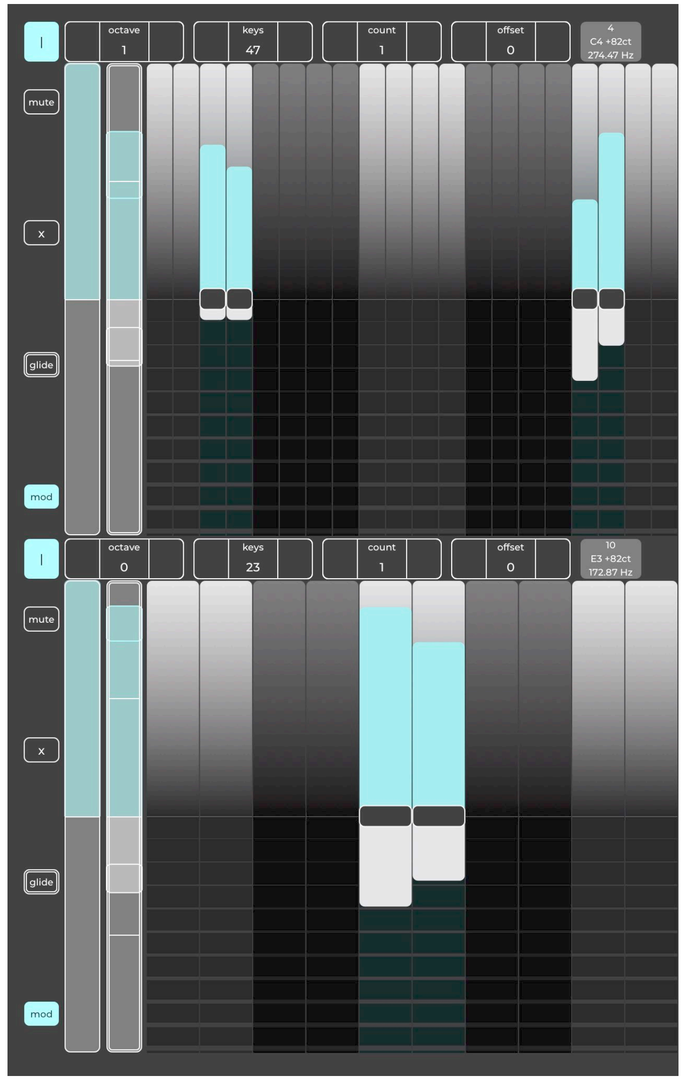
Now try the following three triads that we cannot achieve in 12-TET. For each one, turn off the middle note, and replace it with a new one:

| | | |
|---|---|---|
| **Neutral**: | **1, 10, 19** | This third is close to a just 11:9. It has a sound quite foreign to our 12-TET ears because it is halfway between a minor and major, which adds to its mysterious, almost mystical sound. |
| **Super Major**: | **1, 12, 19** | This has a bright, shimmering sound. Again, this is a sound that we cannot create in 12-TET, nor are we used to the sound. The ability to create new sonorities that break away from hundreds of years of 12-TET is the attraction to using different EDOs. |
| **Sub Minor**: | **1, 8, 19** | This is a "minor minor" chord. The third is close to a 7:6 ratio in just intonation. Like the chords above, this produces a sound that our ears can't quite identify and adds to its mystery. |
| **Sus4**: | **1, 14, 19** | This is a bit brighter than our 12-TET sus4 chord. |

Now, keeping the notes 1 and 19 (the root and 5th), try playing all the notes between them, one at a time, to create other interesting 3-note chords.

Try playing any of the above chords and add one or more notes on either side of these notes. You'll hear the added richness of the frequencies and the beating – and sounds you'll never achieve in 12-TET.

**Experiment 6:** In *sinusedo*, open two keyboards, press the 'x' buttons if any notes are playing.




Upper keyboard: octave=**1**, keys=**47**, count=**1**, mod=**ON**

Lower keyboard: octave=**0**, keys=**23**, count=**1**, mode=**ON**

On both keyboards, turn on the notes shown on the right (or close to these). Now adjust the volumes and mod rate sliders to be similar.

*Which components created this complex waveform?*

On the upper keyboard, the two notes on the left are creating a beat of about 4 beats per second (274.47 - 270.45). Their modulation sliders are set to 0, so you're hearing only the beats from them.

The rightmost two notes on the upper keyboard are also creating about 5 beats (5 Hz ~ 337.42 - 332.48). In addition, they have modulation turned on to different rates, one about 7 Hz, the other at about 4 Hz. (Note: Each horizontal line = 2 Hz for the LFO.)

So, just with the upper keyboard, we have beating at 4 and 5 Hz, and modulation at 4 and 7 Hz. When the original frequencies combine with 4, 5, and 7 Hz, a more complex wave results. On the lower keyboard, press **mute**=ON to hear only the upper keyboard. Adjust the volumes and mod levels to hear the interference patterns.

The lower keyboard has two notes, and is producing about 5 beats per second. In addition, the notes have modulation applied at 8 Hz and 6 Hz. The combination of the base frequency with the beats and different modulation creates a quite complex wave.

Listen to each keyboard separately, then turn them both on.

