# PedalScope Methods, Explained

A plain-language companion to the technical reference

PedalScope

Describes PedalScope 1.1 (43)
Published 30 September 2026

© 2026 Richard Hoge. This document is licensed under CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/). The accompanying scripts are licensed under MIT (https://opensource.org/license/mit). PedalScope and its mark are not part of either grant.

## Contents

**1 What this companion is** **4**

**2 Harmonic Distortion** **4**

2.1 What this measurement tells you, and what it cannot . . . 4
2.2 The test signal . . . 6
2.3 How the answer is pulled apart . . . 7
2.4 THD, in one sentence . . . 9
2.5 Three checks the app makes before it believes a number . . . 9
2.6 What the result looks like . . . 10
2.7 How you can check that the app does what this says . . . 11
2.8 Where it stops being a measurement . . . 13

**3 Transfer Curve** **14**

3.1 What this measurement tells you, and what it cannot . . . 14
3.2 The test signal . . . 16
3.3 How the picture is built . . . 16
3.4 Three things the app checks before it believes the picture . . . 18
3.5 The settings . . . 20
3.6 What the result looks like . . . 20
3.7 How you can check that the app does what this says . . . 20
3.8 Where it stops being a measurement . . . 22

**4 Compression** **25**

4.1 What this measurement tells you, and what it cannot . . . 25
4.2 The test signal: a ladder of loudnesses . . . 26
4.3 How the curve is built . . . 27

4.4 What the app decides when it reads a record back . . . 28
4.5 The settings . . . 32
4.6 What the result looks like . . . 32
4.7 How you can check that the app does what this says . . . 34
4.8 Where it stops being a measurement . . . 36

**5 Gain Map** **38**
5.1 What this measurement tells you, and what it cannot . . . 38
5.2 The test signal, and the ladder of loudnesses . . . 39
5.3 How the map is built . . . 40
5.4 What the app decides when it reads a record back . . . 41
5.5 The settings . . . 45
5.6 What the result looks like . . . 45
5.7 How you can check that the app does what this says . . . 45
5.8 Where it stops being a measurement . . . 48

**6 Slots: how a spectrum is read** **49**

**7 Chord IMD** **50**
7.1 What this measurement tells you, and what it cannot . . . 50
7.2 The test signal, and the lattice . . . 51
7.3 How the answer is read . . . 52
7.4 What the app decides when it reads a record back . . . 54
7.5 The settings . . . 58
7.6 What the result looks like . . . 58
7.7 How you can check that the app does what this says . . . 59
7.8 Where it stops being a measurement . . . 62

**8 Waveform Matrix** **64**
8.1 What this measurement tells you, and what it cannot . . . 64
8.2 The test signal: a grid of notes and loudnesses . . . 65
8.3 How the pictures are cut . . . 66
8.4 What the app decides when it reads a record back . . . 68
8.5 The settings . . . 70
8.6 What the result looks like . . . 70
8.7 How you can check that the app does what this says . . . 71
8.8 Where it stops being a measurement . . . 74

**9 Floors: how deep a reading can see** **76**
9.1 What a floor is . . . 76
9.2 Where floors come from . . . 76
9.3 An amount, not a share . . . 77

9.4 The largest binds . . . . . 77
9.5 What a reading near a floor becomes . . . . . 78
9.6 Averaging lowers noise, not residue . . . . . 79
9.7 Above every floor and still not the pedal . . . . . 80
9.8 Floors are measured, never computed . . . . . 80
9.9 Where it stops . . . . . 80

**10 Glossary, in plain language** **82**

**Changes since the previous version** **90**

# 1 What this companion is

The technical reference states each measurement's method exactly, with equations and every parameter. This companion says the same things in plain language, one measurement per chapter, using the same defined terms — the glossary at the end is generated from the same source the technical document uses, so a word means one thing in both. It never says what a measurement sounds like: the instrument measures, and listening is yours. Two chapters are not measurements: Section 6, read before Chord IMD, says how a spectrum is read, and Section 9, after the six, gathers what they each said about floors into one account. Everything in this document describes a pedal that has settled; how a pedal changes over time is being measured next, and gets its own chapters when it arrives.

The chapters follow the technical document's headings in the same order, so the two can be read side by side: where this document says "the app cuts each blip out with a window", the technical document gives the window's shape and width. Every number quoted here is read from the same methods manifest the technical document reads, which the app's own code writes out; nothing is typed by hand. Each defined word is a link to its entry in the glossary at the end.

# 2 Harmonic Distortion

## 2.1 What this measurement tells you, and what it cannot

Play one clean note into a distortion pedal and more than one note comes out. The extra notes are not random: they sit at exact whole-number multiples of the note you played — twice its frequency, three times, four times, and so on. Those multiples are the note's harmonics. The Harmonic Distortion measurement lists how much of each harmonic the pedal adds, for every note across the range, at one loudness.

Two ideas make that sentence work, and the rest of the chapter leans on both.

**Why a changed shape is new notes.** A pure tone, drawn against time, is a smooth sine wave: one frequency and nothing else. Any wave that repeats at the same rate but has a different shape — flatter tops, a kink, a lopsided bottom — is made of that frequency plus its multiples. That is not a way of speaking; it is what "shape" means for a repeating wave. A wave is exactly a stack of harmonics, and changing its shape changes how much of each harmonic is in the stack. The fundamental is the note itself; each harmonic order names which multiple an added tone is.

**What clipping does to a sine.** A distortion circuit has a ceiling. Below it the wave passes through unchanged; at it the peaks are flattened or rounded — that is clipping. Figure 1 shows three versions of it, computed exactly, with each harmonic's level in decibels relative to the note that went in. A soft clipper (the top row, the tanh curve that serves as the worked example throughout) rounds the peaks and produces a ladder of harmonics that falls away quickly; a hard clipper flattens them and produces a ladder that falls away slowly, so the high harmonics stay strong. Both of those treat the top and bottom halves of the wave the same way, and both produce only the odd harmonics (third, fifth, seventh, . . . ); the even ones are exactly absent. The third version clips the bottom half less than the top, and the even harmonics (second, fourth, sixth, . . . ) appear. That is the whole rule: symmetry gives odd harmonics only, and any difference between the two halves adds the even ones.

The measurement, then, is a table: for each harmonic (the app reads 9 of them, the note itself and eight above it) and for each note in the range, how loud that harmonic is compared with the note that went in. It is drawn as one curve per harmonic across the notes — the harmonic

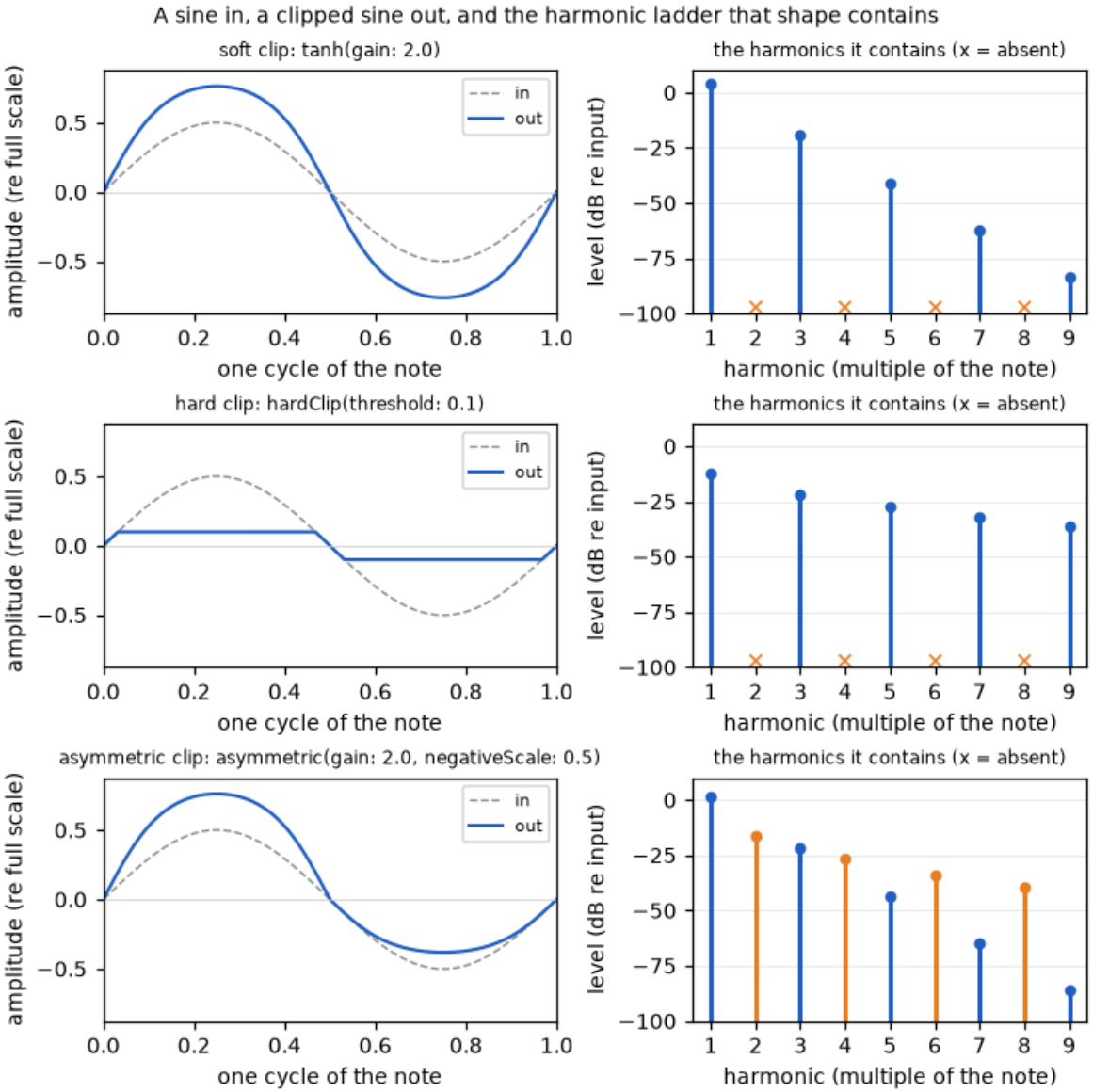



Figure 1: Three clippers, computed from their formulas at the amplitude the worked example uses (0.5 of full scale). Left: one cycle of a sine in (gray, dashed) and what comes out (blue). Right: the exact harmonic content of the output, in decibels relative to the input — each step of 20 down is a tenth of the size. Odd harmonics in blue, even in orange; a cross marks a harmonic that is exactly absent.

response — with a single-number summary, the total harmonic distortion, beside it. Everything is measured relative to the strength of the test tone (input-normalized), so a plain wire reads as "0 dB" for the note itself and nothing for any harmonic, and two measurements made at the same drive on different rigs can be laid over each other.