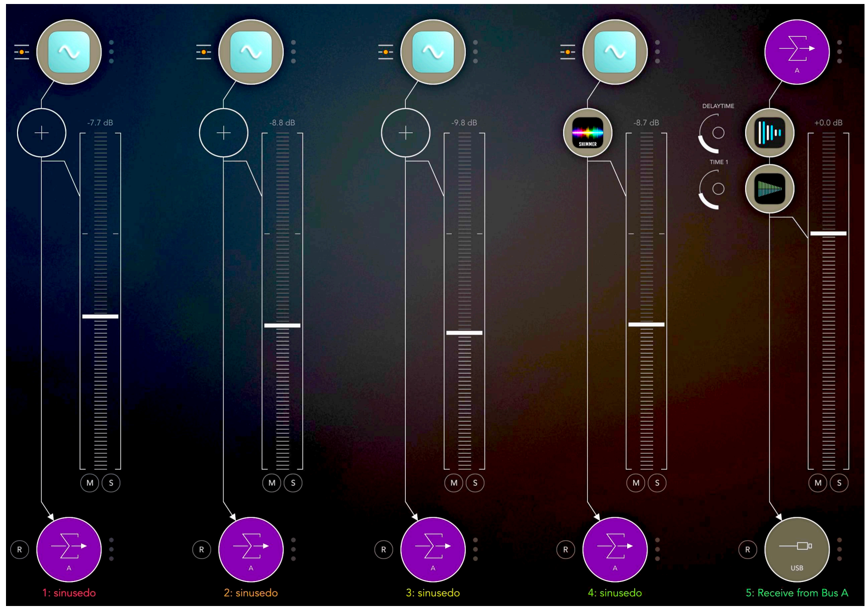
Turn **Glide**=ON for both keyboards (double-tap the Glide buttons). Slide one or more fingers across in the top half of either or both keyboards to hear the sweep of all the frequencies.

Turn on/off different notes, and experiment with their volumes and mod rates.

You should now have a good understanding of the interface, beats, modulation, and wave interference.

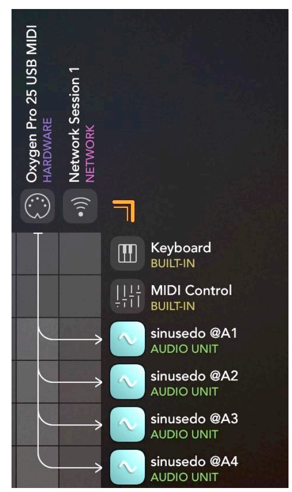
### Experiment 7: **Using multiple instances of *sinusedo***

In this example, we'll use 4 instances of in AUM, sending all four to the same mix bus onto which we can add any effects we wish, such as delay, reverb, gating, panning, compression, tape machine, transient shaper, or shimmer.



In AUM, assign 4 instances of *sinusedo* and send to a mix bus.
*Credits: Shimmer (by 4Pockets), BLEASS delay, Other Desert Cities (by Audio Damage)*

1. Create four instances of *sinusedo*.
2. To start, open each instance of *sinusedo* and turn off **mod** on the upper keyboards. This will allow us to hear the beats among the notes without changing the waves with modulation.
3. If you added any effects on the tracks or on the mix bus, disable them for now so that you can hear only the pure sine waves from *sinusedo*.
4. For this experiment, we'll set the number of **keys** in these four instances to 17, 23, 67, and 71. Of course, you can use any values you want from 4–72. We've chosen these values because they are prime and therefore won't have frequencies that are multiples of any other keys. This will increase the inharmonicity between notes and create interesting wave interferences.



In AUM, route your MIDI keyboard to all four instances of *sinusedo*.

5. *How do keys on a MIDI keyboard align with sinusedo keys?*

   Whereas there are 100 cents between adjacent notes (semitones) on a MIDI keyboard, there are fewer (or more) cents if k ≠ 12 on a $k$-EDO keyboard. In this example, the number of cents between each adjacent note in the four instances of *sinusedo* are, using formula [ 4 ]:

$$\frac{1200}{17}, \frac{1200}{23}, \frac{1200}{67}, \frac{1200}{71} \approx 70.59,\ 52.17,\ 17.91,\ 16.90 \text{ cents}$$

   So *sinusedo* must divide the number of $k$-EDO keys and distribute them as evenly as possible among the 12 MIDI keys per octave. Therefore, every adjacent key on the MIDI keyboard will skip several keys on the $k$-EDO keyboard.

$$\frac{17}{12} \approx 1.4, \frac{23}{12} \approx 2.7, \frac{67}{12} \approx 5.6\ , \text{ and } \frac{71}{12} \approx 5.9$$

   This tells us that every every note we play on a MIDI keyboard will skip either 1 or 2 notes on a 17-EDO keyboard, 2 or 3 notes on a 23-EDO keyboard, either 5 or 6 notes on a 67-EDO or 71-EDO keyboard. You can see this by playing consecutive notes on a MIDI keyboard and watching where they appear on these four *sinusedo* keyboards. The higher the EDO, the more *sinusedo* keys will be skipped.

6. Try playing some notes and chords on the MIDI keyboard. *sinusedo* maps each note to the closest frequency on the *k*-EDO keyboard.

   For example, when you play A4 on the MIDI keyboard, the four *sinusedo* keyboards for 17, 23, 67, and 71-EDO will play the frequencies: 389.34, 390.03, 392.67, and 391.36.