**What it cannot tell you.** Stated once, so nothing below has to hedge.

1. **Only one loudness.** One sweep is one drive level. A pedal that cleans up when you play softly, or hardens when you dig in, does so at other loudnesses this measurement never visits; the Gain Map and the Compression measurements own that axis.
2. **Only one note at a time.** Two notes played together make the pedal produce sums and differences of their frequencies as well as harmonics. That is a different measurement — Chord IMD — and nothing here predicts it.
3. **A number can be hiss, not harmonic.** The analysis reads whatever lands in each harmonic's slot, and the pedal's own noise lands there too. A separate check (harmonic shaping, below) says whether a reading is something the circuit did or just its noise; a reading is not automatically a harmonic.
4. **Nothing below the rig's own floor.** The interface, the cables and the converters have noise and distortion of their own. Below that level the app is measuring the rig, and it says so rather than quoting a figure (measurement floor, below).
5. **Nothing about listening.** The chart says how much of each harmonic there is. It says nothing about what the distortion sounds like.

## 2.2 The test signal

The test tone is a single sine that glides smoothly from a low pitch to a high one — from 20 Hz to 10 000 Hz over about 10 seconds in the standard plan — the synchronized swept sine. One signal visits every note in the range, so one recording holds the answer for the whole neck. Figure 2 shows its first quarter second and how its pitch climbs over the whole sweep: the pitch rises by the same musical interval in each second, which is why it draws as a straight line on the pitch axis of the lower panel. The instantaneous frequency is the pitch the tone is at, at any given moment.

The speed of the glide is not free to choose. The trick that makes the whole measurement possible is this: when a pedal clips a gliding tone, the second harmonic it produces is itself a gliding tone, at twice the pitch — which is to say, it is a copy of the test tone that is running ahead of it by a fixed amount of time. The third harmonic is another copy, further ahead; the fourth further still. For those copies to line up exactly, the tone's climb has to be timed so that it completes a whole number of cycles before each copy's head start — the synchronization condition, met by nudging the requested length slightly (the standard plan's 10-second request becomes 9.943 s; its rate constant is 1.6 s). The nudge is what lets each harmonic be pulled out afterwards as its own clean copy. In the standard plan the second harmonic runs 1.109 s ahead of the note, the third 1.758 s and the ninth 3.516 s.

The first and last moments of the tone are faded in and out rather than switched, so nothing bangs. During those fades the pedal was driven by a tone that was still rising or already falling, and a reading there describes the fade, not the pedal. The range of notes the tone played at full strength — the excited band, from 20.63 Hz to 9937.7 Hz in the standard plan — is where every number in the chapter is meant.

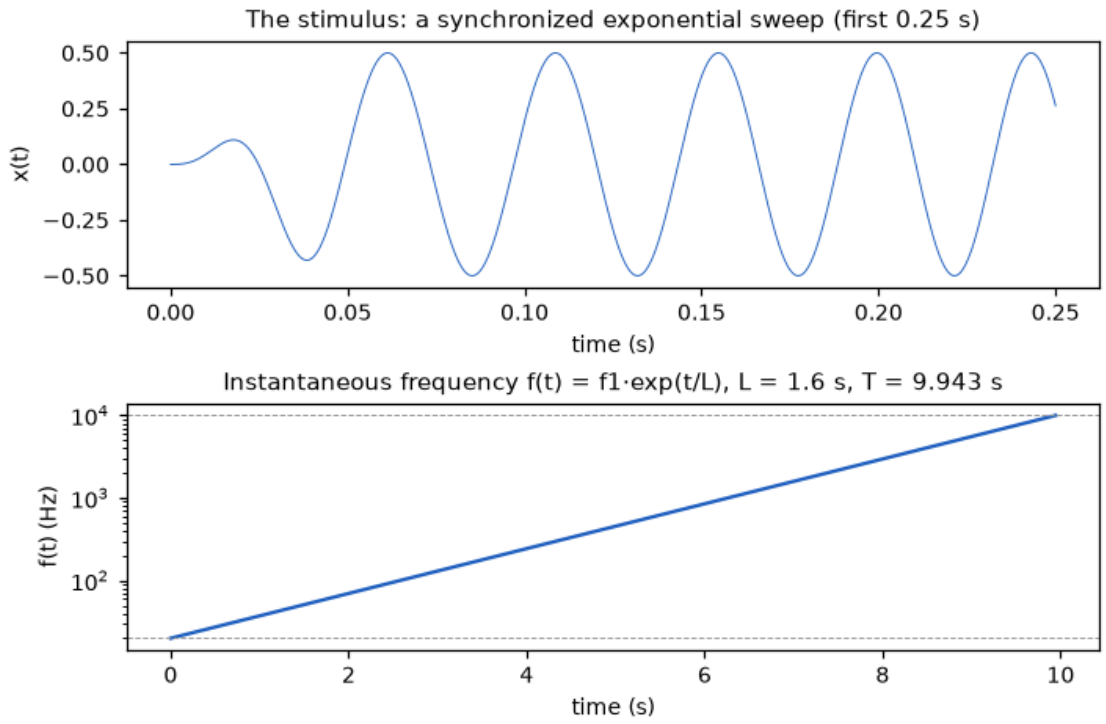



Figure 2: The test tone: its first quarter second (top) and its pitch over the whole sweep (bottom).

### 2.3 How the answer is pulled apart

The recording that comes back from the pedal is a gliding tone with its harmonics riding on it, all overlapping in time. Separating them takes three steps.

**Undoing the glide.** The first step is deconvolution: applying the exact mathematical undo of the test tone — its inverse filter, written down from the tone's formula rather than measured — to the recording. Undoing a glide turns it into a single sharp blip at one instant. Because every harmonic is a copy of the tone running a known amount ahead, each harmonic turns into its own blip, that same amount earlier in time. The result, the impulse response, is shown in Figure 3: the note itself is the blip at zero, and the harmonics stand in a row to its left, each at its own head start. Each one is a harmonic pulse. The example is a symmetric soft clipper, so only the odd harmonics have blips; at the even positions there is nothing, as there should be.

**Cutting each blip out.** The second step cuts each harmonic's blip out of that row with an extraction window: a slice of time centered on the blip, tapered smoothly to nothing at both ends so that the cut itself adds no edges. Each slice is as wide as it can be without reaching into the neighboring blip, up to a cap of 0.2 s on either side — so the note and the first few harmonics get the full 0.4 s, and higher harmonics, whose blips crowd closer together, get less (the ninth gets 0.1785 s). Figure 4 shows the nine slices over their blips. A wider slice can tell finer pitch differences apart, so the high harmonics are read a little more coarsely than the low ones. If a sweep is so short that a slice would shrink below 16 samples on either side, that harmonic and every one above it is not reported at all: a short sweep simply carries fewer harmonics.

**Reading a harmonic at a note.** Each slice, once cut out, holds one harmonic's behavior across every note at once. The app turns it back into a curve against frequency, and reads the harmonic for the note you care about at that harmonic's own frequency: the third harmonic of a

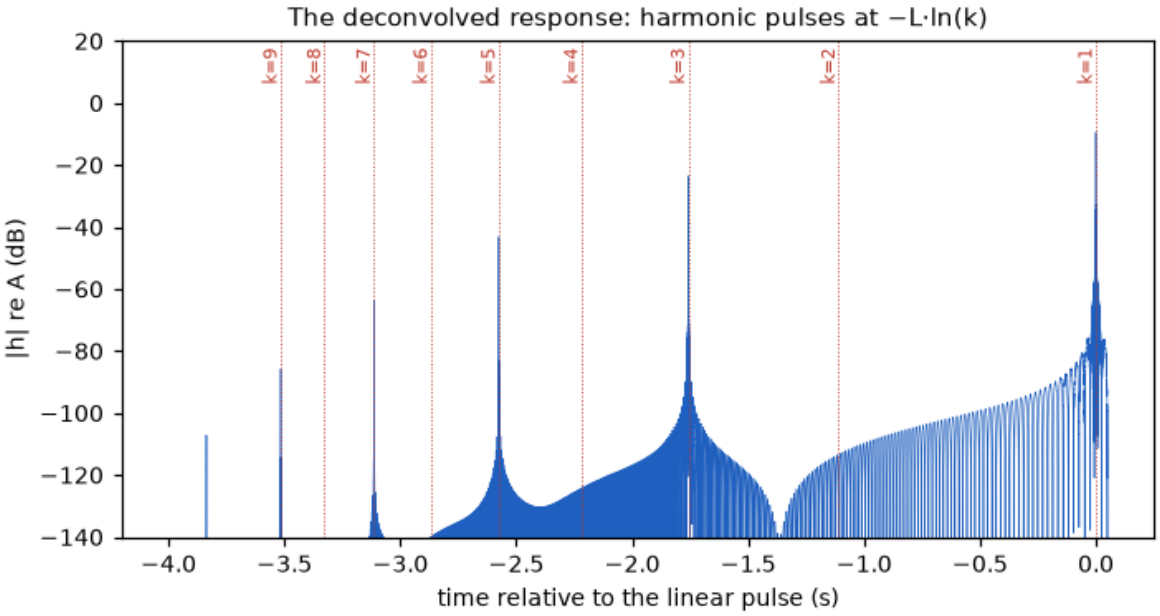



Figure 3: The recording after the glide is undone, for the worked example (a symmetric soft clipper). The note is the blip at time zero; each harmonic is a blip that much earlier. The even positions are empty because the clipper is symmetric.

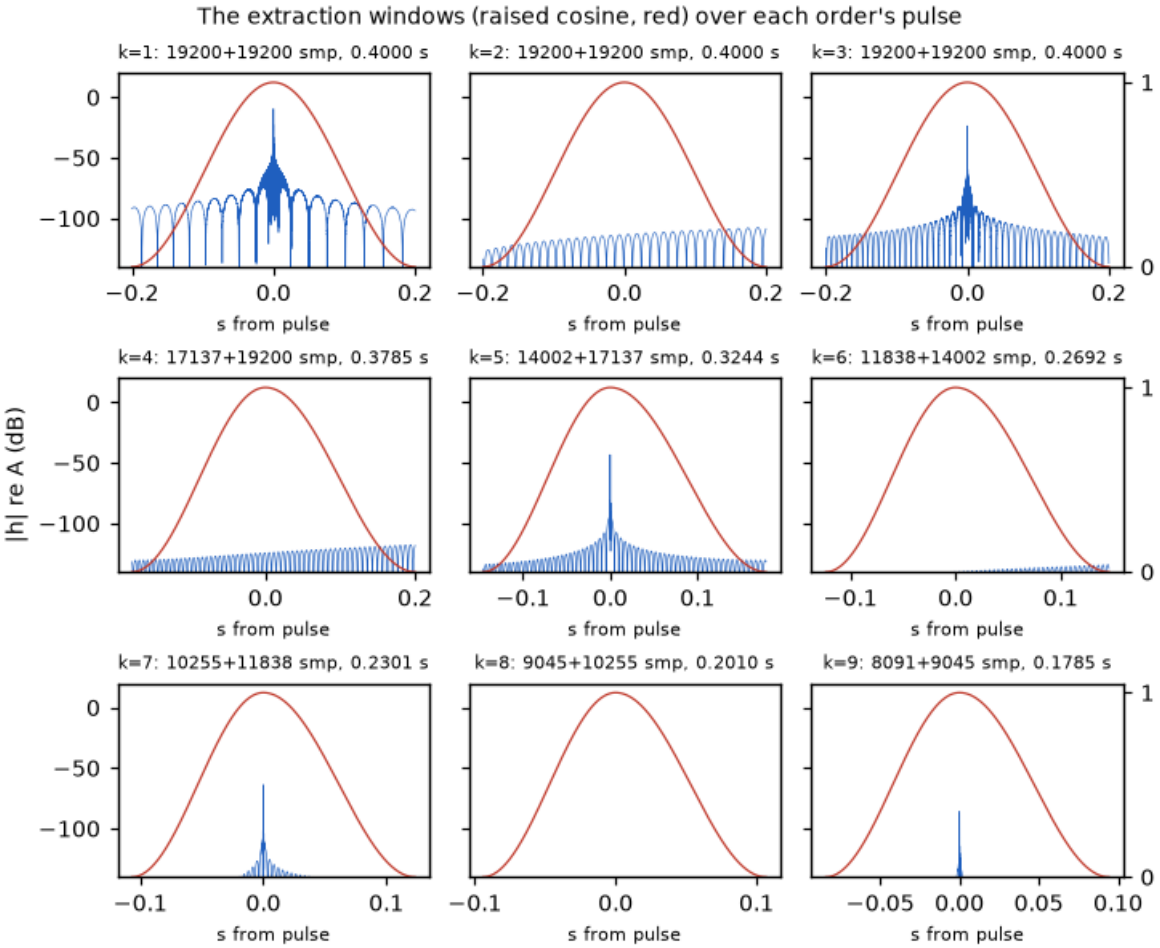



Figure 4: The nine slices (red) over their blips (blue). The first three take the full width; from the fourth on, the gap to the next blip sets the width.

note is read at three times the note's frequency, the ninth at nine times. (The curves are drawn against the note you play, but each point on the ninth-harmonic curve was read nine times higher up.)

**Where a reading stops being accurate.** A digital recording cannot hold a tone above half its sample rate; anything higher folds back down to a lower frequency instead of vanishing (an alias image). For a high enough note, the tenth harmonic — which the pedal produces whether or not the app reads it — folds back onto exactly the frequency where the ninth is being read, and from that note upward the ninth harmonic's reading is contaminated. Every harmonic has such a note, and the higher the harmonic, the lower the note at which its next neighbor lands on it. That note is the harmonic's validity bound, and past it the app draws the curve faded and uses none of the values. In the standard plan at 96 000 Hz, the ninth harmonic is readable up to 5053 Hz (around D#8), the fifth up to 8727 Hz — all far above the top of a guitar's range, so for the notes you actually play every harmonic is inside its bound.

| harmonic | arrives earlier by (s) | slice kept (s) | readable up to (Hz) | nearest note |
|---|---|---|---|---|
| 1 | 0 | 0.4 | 9937.69 | D#9 |
| 2 | 1.109 | 0.4 | 9937.69 | D#9 |
| 3 | 1.758 | 0.4 | 9937.69 | D#9 |
| 4 | 2.218 | 0.3785 | 9937.69 | D#9 |
| 5 | 2.575 | 0.3244 | 8727.27 | C#9 |
| 6 | 2.867 | 0.2692 | 7384.62 | A#8 |
| 7 | 3.113 | 0.2301 | 6 400 | G8 |
| 8 | 3.327 | 0.201 | 5647.06 | F8 |
| 9 | 3.516 | 0.1785 | 5052.63 | D#8 |

### 2.4 THD, in one sentence

The total harmonic distortion figure adds up everything the pedal added at multiples of the note — every harmonic from the second up that lies inside the audible band (20 Hz to 20 000 Hz) and inside its validity bound — and divides the total by the note itself, so that "ten percent" means the added harmonics together are a tenth the size of the note. It is a proportion of the note, not a verdict, and it is computed the same way at any sample rate because the audible band is fixed. It is THD and not THD+N: noise is deliberately not in the sum, which is why the checks in the next section matter — a harmonic slot that holds only hiss would otherwise count as distortion. Toward the top of the neck the sum loses its harmonics one at a time as each leaves the audible band or its validity bound; from the first note where that happens the app draws the curve broken, because it is then counting fewer harmonics, not measuring less distortion. The curve is sampled at 128 notes from 30 Hz upward (the very lowest notes are fade-in territory and are skipped).

### 2.5 Three checks the app makes before it believes a number

**The floor.** Every rig has a measurement floor: a level below which what the app reads is the interface's own noise and distortion, not the pedal's. The app measures it during calibration, with a cable in place of the pedal, and reads it against every harmonic at every note. A reading that sits at or below the floor is reported as a bound — "at most this much" — never as a figure, because nothing quieter than the rig can be seen through the rig. The floor is not one number: it depends on the note, on how loud the sweep was played, and on how many sweeps were averaged (playing the sweep 4 times back to back and combining the results, the app's default, lowers the

noise part of the floor by about 6 dB; eight sweeps by about 9 dB). That lowering applies only where what the rig contributes is noise; at the faded ends of the sweep, where the residue is the measurement's own, averaging buys nothing and the floor stays put. This floor is gathered with the other measurements' floors in Section 9.

**Fundamental presence.** The THD figure, and everything else that compares a harmonic with the note, divides by the note. A few devices — an octave-up fuzz turned all the way up, a ring modulator — remove so much of the note that what comes back at the note's frequency is no longer the pedal but the analysis machinery's own leakage. The app has measured how far below the loudest harmonic a reading of the note can sit before that happens — 76 dB, established on devices whose answers were known in advance — and where a record crosses that line it stops quoting anything that divides by the note (fundamental presence). The harmonic levels themselves are still real and are still shown; the ratio to a note that is not there is not.

**Shaping: circuit or hiss?** The analysis reads whatever landed in each harmonic's slice, and the pedal's own hiss lands there too. For a symmetric clipper the even harmonics are exactly absent, so what the app reads at the even positions *is* its hiss — often well above the rig's floor, and still not distortion. The harmonic shaping check separates the two by looking at how a harmonic's reading behaves across neighboring frequencies. Something a circuit did varies smoothly and consistently from one frequency to the next; noise does not — its readings scatter, each one unrelated to its neighbors. The app computes a coherence score for that orderliness over 32 neighboring readings (pure noise scores about 0.18 on it, a perfectly orderly response scores 1), converts the score into how far the reading stands above the noise in it, and calls the reading *shaped* only when that margin is at least 6 dB. With the score comes a read scatter: how far a single reading might be off purely because of the noise in it, stated in dB alongside the level (a reading that is pure noise carries about 5.6 dB of it). A reading that fails the check is drawn dotted and faded, the legend says "at the floor — noise, not a response", and nothing downstream — the THD sum, the comparison distance, the summaries — treats it as a harmonic.