   Notice that these are all within 3 beats (3 Hz) of each other. For example, if you disable the 23 and 71 instances and enable only the 17 and 67 instances then play A4, you should hear 392.67 – 389.34 = 3.33 beats per second, or about 10 beats every 3 seconds. Do the same using different combinations of *sinusedo* instances. This is a good way to train your ear to hear the beats and to appreciate the interaction among waves. When you have all four enabled, listen to the much more complex wave forms. The waveform will be different with every combination of notes because the various frequencies produce different number of beats and interference.

7. Open one of the *sinusedo* windows, and while you're playing notes or chords on the MIDI keyboard, turn on/off different notes in the *sinusedo* keyboard(s). Double-tap on the **Glide** button, and try sweeping your finger on the upper or lower keyboards using one or more fingers.

   Try setting the lower keyboards to different EDO and play notes that are as close as possible to the frequencies you're playing on the MIDI keyboard.

8. Turn on the mod buttons, add a few notes on the lower or upper keyboards, and set the mod rates to be different from each other to create even more beating and to create much more complex waves.

9. Try other combinations of keyboards. For example, try 12, 13, 14, 15 EDO or those at the highest end (with the smallest ratios between consecutive notes): 69, 70, 71, 72.

10. Try adding other synths and percussion. Experiment adding various effects.

# OMNI mode – hearing notes and chords in $n$-EDO

*sinusedo* enables you to play notes and chords on an external MIDI keyboard (which is of course in 12-TET) and hear the closest frequencies in $n$-EDO.

How do the intervals of a major 3rd (A-C#) and a perfect 5th (A–E) sound in 31-EDO compared with 12-TET? What about a minor 3rd (A–C)?

Using *sinusedo*, you can explore how the frequencies of individual notes and chords that we are accustomed to hearing in 12-TET sound in other EDO tunings. Also, playing the same notes and chords in two EDO tunings at the same time can produce some interesting sounds.

As an example, in *just* intonation[1] a major 3rd has a ratio of 5:4. So a M3 above A4 (440.0 Hz) would be 440.0 x 5/4 = 550.0 Hz. In 12-TET, however, a M3 above A4 has the frequency of about 554.37 Hz, which is about 14 cents higher than a pure M3, certainly a large enough difference to hear.

By contrast, in 31-EDO, the *closest note* that is a M3 above A4 is 550.25 Hz, which is 8 notes above A4.[2] This is only about 0.8 cents higher, or less than 1% of a semitone difference – too small for our ears to distinguish.

Similarly, a pure minor 3rd has a ratio of 6:5, so a m3 above A would be 440.0 x 6/5 = 528.0 Hz. In 12-TET, a m3 above A is 523.25, a difference of about 16 cents – large enough to hear. In contrast, using 31-EDO, the closest m3 to A4 is 526.18 Hz, a difference from pure intonation of only about 6 cents – too small to hear.

When this same m3 is played simultaneously in 12-TET and 31-EDO, you'll hear |523.25 – 526.18| ~ 3 beats per second.

---

[1] In *just (pure) intonation*, intervals are tuned in whole-number ratios, such as 3:2 for a perfect 5th, 4:3 for a major 3rd, etc. These ratios are based on the harmonic series. Many tuning systems developed over many centuries have strived to find a balance between having the ratios of all intervals as close as possible to these pure ratios and having the flexibility to transpose music to all keys. Western music has settled on 12-TET which allows us to transpose to any key, but this places some intervals more "out-of-tune" with their just intonation counterparts.

[2] See page 30 for a method to calculate the closest note in $n$-EDO to any note in 12-TET.

### Experiment 8: **Using an external keyboard to hear *n*-EDO notes and chords**

1. In AUM, be sure you have routed your external keyboard to sinusedo.
2. In sinusedo, open the setup menu, and set the *upper keyboard midi mode* to OMNI.
3. Set the upper keyboard to octave=1, keys=**31**, count=2, mod=OFF
4. Set the lower keyboard to octave=1, keys=**12**, count=2, mod=OFF
5. On the lower keyboard, set mute=ON (lit), then turn on the notes A4 and C5.
6. On the upper keyboard, set mute=ON (lit), then hold down notes A4 and C5 on your external keyboard.
7. On the lower keyboard, turn mute OFF (unlit) and listen carefully to this minor 3rd interval in our traditional 12-TET.
8. On the lower keyboard, set mute=ON (lit) and the upper keyboard mute=OFF (unlit). You're now hearing a m3 interval in 31-EDO.
9. Jump back and forth between them. Can you hear the small difference? It's very small – about 10 cents, or 1/10th of a semitone, but big enough if you listen carefully. The 31-EDO minor 3rd interval is a bit larger (sharper) than in 12-TET.
10. Now turn both mute buttons OFF (unlit) so you are hearing 12-TET and 31-EDO together. As explained above, you should hear approximately 3 beats per second.
11. Turn on only A4 and E5 on the lower keyboard, and hold down only A4 and E5 on your external keyboard.
12. Listen to this perfect 5th played together in 12 and 31 EDO.

    A pure P5 would be 440.0 x 3/2 = 660.0 Hz.
    In 12-TET, it is 659.26 Hz, and in 31-EDO, it is 658.03.

    So, when hearing them at the same time, you should hear approximately | 659.26 – 658.03 | = 1 beat per second.
13. Now add the C5 on both keyboards. If you listen carefully, you should hear the 3 beats per second between the two major 3rds, and the 1 beat per second between the two perfect 5ths.
14. Experiment listening to different melodies and chords in various n-EDO tunings, by themselves and compare them with the same notes in our traditional 12-TET tuning.

**Method to calculate the closest note in $n$-EDO to any note in 12-TET**

The formula and example below are only for those readers who are curious how to find the correlation in frequency and note number from any note in 12-TET to its closest match in n-EDO.

For example, if I play a C4 in 12-TET (~261.63 Hz) and want to know the frequency of the closest note in 31-EDO and its location (number of notes away from A4), how would I do this?

An easy method is to calculate the minimum and maximum number of notes away from A4 the note could lie and the frequencies of these notes. Whichever one is closest in frequency to the note in 12-TET is the one we want.

$$
\begin{aligned}
&[1] \quad n_{min} = \left\lfloor \frac{dk}{12} \right\rfloor, \quad n_{max} = \left\lceil \frac{dk}{12} \right\rceil \\
&[2] \quad f_{min} = 440.00 \times 2^{\frac{n_{min}}{k}} \\
&[3] \quad f_{max} = 440.00 \times 2^{\frac{n_{max}}{k}} \\
&[4] \quad f_{12} = 440.00 \times 2^{\frac{d}{12}} \\
&[5] \quad \text{if } (|f_{12} - f_{min}| < |f_{12} - f_{max}|), \\
&\qquad\quad \text{then } N = n_{min}, F = f_{min} \\
&\qquad\quad \text{else } N = n_{max}, F = f_{max}
\end{aligned}
$$

where:

$k$ = EDO number

$d$ = # notes above or below A4 in 12-edo

$n$ = approx # of notes away from A4

$f_{min}, f_{max}$ = freq of the 2 closest notes in $k$-EDO

$f_{12}$ = exact freq in 12-EDO

$N$ = exact # notes below or above A4 for our note in $k$-EDO

$F$ = exact freq of our note in $k$-EDO

**Example**: In 12-TET, a C4 has the frequency of 440.0 x 2^(-9/12) ~ 261.63 Hz
The closest note to this in 31-EDO can be computed as:

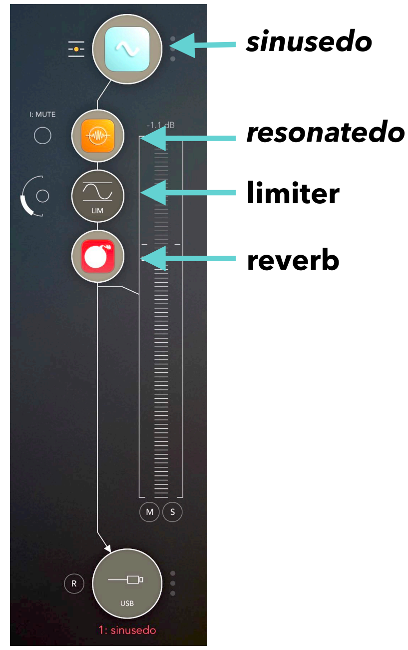
$$
\begin{gathered}
n_{min} = \left\lfloor \frac{dk}{12} \right\rfloor = \left\lfloor \frac{-9(31)}{12} \right\rfloor = -24 \\
n_{max} = \left\lceil \frac{dk}{12} \right\rceil = \left\lceil \frac{-9(31)}{12} \right\rceil = -23 \\
f_{min} = 440.0(2^{\frac{n_{min}}{k}}) = 440.0(2^{\frac{-24}{31}}) \approx 257.28 \text{ Hz} \\
f_{max} = 440.0(2^{\frac{n_{max}}{k}}) = 440.0(2^{\frac{-23}{31}}) \approx 263.09 \text{ Hz} \\
f_{12} = 440.0(2^{\frac{-9}{12}}) \approx 261.63 \text{ Hz} \\
|f_{12} - f_{min}| = |261.63 - 257.28| = 4.35 \\
|f_{12} - f_{max}| = |261.63 - 263.09| = 1.46 \\
1.46 < 4.35 \;\therefore\; N = -23 \text{ and } F \approx 263.09 \text{ Hz}
\end{gathered}
$$

Therefore, the closest note to C4 in 31-EDO has a frequency of ~263.09 Hz and is located 23 keys below A4. A just (pure) C4 would be 264.0 Hz, so the 31-EDO C4 is only 1 Hz lower, or about 6 cents lower.

## Experiment 9: **Combining *sinusedo* with *resonatedo***

*resonatedo*[1] (also by Alex Nadzharov) is a resonator effect with microtonal pitch control that is an audio-effect counterpart for *sinusedo*.

It can be used as a standalone or AUv3 effect for any incoming audio; however, its microtonal feature set and two-keyboard layout are paired with those of sinusedo, so they both make an ideal microtonal synthesizer-effect pair.




1. In AUM, create an audio channel and add *sinusedo*.
2. Add *resonatedo* as an AU extension below it.
3. Add a limiter as an AU extension below resonatedo. (AUM's built-in Signal Processing > Dynamics > Peak Limiter is fine for this purpose, although you can use a more sophisticated limiter if you wish.)