**Averaging, since it came up.** When the sweep is played more than once, each repeat is analyzed on its own and then the repeats are combined. Before combining, each repeat is lined up with the first to a fraction of a sample (derotation) — a timing drift of even a fraction of a sample between repeats would otherwise blur the high harmonics into looking politer than they are. The combination is of the full readings, direction included, never of the levels alone: averaging levels would make a biased number more precise, not more right. This is sweep averaging; the app offers 1, 2, 4, 8 sweeps and defaults to 4, with a 0.5 s gap between repeats.

### 2.6 What the result looks like

Figure 5 is the worked example — a symmetric soft clipper whose exact answer is known — as the pipeline reads it. Each solid curve is one harmonic's measured level across the notes, in dB relative to the input; the dashed line under each is the exact answer, which the solid curve sits on. Where a curve turns dotted it has passed its validity bound, and the shaded strip at the far right is the fade-out, where the last point reads the fade rather than the device. The even harmonics are not drawn at all: for a symmetric clipper they are exactly absent, and the pipeline reads nothing there but the arithmetic's own dust, far below the chart. The note itself reads above zero because this clipper also adds gain.

The two charts the app draws for a real record are shown next, rendered from the app's own chart code on its simulated pedal (never a screenshot). The first is the Harmonic Distortion chart itself. Each curve is one harmonic; the legend names it and says what it is musically — the second

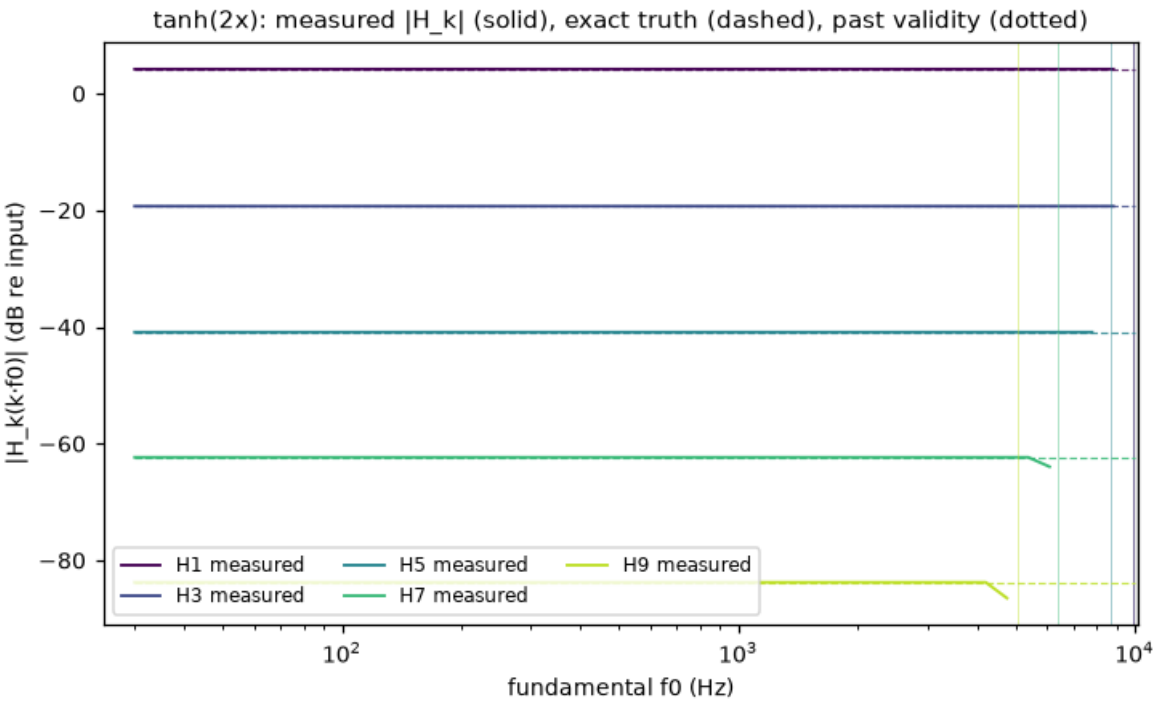



Figure 5: The worked example: measured level of each odd harmonic (solid) against the exact answer (dashed) across every note. Dotted segments lie past the harmonic's validity bound; the shaded strip is the fade-out.

harmonic is an octave up, the third an octave and a fifth, and so on — with the odd family in one color and the even family in another, so the symmetry rule from the first section can be read straight off the chart. The horizontal axis is the note you play, labeled with note names and their frequencies; the vertical axis is the harmonic's level in dB relative to the input, and its label says how the axis was scaled. The shaded band marks a guitar's range. Where a curve fades and breaks near the top it has crossed its validity bound. The lines under the chart state the drive level the sweep was played at and, for a real capture, how many sweeps were averaged — read them before comparing two records, because a harmonic level only means something at a stated loudness.

The second chart is Distortion vs. Note: the THD figure, in percent, across the same notes. A flat curve means the pedal adds the same proportion of harmonics at every note; a curve that falls toward the top of the neck usually means the pedal's own tone filter is shaving the harmonics off. Where the curve turns dashed, the sum has begun losing harmonics to the band edge, one at a time, and the line under the chart names the note at which each one left — from there the curve falls because it counts fewer harmonics, not because the pedal cleaned up. The vertical axis is fitted to the notes before that point, so the dashed tail may run off the bottom of the chart; that is the accurate picture of a sum with fewer terms in it.

## 2.7 How you can check that the app does what this says

Everything above is a description, and a description can drift from the code it describes. The pipeline behind these documents guards against that in a way anyone can rerun. The app's own measurement code writes out the methods manifest: every setting that determines a result, obtained by calling the code that owns it. From that manifest and the published method, a second program was written — in Python, independently, not by translating the app's code — and the parity test runs both on the same made-up devices and demands the same answer.

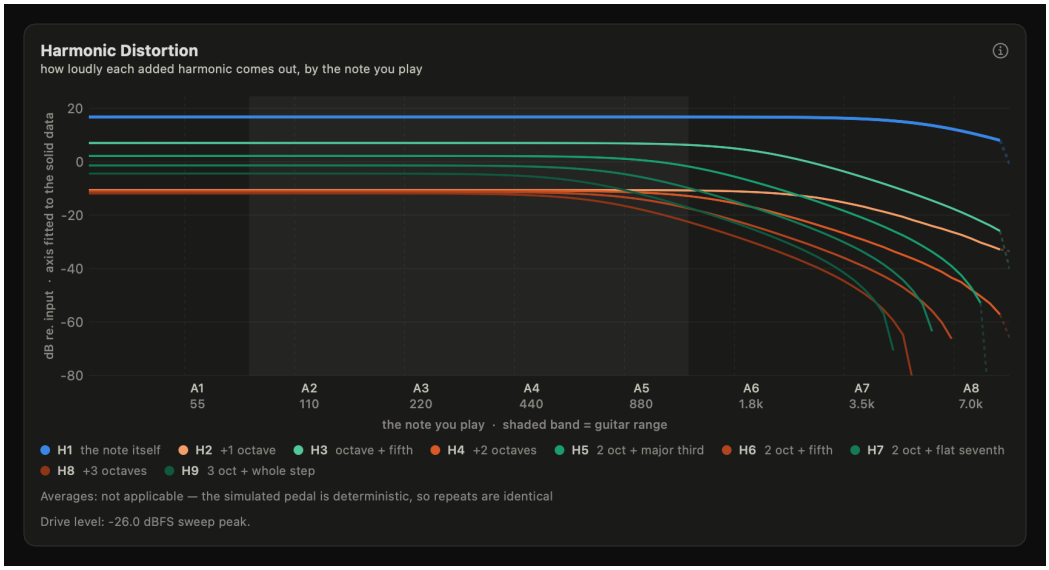



Figure 6: The Harmonic Distortion chart as the app draws it, for its simulated pedal.

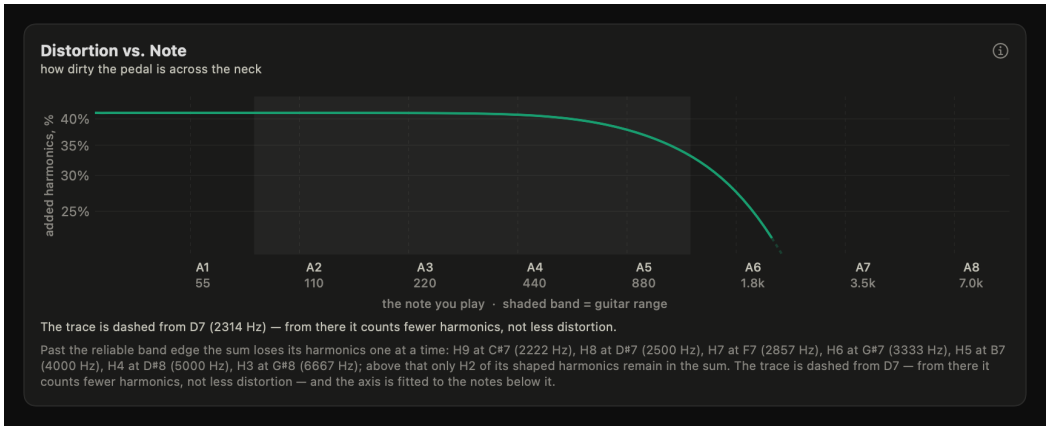



Figure 7: The Distortion vs. Note chart as the app draws it, for the same simulated pedal.

The made-up devices are the point. Three of them are formulas — the soft clipper of the worked example, a hard clipper and a gentle polynomial curve — whose exact harmonic content can be computed on paper, so both programs are held not only to each other but to the truth. The fourth is a phantom capture: a recording built to contain exactly a chosen set of harmonics and nothing else, so the analysis is asked to read back precisely what was put in.

The demands, in numbers the pipeline writes into this document: the two programs must agree to within a hundred-millionth of a decibel (1e-08 dB) at every harmonic and every note (on the run that built this document the worst disagreement was under a ten-billionth of a decibel, $\leq$ 1e-10 dB — the two differ only in the last digits of floating-point arithmetic); on the phantom, both must be within 0.01 dB of the exact answer over the whole range; on the formula devices, within stated tolerances (tanh 0.01 dB, poly 0.01 dB, hardclip 0.02 dB) over the lower half of each harmonic's range — the upper half is where a computed device, unlike a real pedal through a real interface, folds harmonics from above the sample rate back into the band, and that folding is the device's, not the method's. For the worked example the exact answer is 4.2083 dB for the note itself, -19.292 dB for the third harmonic and -83.757 dB for the ninth, and both programs read them back within those tolerances. A fifth case adds noise and averages four sweeps, and requires every reading the shaping check calls shaped to sit within 3 times its own stated scatter of the truth — the check on the check.

The test runs automatically on every change to the app, and the tables and figures in both documents are regenerated on every change and compared with the committed copies: a method change that the description does not follow is a failing build, not a stale page. The scripts, the test and the instructions for running them are in the repository named on the first page.

### 2.8 Where it stops being a measurement

- **The faded ends.** The very first and very last moments of the sweep are ramps, and a reading there is a reading of the ramp. Numbers from those notes describe the measurement, not the pedal; the app excludes them where it can and draws them faded where it cannot.
- **A loud harmonic leaks a little into its neighbors' slices.** How much is measured, not assumed, and the app never claims a harmonic deeper below the loudest one than that leakage allows. Where the note itself is that far down, the fundamental-presence check withdraws every ratio to it.
- **A number can be noise.** A harmonic slice that holds only hiss still yields a number. The shaping check is there to say so, and a reading it rejects must not be read as a harmonic, however confidently it is drawn.
- **The simulated pedal folds its own harmonics.** The app's virtual pedal applies its clipping to the signal sample by sample, so the harmonics it makes above half the sample rate fold back into the band — an error a real pedal through a real interface does not carry. The user guide's Simulation Mode page states it; the parity numbers above measure it, and it is largest in each harmonic's top half-octave.
- **Short sweeps carry fewer harmonics.** Each harmonic needs a slice at least 16 samples wide on each side of its blip; a sweep much shorter than the standard plan runs out of room for the high ones, and they are simply not reported.
- **The range's top point.** The THD curve asks for 128 notes and can deliver 128: its topmost point lands a hair above the sweep's end and is dropped rather than read under the fade.

# 3 Transfer Curve

## 3.1 What this measurement tells you, and what it cannot

The Harmonic Distortion chapter inferred a pedal's clipping from the harmonics it adds. This measurement draws the clipping itself. The app plays one steady low note into the pedal, records what comes back, and plots the output against the input, moment by moment, over one cycle of the note: the transfer curve. A plain wire draws a straight diagonal line — what comes out equals what went in. A clipper draws a line that bends over at the top and the bottom, and the bends are its clipping shape, seen directly rather than deduced.

**Reading a picture of output against input.** Each instant of the recording is one dot: how far across is the input at that instant, how far up is the output. As the note swings up and down, the dot travels out to the right, back through the middle, out to the left and back again, and over a whole cycle it traces a path. If the pedal's output depends only on the input at that same instant — the same input always gives the same output, whether the wave is on its way up or on its way down — the outward and return journeys land on the same path, and the picture is a single line. Figure 8 shows the worked example, a soft clipper with no filter, both ways: the path as the app sees it on the left, and the same input and output cycles laid out against time on the right. The path retraces itself exactly, so the line is the device's clipping formula, and the exact formula drawn over it cannot be told apart from it.

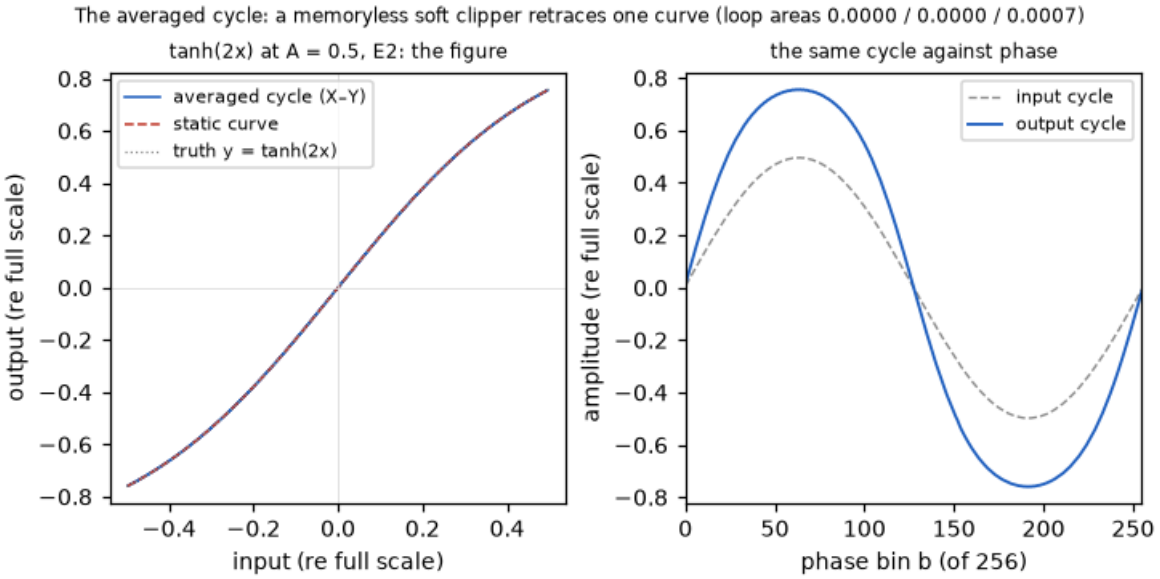



Figure 8: The worked example, a soft clipper with no filter, at the loudest hardware drive and the standard note. Left: the averaged cycle as the app plots it, output against input, with the app's static curve (dashed) and the exact formula (dotted) over it — the three coincide. Right: the same input and output cycles against position in the cycle. The title gives the three loop areas, all zero to the resolution of the slots.

**Why a time shift draws a loop.** Now suppose the output is an undistorted copy of the input that arrives a little late. At any moment the output is what the input was a moment ago, so the same input value now meets two different outputs — one on the way up, another on the way down — and the outward and return journeys separate: the path opens into an oval. Figure 9 builds one. On the left, a sine and a shifted copy of itself; in the middle, the oval the two draw when one is plotted against the other; on the right, what the worked example's clipper followed

by the same shift draws — the bends are the clipper, the opening is the shift. How fat the oval is depends on the size of the shift as a fraction of the cycle: no shift is a line, a quarter of a cycle is a circle. A shift like this is what every tone filter does, because a filter passes each frequency with its own small delay, so a pedal with any tone shaping in it draws an oval even when it adds no distortion at all.

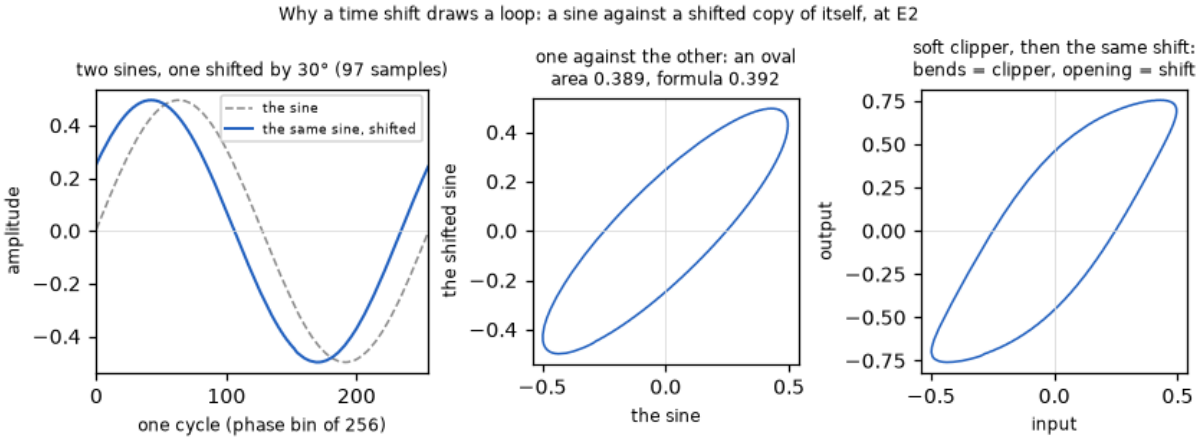



Figure 9: Left: a sine and the same sine shifted in time by a twelfth of a cycle. Middle: the oval the pair draws when one is plotted against the other, with its area as a fraction of the box and the value the formula for a pure shift predicts. Right: the worked example's soft clipper followed by the same shift — the bends are the clipper and the opening is the shift.

**The word "memory".** There is a second way to open the loop, and it is the one this measurement is really after. If the circuit's output depends not only on the input now but on where the signal has just been — a bias point that drifts with the signal, charge stored in a capacitor that leaks back out over the next few cycles — then the same input value again meets different outputs on the way up and on the way down, and the path opens. That is memory, and its loop looks like the filter's oval. Telling the two apart is the whole difficulty of this chapter, and the phase removal below is the app's answer.

The measurement's result is the averaged cycle itself, the line through the middle of it, and three loop areas, each stated as a fraction of the picture's box: the total, the part a tone filter's shift accounts for, and the part left over.

**What it cannot tell you.** Stated once.

1. **Only one loudness.** One note at one drive is one operating point. A pedal whose clipping starts above that drive draws a straight line — not clean, merely not driven into its bends. The Gain Map and the Compression measurements visit other loudnesses.