   Note that although a limiter isn't necessary, it's advisable because when the frequencies of the notes in *sinusedo* are close to (or exactly the same as) those set in *resonatedo*, the volume due to high resonance at those frequencies will often cause extreme distortion, especially when the *color* parameter of *resonatedo* is high
4. If you wish, add any reverb after the limiter.
5. In *resonatedo*, set octave=1, keys=12, count=2, mix=50%, color=50%. Turn on five or six notes.
6. In *sinusedo*, set octave=1, keys=17, count=2, glide=on
7. Now glide your finger along the top keyboard in *sinusedo*. When the frequencies are close to those of the keys you have chosen in *resonatedo*, you will hear the increase in resonance.
8. Try turning on a few notes in sinusedo and experiment with different amounts of modulation.


---

[1] Here are the links for the *resonatedo* **manual**, **web site,** and **app store**.

9. In *sinusedo*, open the setup menu and change the MIDI mode to OMNI.
10. Turn off all the notes in *resonatedo*.
11. In AUM, route your external MIDI keyboard to *sinusedo,* and route AUM's built-in keyboard to *resonatedo. Open the AUM keyboard*
12. On AUM's keyboard, hold down any combination of notes (or press the ∞ symbol to sustain the notes you press).
13. On the external keyboard, play a melody (or just random, single notes) in *sinusedo*. Listen to the resonance when those notes are near the ones you are holding down with AUM's built-in keyboard.

    You can also play chords on the external keyboard, and you will hear these chords interpreted in the closest frequencies in whatever EDO you've chosen (in this example 17-EDO).
14. At any time you can change the notes held in *resonatedo* to have different resonant notes sound, even with the same sequence of notes played on the external keyboard.
15. You can also try gliding across the keys in *sinusedo* or combine any of these techniques to create interesting resonances and modulations.
16. Experiment with different levels of the mix and color parameters in *resonatedo*.

## Conclusion

Playing *sinusedo* and a MIDI keyboard while using multiple number of EDO keys, glide, and modulation then applying any effects you want can create beautiful sonorities that are impossible (or certainly difficult) to create using only 12 MIDI notes per octave. Sometimes the unique sounds you can get from *sinusedo* are great to lay above or below your main tracks.

*Most of all, have fun!*

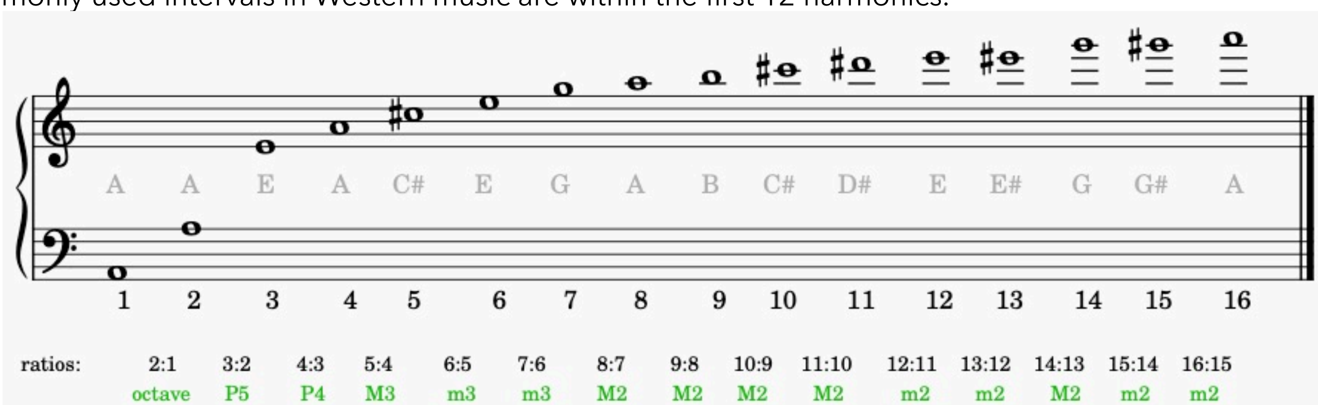
We hope you've enjoyed experimenting with the various aspects of *sinusedo*.
Please share any compositions you create using it. If you have any comments or questions, please contact us!

# Appendix: EDO tunings – a brief look

To understand the historical, musical, and mathematical reasons why Western music slowly gravitated toward today's 12-tone equal temperament (one of an infinite number of EDO tunings), it's necessary to briefly look at the *harmonic series*, *just intonation,* and *temperaments*.

Pythagoras (c.569–475 BC) is credited as the first person (or certainly among the first) to discover and document the harmonic ratios of a strings (monochords) of length 1x, 2x, 3x, 4x, 5x, ...played with each other. The first pure, whole-number ratio is 2:1, which we call an *octave*. The next higher, whole-number ratio is 3:2, a string of length 3x vibrating together with a string of length of 2x. This produces a harmonic relationship that today we call the perfect 5th. Similarly, the next pure interval is 4:3, which we call a perfect 4th, followed by 5:4, which we call a M3, etc.