2. **Whether an open loop is memory or a tone filter.** Ordinary tone filters open the loop with no memory in the circuit. The phase removal exists to take the filter's part out, and even then the middle band of loop sizes is reported with a hedge, for a reason the next section gives.

3. **The clip levels of a lopsided clipper.** Every pedal has a capacitor on its output that removes the wave's average level. It centers a clipper that clips the top harder than the bottom, so the two clip levels come out equal in the picture even when the circuit's are not; the lopsided shape survives as even harmonics in the Harmonic Distortion measurement instead (the limits section).

4. **The rig's own timing error.** If the recording is lined up with the test tone a fraction of a sample late, the picture opens into an oval the measurement cannot tell from the device's own — and it is a property of the rig, not of the pedal (the limits section).

5. **Nothing about listening.** The picture shows the clipping shape. It says nothing about what the distortion sounds like.

Everything in this document describes a pedal that has settled; how a pedal changes over time is being measured next, and gets its own chapters when it arrives.

### 3.2 The test signal

The test tone is a single sine at one note, held steady. The standard note is E2 (82.4 Hz, the low E string), and the standard loudness is -26 dBFS, the same drive the Harmonic Distortion sweep uses, so that the two measurements describe the pedal at one operating point. The tone lasts 1.5 s — about 124 cycles of the note — with 0.2 s of silence before it and 0.2 s after, and its first and last 20 ms are faded rather than switched, so nothing bangs. The first 8 cycles (about 0.097 s) are left out of the analysis while the circuit settles into the tone; everything from there to the end of the tone is used. The app measures the delay through the rig on every capture and lines the recording up with the tone before anything below happens.

The app offers 11 probe notes (E2, A2, E3, A3, E4, A4, E5, A5, E6, A6, C7) and a note-range capture at E2, A3, A4; the drive can be set from -60 to -6 dBFS on hardware (a plugin can be driven to 0 dBFS, digital full scale). The worked example throughout this chapter uses the loudest hardware drive, -6 dBFS — an amplitude of 0.5 of full scale — because at the standard drive the reference clippers the method is checked against barely bend at all. That is the point of the level control: this measurement shows the shape only where the note reaches it.

### 3.3 How the picture is built

Seven steps, each in words; the technical reference gives each one as an equation.

**Folding the cycles together.** The recording holds about 115 usable cycles of the note. Each cycle is divided into 256 equal slots by position within the cycle, every sample goes into its slot, and each slot's samples are averaged — the input's and the output's separately. That gives one averaged cycle: 256 input values and 256 output values, one pair per slot. Noise, which is different in every cycle, averages toward nothing; the shape, which repeats, stays. The right-hand panel of Figure 8 is that cycle, and the left-hand panel is the same 256 pairs plotted against each other. Every sample from the skipped opening to the end of the tone is folded in, the closing fade included — the limits section returns to that.

**The box.** Loop areas need a scale, or a loud signal's loop would read larger than a quiet one's for no reason. Every area is divided by the area of the rectangle the whole picture fits inside, the box normalization, so that an area is a fraction: a loop filling the whole box would read one, and a line reads zero.

**Counting the area lobe by lobe.** A loop can cross itself. A figure-eight is two lobes traced in opposite directions, and the textbook way of measuring an enclosed area, which keeps track of direction, would let the two cancel and report nothing. The app instead cuts the path at every crossing and adds up the size of every lobe regardless of which way it was traced: the unsigned lobe area. A lobe smaller than 0.0002 of its own box is dropped as the jitter of the averaging.

**The oval a tone filter draws.** The best-fitting oval is drawn through the loop using only the note itself — the fundamental of each of the two cycles, nothing above it. That oval is the fundamental ellipse, and its area is exactly what a shift of the note in time would draw: the linear part of the loop, a filter's doing, never memory. Figure 10 shows the worked example's clipper behind a tone filter (a high-pass just below the guitar's range): the raw loop, the oval fitted to it, and what remains.

**What is left.** The oval's output is subtracted from the output cycle, and the loop that remains is measured the same way, lobe by lobe: the nonlinear loop area, the headline memory figure. It is the harmonics' own loop. In a clipper with no memory and no filter the harmonics keep step with the note and retrace; a harmonic that has been shifted — by a filter, or by memory — encloses area here.

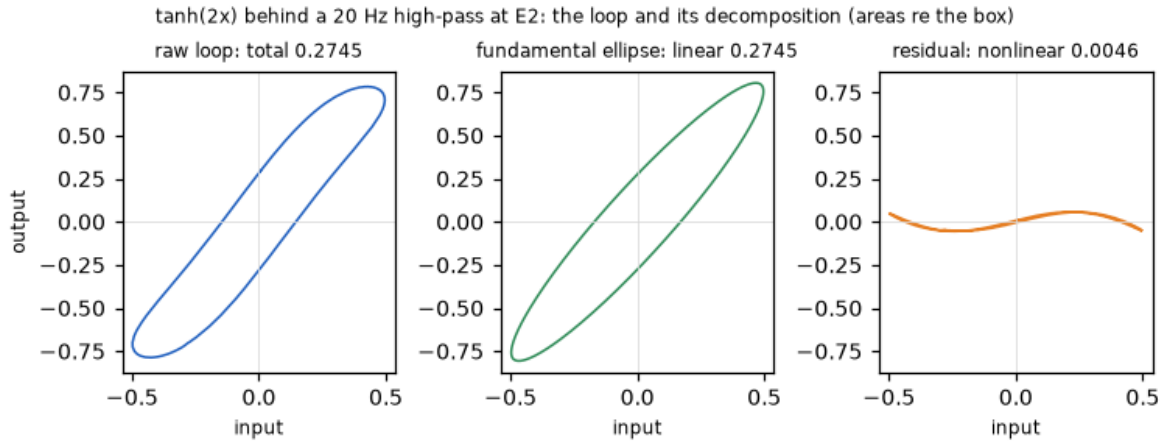



Figure 10: The worked example's soft clipper behind a 20 Hz high-pass at the standard note: the raw loop (left), the oval fitted to its fundamental (middle), and what is left once the oval's output is subtracted (right), each titled with its area as a fraction of the box. The filter draws the oval; what is left is small.

**The line through the middle.** The static curve is the app's estimate of the clipping shape with the loop ignored. The cycle is cut at the input's lowest and highest points into a rising pass and a falling pass; each pass is resampled onto the same 101 evenly spaced input values, and the two are averaged. It is done pass by pass, rather than by sorting every sample by its input value, because a sine spends most of its time near its peaks and hardly any near zero — sorting by input value would starve the middle of the curve.

**The memory verdict.** The memory verdict is the app's three-way call on the nonlinear loop area (of the corrected picture, when the phase removal has been applied): under 5 percent of the box the device reads memoryless-ish, from 15 percent it reads memory, and between the two the reading is hedged, deliberately. The reason is a tone filter placed *ahead* of the clipper. Such a filter shifts the note before it is clipped, and the harmonics the clipper then makes are shifted along with it — the second harmonic by twice the note's shift, the third by three times — so the harmonics no longer keep step with the note, and they enclose area with no memory in the circuit at all. From a single note the app cannot tell that from mild memory, and it says so instead of choosing.

Every one of these quantities is recomputed from the stored cycle each time a record is read; the record keeps the cycle and nothing that a later fix would have to migrate.

### 3.4 Three things the app checks before it believes the picture

All three lean on the pedal's paired Harmonic Distortion record — the sweep measurement of the previous chapter, of the same pedal at the same setting — when one exists. Nothing they decide is stored; each is re-decided from the two records every time.

**The right way up.** A plugin is measured inside the computer, and its recording is lined up with the test tone by finding the best match — which, for a steady tone, can land exactly half a cycle off. The picture then comes out upside down, and for a lopsided clipper worse than that: each input is paired with the output from half a cycle later. The orientation resolution check reads where the note sits in the captured cycle. When it sits within 2.9 degrees of exactly in step or exactly reversed, the alignment is known to have locked to the tone; if the pedal's polarity (next) then disagrees with the way the picture is drawn, the output cycle is moved on by half a period and the picture redrawn. In every other case the picture is shown as captured and nothing is implied. A hardware capture is lined up by a different method that keeps the pedal's true timing, and is immune.

**Polarity.** polarity is whether the pedal turns the wave upside down. It is read from the paired record, over the low notes: does the note come back the same way up or inverted, with the louder readings counting for more. The confidence is how consistently the answer comes out, on a scale where one is perfect agreement, and the reading counts as confident only above 0.35. A device that shifts the note by a quarter of a cycle is neither one nor the other; it reads as not confident rather than as a guess.

**Is the note there?** The memory verdict, the polarity statement and the pairing check below all depend on the note itself being in the recording. A few devices remove it (an octave-up effect, a ring modulator), and what is then read at the note's frequency is the analysis's own leakage. The app takes the fundamental presence reading from the paired record, which separates the note from its harmonics far better than one averaged cycle can (the 76 dB bound of the previous chapter), and where the note is absent it draws the raw picture and states none of the three.

**The phase removal.** This is the longest part of the chapter, and the part a reader most needs.

*The problem.* A tone filter shifts each frequency in time by its own amount. A filter *after* the clipper shifts each harmonic by the filter's own delay at that harmonic's frequency. A filter *before* the clipper shifts the note, and the clipper's harmonics inherit multiples of that shift — two, three, four times it. The two cases look alike in the picture and are corrected differently, and the obvious repair (correct each harmonic by the filter's delay at that harmonic's frequency) is wrong exactly when the filter sits before the clipper — which is where a pedal's tone control usually sits.

*What the paired record knows.* The sweep measurement read each harmonic at its own frequency as the pedal made it, filters before and after included. So the paired record already holds every harmonic's true timing, whichever side of the clipper the filters sit on. The phase removal rests on that: the picture's harmonic timings are compared with the paired record's, and the difference is what the filters did to the picture.

*Lining the two recordings up.* The two measurements were made separately and each measured its own delay through the rig, so a small time offset between them is legitimate, and it has to be found before anything else — treated as a filter shift, it would be "removed" wrongly. The frame fit finds the one offset that makes every harmonic agree at once, louder harmonics counting for more, by scanning every offset within half a cycle. Whatever disagreement remains after that, over the harmonics above the note, is the pairing residual: the test of whether the two records

really describe one device at one setting. Above 0.25 radians (14 degrees of a cycle) the app declines.

*Who votes.* A harmonic takes part only when both records hold it clearly — at least -46 dB relative to the note on each side — and it is inside its validity bound in the paired record. Fewer than 3 voting harmonics is too little evidence, and the app says so rather than fitting through what it has.

*The two allowed positions, and the fold.* In a pedal with no memory and no filtering, every harmonic sits in one of two positions relative to the note: exactly in step with it, or exactly reversed. Which of the two is the device's own doing — a clipper's third harmonic is reversed — and the picture keeps it. That pair of positions is the memoryless lattice, and a measured harmonic's distance from the nearer of its two positions is what the filters added. But "nearer" is ambiguous once a filter has shifted a harmonic more than a quarter of the way round, so the app folds in order. The note itself goes first, anchored by the polarity: inverting means reversed; without a confident polarity the nearest position is taken, and the sign is then quoted only within a stated confidence. Each higher harmonic is folded against a prediction made from the ones already folded — a straight ramp with the harmonic number, which is the shape a filter before the clipper produces, plus the paired record's own low-frequency shape, which is the filter after it. The working assumption is that what a harmonic carries beyond that ramp is under a quarter turn: true of the resistor-and-capacitor networks pedals are built from, and weakest at the lowest harmonic, where it has held on every device class measured.

*The correction.* Once each harmonic's added shift is known, the whole picture is re-timed: a smooth correction across every frequency in the cycle that moves each one in time and makes none louder or quieter. It is measured up to the highest harmonic that voted — the support, that harmonic's number times the note — and extended smoothly above it, and the app's caption states the support and that the correction is extrapolated above it. Figure 11 is the case the obvious repair cannot handle: the soft clipper behind a filter placed before it. The raw picture is a wide loop with the clipping bends on it; the corrected picture is the clipper's own curve, laid over the exact memoryless answer.

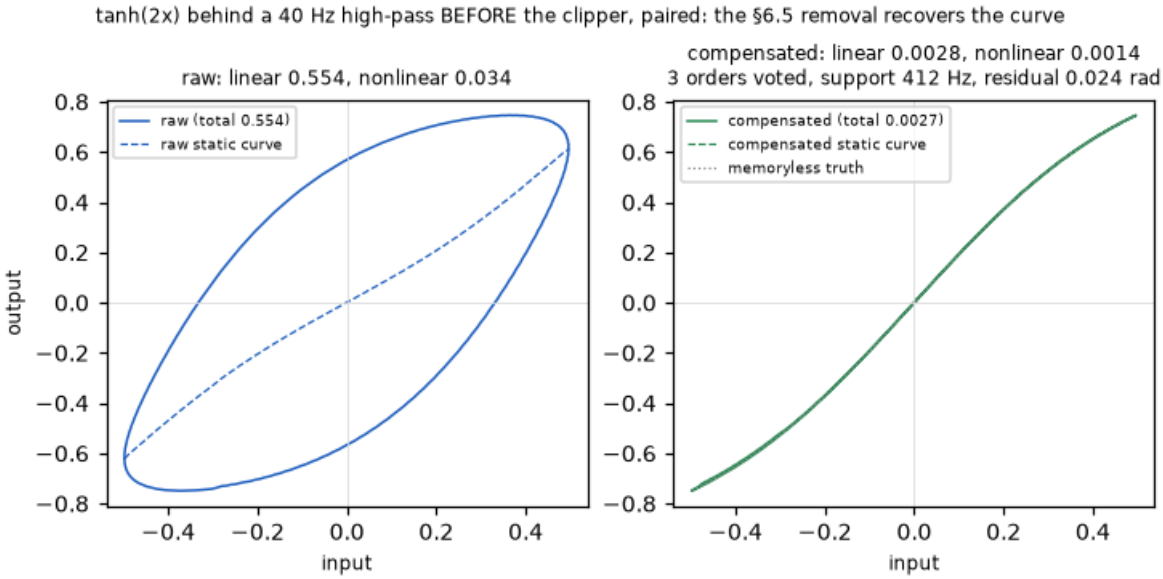



Figure 11: The soft clipper behind a 40 Hz high-pass placed before the clipper, paired with its own sweep measurement. Left: the raw loop and its static curve. Right: the corrected picture over the exact memoryless curve, with how many harmonics voted, the support and the residual.

*Five outcomes.* Corrected; a pairing mismatch, when no offset reconciles the harmonics; too

little evidence, when fewer than 3 harmonics vote; a negligible loop, when the raw total is under 1 percent of the box — there is nothing to correct, and this outranks a mismatch, so a tiny noise-dominated loop never wears a warning (the fit still runs, so polarity can still be stated); and a tripped guard, when a correction that passed the residual nonetheless grew one of the loop areas by more than 0.02 of the box over the raw picture, in which case the raw picture stands. A pairing mismatch has two possible causes the residual cannot separate: the two records come from different settings or sessions, which measuring both again together resolves; or the device is one whose harmonics have no fixed timing relationship to the note (a frequency doubler), which no re-measurement resolves. The app names the class, not the cause.

*Polarity, stated.* The picture states the pedal's polarity only when the fit ran, its residual confirmed the pairing, and the note's position resolved; otherwise the tile stays empty. Absence is honest; a guessed sign is not.

### 3.5 The settings

Every setting is in the technical reference's Transfer Curve chapter, in a table read from the manifest; this chapter repeats only the handful a reader acts on. The standard note is E2 at 82.4 Hz and the standard loudness -26.02 dBFS (an amplitude of 0.05 of full scale); the tone lasts 1.5 s at 96 000 Hz, so one cycle is about 1 165 samples; the first 8 cycles are skipped; the cycle is folded into 256 slots and the static curve is drawn on 101 input values; the memory verdict's two thresholds are 0.05 and 0.15 of the box; and the phase removal declines above a residual of 0.25 radians, needs 3 voting harmonics, and does nothing at all when the raw loop is under 0.01 of the box.

### 3.6 What the result looks like

Figure 8 is the pipeline's own rendering of the worked example. On the left, the averaged cycle plotted output against input, with the static curve and the exact formula over it; the three coincide, which is what a memoryless device with no filter looks like here. On the right, the same input and output cycles side by side against position in the cycle: the output is the input with its peaks pushed in and rounded — the clipping — and taller, because this clipper also adds gain. The title gives the three loop areas, all zero to the resolution of the slots.

The app's own chart, rendered from its chart code on its simulated pedal (never a screenshot), is Figure 12. Four tiles run across the top: the drive note and its frequency; the memory figure — the nonlinear loop area, with the verdict's word under it; the linear phase loop — the oval's area, labeled as filter phase and not memory; and the polarity, with a plain-words gloss. Under the chart's title, the lines state whether the phase removal was applied, through how many of the measured harmonics it was fitted, up to what frequency the correction is measured (extrapolated above), which record it was removed against, and the alignment residual in radians. The chart itself draws the loop in blue and the static curve dashed in orange, labeled as both directions averaged; what goes in runs across, what comes out runs up. For the simulated pedal the loop is closed and the two lines coincide. On a real pedal with an open loop, the dashed line is the app's estimate of the clipping shape through the middle of it, and the limits section says where that estimate is weakest.

### 3.7 How you can check that the app does what this says

**The second program.** As in the previous chapter, a second program was written from the description — in Python, independently, not by translating the app's code — and the parity test runs both on the same made-up devices and demands the same answer. Here the made-up devices come in three kinds. Memoryless clippers with no filter, whose exact curve is their formula, so




Figure 12: The Transfer Curve view as the app draws it, for its simulated pedal, with the phase removal applied.

the picture can be checked against the truth point by point. Linear devices — a wire behind a tone filter, and a wire read with a deliberately planted timing error — whose oval must match the formula for a pure shift. And filtered clippers, with the filter after the clipper or before it, measured by the sweep of the previous chapter as well so that the phase removal can be run and its answer checked against the memoryless truth. That last kind is the one that matters: the removal is held not only to the app but to what the clipper would have drawn with no filter at all. In total the test runs 21 cases.

**The demands.** The numbers here are written into this document by the pipeline. The two programs must agree on every value in the cycle, on the static curve, on the corrected picture, on every area and on every fitted quantity to within a billionth (1e-09; on the run that built this document the worst disagreement was under a ten-billionth, $\leq$1e-10, the last digits of floating-point arithmetic), and must find the same time offset to within a billionth of a sample (the measured difference was $\leq$1e-10); every yes-or-no reading — the outcome class, the flip, the polarity, the verdict, the presence — must be identical.

**Against the truth.** The oval drawn for a linear device must match the formula for a pure shift to within a hundredth at the plan's tone length (measured: at worst 0.73 percent short) and to within a thousandth when the tone is ten times longer (measured 0.05 percent) — the limits section says why the length matters. On a smooth clipper the raw static curve must sit within a hundredth of the output peak of the exact curve (measured three thousandths, 0.003); on a hard clipper within five hundredths (measured about three hundredths, 0.027, because 256 slots round a sharp corner); and a memoryless device's nonlinear loop area must stay under five thousandths of the box (measured two thousandths, 0.002).