This process continues, resulting in *just* (or *pure*) intervals. Here are the first 16 notes of the harmonic series built starting on A. The close approximation to our traditional interval names are shown in green. You can see that as the whole number ratios increase, the sizes of the intervals decrease. Almost all of our most commonly used intervals in Western music are within the first 12 harmonics.




The goal of theorists and musicians for the past two millennia has been to find a tuning system that meets two requirements:

1. All intervals use only low, whole-number ratios *(such as 3:2, 5:4, 6:5, 5:3, etc.)*, and
2. All melodies, intervals, and chords can be transposed to any key and sound equally good

*Unfortunately, achieving both of these two seemingly simple criteria is, in fact, impossible.*

***Why?*** You can see from the above chart that the simplest, pure intervals are the octave (2:1), the perfect 5th (3:2), the perfect 4th (4:3), and the major 3rd (5:4).

What if we could construct a tuning using only perfect 5ths? That is, start on any note and go up a perfect 5th. Continuing this, we would end up back on the same note (7 octaves higher). For example, starting on A:

**A** E B F# Db A♭ E♭ B♭ F C G D **A**

This is 7 full octaves – almost the entire range of an 88-note keyboard. So, we would have all 12 possible notes, and we could start on any key.

Starting with any note of frequency $f$, we multiply each successive note by 3/2 (a perfect 5th) twelve times to arrive at A. Alternatively, we could start with that same frequency $f$, and multiply it by 2/1 (an octave) seven times to also arrive at A.

**Then we would have the perfect tuning system that would meet both requirements. *Right?***

Yes...

*but only if twelve perfect 5ths equals seven octaves*:

As you can see, 12 P5 ≠ 7 octaves. When we arrive at the highest note A, there is a difference (error) of almost a quarter of a semitone – definitely large enough to hear!

Although this method satisfies criteria 1 above for an ideal tuning, it violates criteria 2 – because of this large difference, known as the *Pythagorean comma*, we can't transpose intervals and chords to all the keys without some of the most common intervals (such as P5, P4, M3) sounding extremely harsh.

$$\left(\frac{3}{2}\right)^{12} \stackrel{?}{=} \left(\frac{2}{1}\right)^{7}$$

$$129.746337891 \neq 128.00000000$$

$$\text{This is a frequency ratio difference of}$$

$$\frac{\frac{3}{2}^{12}}{2^{7}} = \frac{3^{12}}{2^{19}} = \frac{531441}{524288} = 1.01364326477\ldots$$

$$\text{known as the } \textit{Pythagorean comma}$$

$$\text{This error is approximately}$$

$$1200 \times log_2(1.01364326477) \approx 23.5 \text{ cents}$$

$$\text{which is about a } quarter\ of\ a\ semitone$$

***So, what's the solution?***

What if we took this large difference of 23.5 cents and distributed it among several other intervals? That is, could we "hide" that large error by splitting it up into many smaller errors and spreading it among the 12 notes?

*Yes*. That's exactly the goal of *temperaments*. So when you see the term *temperament*, it refers to any tuning method where the *comma* (error) of some size is split up among the other intervals to be less noticeable.

*Does temperament meet both criteria?* Strictly speaking, no. Temperaments are *compromises* – making some of the intervals a little bit "worse" (narrower or wider) than their *just (pure)* counterparts with the goal of meeting criteria 2 so that we can play in any key.

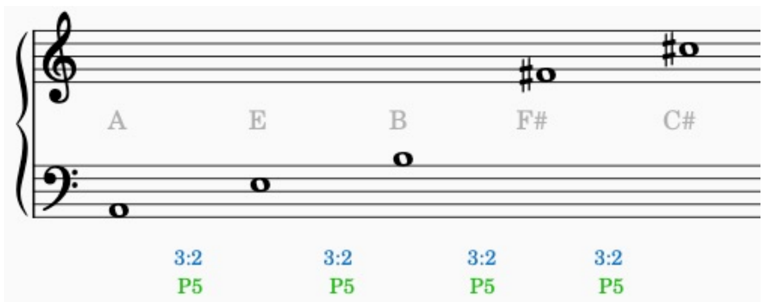
### What about the other intervals in Pythagorean tuning?

Although all the 5ths are *just (pure)* with a 3:2 ratio, how are the other intervals?

During the medieval period (approximately the 4th through 14th centuries), the predominant two intervals were the octave and 5th. Other intervals were used much less often, and "harmony" was still a result of horizontal movement in different voices. Toward the last part of the medieval period, these other intervals were becoming more popular.

By the Renaissance period, vertical harmony was not only standardized but beginning to be formalized. Because vertical harmony was important, other intervals, such as the major 3rd played a much bigger role and therefore needed to be as pure (just) as possible.

Look at the first 5 notes of Pythagorean tuning (which, as shown above, is constructed of perfect 5ths):




The fifth note is a C#, which is a major 3rd above an A (2 octaves lower). Looking at the earlier harmonic series, an ideal major 3rd has a ratio of **5:4**. However, in Pythagorean tuning, you can see that this major 3rd is four perfect 5ths above A, which is the ratio of **81/64**:

$$\left(\frac{3}{2}\right)^4\left(\frac{1}{2}\right)^2 = \frac{3^4}{2^6} = \frac{81}{64} = 1.265625$$

$$\frac{5}{4} = 1.25$$

This means that if we use Pythagorean tuning, we not only end up with a Pythagorean comma (error) of about a quarter of a semitone, but also have major 3rds that are about 1/5th of a semitone too sharp (wide) from their just (pure) counterparts.

$$\frac{1.265625}{1.25} = 1.0125$$

$$\approx 1200 \times log_2(1.0125) = 21.51 \text{ cents}$$

$$\textit{(about 1 / 5 of a semitone)}$$

This difference (1.0125) is called the **syntonic comma**[1].

---

[1] This is also known as the *chromatic diesis*, the *Didymean comma*, the *Ptolemaic comma*, or the *diatonic comma*. There are many ways to calculate the syntonic comma, all of them derived by stacking different sized intervals then comparing them with simple just (pure) ratios.

For example, one octave plus a *just* m3 is 2:1 x 6:5 = 12:5. Three *just* P4 is $(4:3)^3$ The difference is:

$$\frac{\left(\frac{2}{1}\right)\left(\frac{6}{5}\right)}{\left(\frac{4}{3}\right)^3} = \frac{\frac{12}{5}}{\frac{64}{27}} = \frac{81}{80} = 1.0125$$

### How can this problem be solved so that the major 3rds sound better?

One of the earliest solutions for resolving this issue was a method called **Meantone Temperament**, and was widely used from the early 16th century to the end of the 19th century.

There are several variations of meantone temperament, but the two most common are called **quarter-comma** and **third-comma** meantone:

**Quarter-comma meantone temperament** narrows the first four 5ths by ¼ of the syntonic comma: The syntonic comma (1.0125) is divided into four parts (1.0125^(¼) = 1.00311045746) and applied to the first four *ascending* 5ths (such as A-E-B-F#-C#). This produces a pure major third (5:4). Although the 5ths are now 1.003 narrower than pure, this is only about 5 cents (about 1/20th of a semitone), which is too small for us to hear.

Another common type of meantone temperament is called **third-comma meantone**, which divides the syntonic comma into 3 equal parts (1.0125^(⅓) = 1.00414942512) and narrows the first 3 *descending* 5ths (such as A-D-G-C) to produce a just minor 3rd, A-C (6:5). This narrows the 5ths by only about 7 cents (about 1/14th of a semitone), also too small to hear.

### Does this solve our problem?

No, but it moved us closer. Pythagorean tuning gave us pure 5ths, but resulted in a large comma (error). This tuning worked fine for music of the Medieval period when music was composed primarily horizontally, and the two primary intervals were the octave and 5th, but the 3rds were not close to pure.

Meantone temperament improved this by adding pure major 3rds, which were important starting in the late Medieval and throughout the Renaissance periods. The pure 5ths were narrowed, but only by a small amount, so they still sounded good.

The problem is that meantone is tuned starting from a specific note (in our example, A). Some of the 3rds and 5ths were pure or almost pure, but others were extremely dissonant. Therefore, although meantone comes very close to satisfying criteria 1, it fails for criteria 2.

### Is there *any* solution to this tuning problem that actually works?

As explained earlier, *no – there is no perfect solution to satisfy both criteria.*

But in the 19th century, as composers required the freedom to modulate to any key without these harsh dissonances, a major compromise was adopted: **12-note equal temperament.**

Unfortunately, this name is a misnomer. In its strictest sense, temperament distributes a comma (error) among several intervals in order to make some of the intervals pure (or closer to pure) and to eliminate as much as possible strong dissonances in *most* keys.

$n$-note equal tuning, today called ***equal division of the octave*** (EDO), ignores such calculated, exact distribution of the comma and instead simply divides an octave into $n$ equally sized intervals. That is, in EDO tunings, most of the intervals are not *just*; that is, all the intervals except for the octaves don't sound as they "should" in *just* intonation. Every interval is either too flat or too sharp.

### Then why doesn't equal temperament sound bad?

Well, to early adopters, it probably did. Hearing perfect *just* 5ths, 3rds, 6ths, etc. then changing to a system where *none* of them were pure was, at the very least audible and sounded "wrong", and at the worst quite jarring.

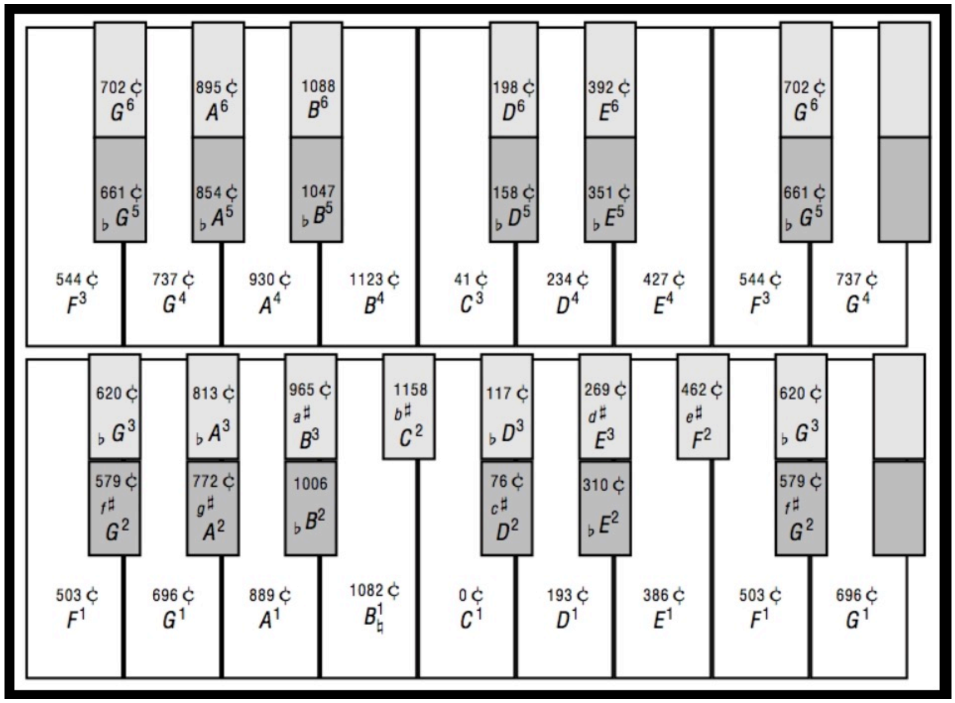
But for more than a hundred years, the majority of Western music has adopted 12-note equal tuning (12-EDO), and we've grown accustomed to the sound of these intervals – so much so that now when we hear some of the *just* (pure) intervals played with those of 12-EDO, they sound "wrong" – too flat or too sharp.

### Where do EDO tunings now fit in the spectrum? Why are they used?

Throughout the 20th century, as composers strived to break away from hundreds of years of tradition in terms of harmony, rhythm, organization, and other accepted historical traditions, a logical step was also to break away from having only 12 notes.

But this wasn't innovation – it was reinventing and improving on the concept of having more (or fewer) than 12 notes per octave. String, wind, brass, and percussion instruments as well as the voice can produce music that is microtonal (smaller than 12 notes per octave). So from ancient time, singers and performers have used microtonality in music, although it wasn't notated using microtonal symbols nor was it developed into a standard or widely used.

During the Renaissance, the music theorist and composer Nicola Vicentino (1511–c.1575) invented a 31-note keyboard, called the *Archicembalo*.
He was among the most progressive musicians of the time. He wrote and published a book of compositions using 31 notes per octave. He believed that the use of microtones in singing and instruments was necessary in order to play pleasing (*just*) intervals in any key.




Using 31 notes per octave (31-EDO), he could play in meantone temperament in all keys and produce *just* (or nearly just) 3rds and 6ths, with the sacrifice being slightly narrowed 5ths which produced beating.

Although several other composers and theorist experimented with EDO tunings that had fewer or more than 12 notes per octave, microtonal tuning was never popular because of the extreme difficulty building, tuning, and playing such instruments. In addition, 12-note meantone and later 12-EDO was so widely accepted that breaking away from it was difficult.

However, the 20th century saw a rebirth and expansion of microtonal systems as a way to create new harmonies free of old traditions.

Today, *xenharmonic* music has become increasingly popular because of the unique melodies and harmonies it can offer. Computers enable us to quickly and accurately play in any tuning system, and software as well as hardware keyboards enable us to play with any number of notes per octave.

### EDO vs. non-EDO tuning methods

Dividing the octave into $n$ equal parts is popular because it allows us to play the same melody and chords starting on any key without changing the ratios between notes.

But all EDOs sound completely different from one another. Playing a melody using 17 notes per octave and one with 15, 23, 27, 33, or 53 notes sounds radically different. Each of these have a unique character that you cannot achieve using a different tuning.

For example, 31-EDO (which is what Vincentino used for his instrument) enables you to play meantone tuning in all keys. Adding one more note, 32-EDO, has a completely different sound.

EDOs which offer a certain degree of similarity with 12-EDO Western tuning, especially 5ths that are close to pure (3:2) are 17, 19, 22, 29, 31, 39, 49, 50, and 53.

EDOs that are more distant in relationship to 12-EDO are 26, 27, 32, 33, 37, and 43.

For intervals and sounds that are even further removed from those of 12-EDO, try 11, 13, 14, 15, 16, 18, 20, 21, 23, and 25.

*sinusedo* enables you to play any EDO from 4–72 and have different EDOs on each of the two keyboards. Playing in two different EDO at the same time, for example one keyboard for chords and the other for melody, can create some amazing, non-traditional, complex harmonies.

In addition to dividing the octave into equal-sized intervals, some applications enable you to experiment with tunings where you can define not only the number of notes per octave but also the exact size between each of the notes. The product of all ratios must equal 2 in order to create an octave.

For example, you may want a 7-note octave and specify the ratios of:
1.10000000, 1.11818182, 1.11382114, 1.10948905, 1.15131579, 1.06285714, 1.07526882
The product of these 7 ratios is 2.000000 (an octave).

non-EDO tunings can create some interesting melodies and chords and are certainly worth exploring. Dozens of non-EDO tunings are used throughout the world and each have a distinct sound.

### Conclusion

The world of microtonal tuning theory, instruments (analog, digital, and software), music, and performance is enormous and can become extremely complex, both conceptually and mathematically.

*sinusedo* offers an easy way to experiment with various EDO tunings by themselves and in combination with other EDOs. Breaking away from (or used in conjunction with) our traditional 12-EDO Western tuning will hopefully inspire you to create new sounds and open new doors aurally and compositionally. *Have fun!*