**The phase removal.** Every paired filtered device must come out corrected on both sides, with the planted time offset found to within a half of a sample (measured about four tenths, 0.39), a residual under 0.05 radians (measured 0.0237; the app's own accept bar is 0.25), the memoryless curve recovered to within a hundredth of peak (measured about six thousandths, 0.0055), and both corrected loop areas under five thousandths of the box (measured about three thousandths, 0.0028). The removal's parity is established: every paired filtered case compensates on both sides and the two implementations agree on the compensated figure to the Swift-vs-Python tolerance.

**Two results in words.** Two of those results are worth stating in words. The clipper behind a filter placed before it has no memory at all, and its raw nonlinear loop area reads 0.0343 — inside the memory verdict's hedged band; the removal takes it to 0.0014. The hedge in the pipeline section is exactly this case. And the half-cycle lock is undone: on the case built to land half a cycle off, the note's captured position read 3.144 radians, a half turn to within about three thousandths of a radian, the paired polarity read non-inverting, and the resolved picture rises from left to right.

**When it runs.** The test runs automatically on every change to the app, and the tables and figures in both documents are regenerated on every change and compared with the committed copies; the scripts and the instructions for running them are in the repository named on the first page.

### 3.8 Where it stops being a measurement

- **The oval's formula is for a linear device only.** The rule that turns a shift into an oval's area assumes the device's peak is the note's own. A clipped output's peak is not, so on a

clipper the oval's area can read larger than that rule gives. A reader who wants the shift as an angle should apply the rule to a linear-device measurement only, and the pipeline's own check reads the rule to 0.73 percent at the plan's tone length and 0.05 percent at ten times it — so a deficit of several percent on a hardware capture is in the measurement chain, not in the rule.

- **The closing fade is folded in.** The skipped opening cycles protect the start of the tone; the last 20 ms ride their fade into the averaged cycle. Their share of the average shrinks with the tone's length, which is the whole of the deficit above: it is why the pipeline checks the oval at ten times the tone length as well, where the share is gone.

- **A leftover timing error is a loop.** Any timing error left between the tone and the recording — the app removes a whole number of samples, and the fraction of a sample at least stays — shifts the note by an amount that grows with the note's frequency, and draws an oval on a plain wire (Figure 13). On a pedal that residual alignment error loop cannot be told from the pedal's own filter loop. A reader can test it: measure two notes on one rig, and a single timing error must account for both ovals. The nonlinear loop area is untouched by it on a linear device.

- **The output capacitor hides unequal clip levels, not unequal harmonics.** A clipper that clamps at different heights carries an average level, and the coupling capacitor every pedal has on its output strips it, leaving equal heights from an unequal circuit (Figure 14). The shape asymmetry remains, and the even harmonics in the paired sweep measurement are where it is read. Because every pedal has one, a transfer curve essentially never shows the raw asymmetry of an asymmetric clipper.

- **The correction is measured only as high as the last voting harmonic.** A harmonic too quiet to clear the floor on either side does not vote — the soft clipper at this drive seats 3 harmonics, the asymmetric clipper 7 — and above the support the correction is extended, not measured. A hard-clipped device has content far above it.

- **The paired timing is read at the nearest point.** The paired record's timing for a harmonic is read at the nearest frequency it holds, so a read can be off by up to half a step of that grid; the frame fit absorbs it as a spurious offset of a fraction of a sample (measured up to 0.39 samples on the check cases) and leaves the rest in the higher harmonics' residuals.

- **The slots round a corner.** With 256 slots per cycle a hard clip's corner is averaged over one slot's width of input, and the static curve departs from the exact curve there by a few percent of the clip level; the smooth devices read under one percent of peak.

- **The line through an open loop is weakest at the ends.** The static curve averages the rising and falling passes, and the two meet at the input's extremes, which is where the estimate is least trustworthy; on a wide oval it departs from the true line by a few percent of peak there. The app draws it dashed, and that is the reason to read its ends with care.

- **A declined removal names its class, not its cause.** The residual separates a genuine pairing from a mismatch; it does not separate a session mismatch from a structural one, and the two need different remedies (the phase removal, above).

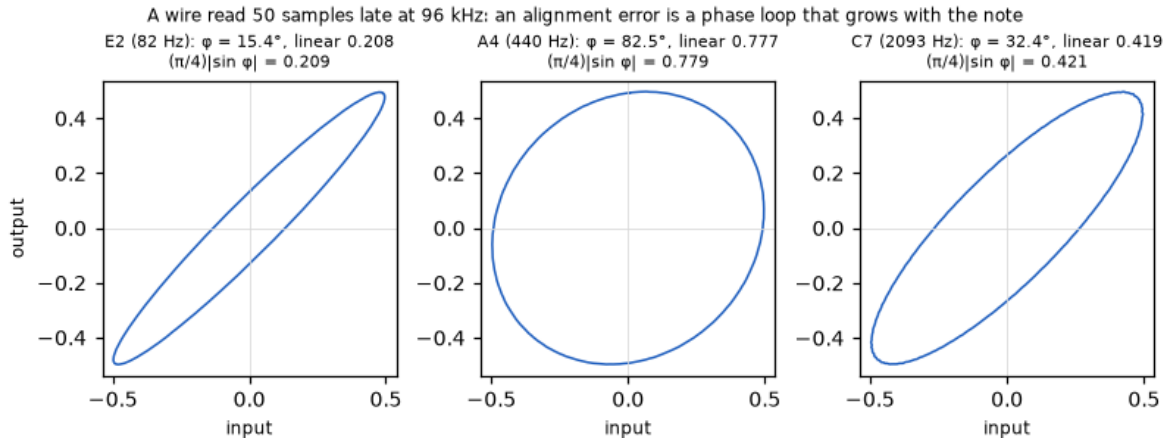



Figure 13: A plain wire, read 50 samples late, at three notes: a timing error is a loop that grows with the note. Each title gives the shift the error amounts to at that note, the measured oval area and the value the formula for a pure shift gives.

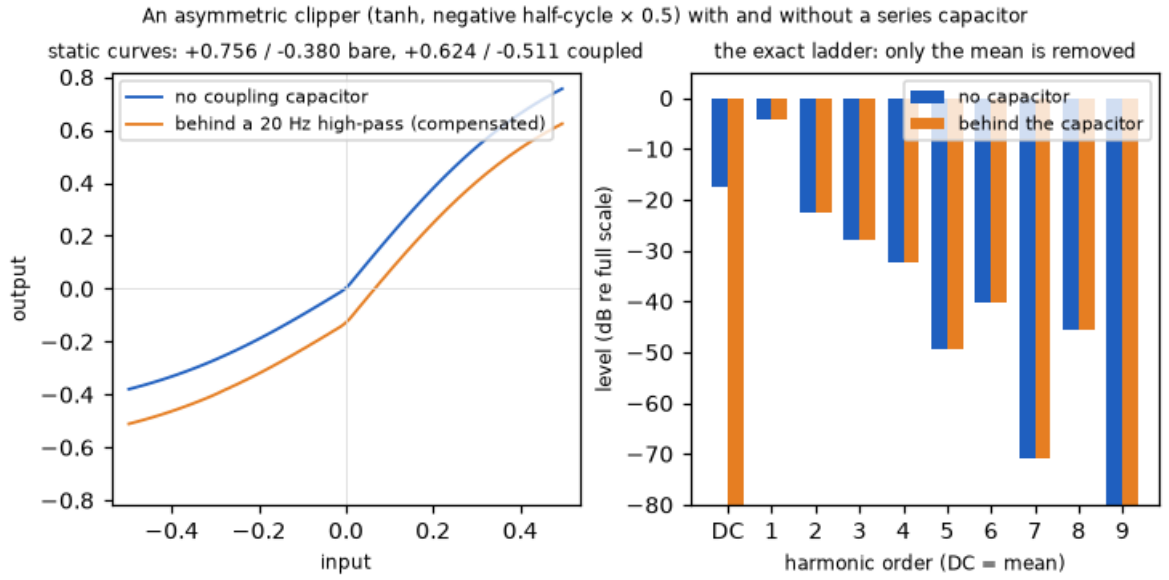



Figure 14: A lopsided clipper (the bottom half scaled to one half) with and without a series capacitor at its output: the static curve's unequal clip levels are centered by the capacitor (left), while the exact harmonic ladder loses only its average level — the even harmonics survive (right).

# 4 Compression

## 4.1 What this measurement tells you, and what it cannot

The Harmonic Distortion measurement listed what a pedal adds to one note at one loudness. This measurement keeps to one note — one pitch, held steady — and changes the loudness instead: the note is played at every rung of a ladder of loudnesses, quietest first, and at each rung the app reads two things, how loud the note came out and how much distortion came with it. The result is the compression curve, the app's "touch response": how much louder the output gets for each step louder at the input.

**A curve of output against input.** The picture the chapter turns on is simple to draw and worth drawing once. Across the bottom is how loud the note went in, up the side is how loud it came out, both in decibels, and each rung of the ladder is one dot. A pedal that does nothing but pass the note draws a straight line at exactly one dB out for every dB in — *one-for-one*, the line every panel in this chapter draws dashed. A pedal with gain draws the same line higher up. A pedal that compresses draws a line that starts one-for-one and then bends over: past some loudness each step louder in produces less than a step louder out, and at the extreme none at all. Figure 15 is the whole idea in one picture, computed from a steep soft clipper (the tanh curve of the earlier chapters with its input multiplied by eight before the bend, as the panel titles print — steeper because the gentler one barely bends inside the ladder, a fact the section on reading a record back returns to). On the left is the test note's loudness over time: the ladder climbing, one rung marked. In the middle is that one rung close up. On the right is the curve the rungs draw, with the two straight lines the app fits through it and the corner where they meet. That corner is the knee, and most of this chapter is about how it is found and when it may be believed.

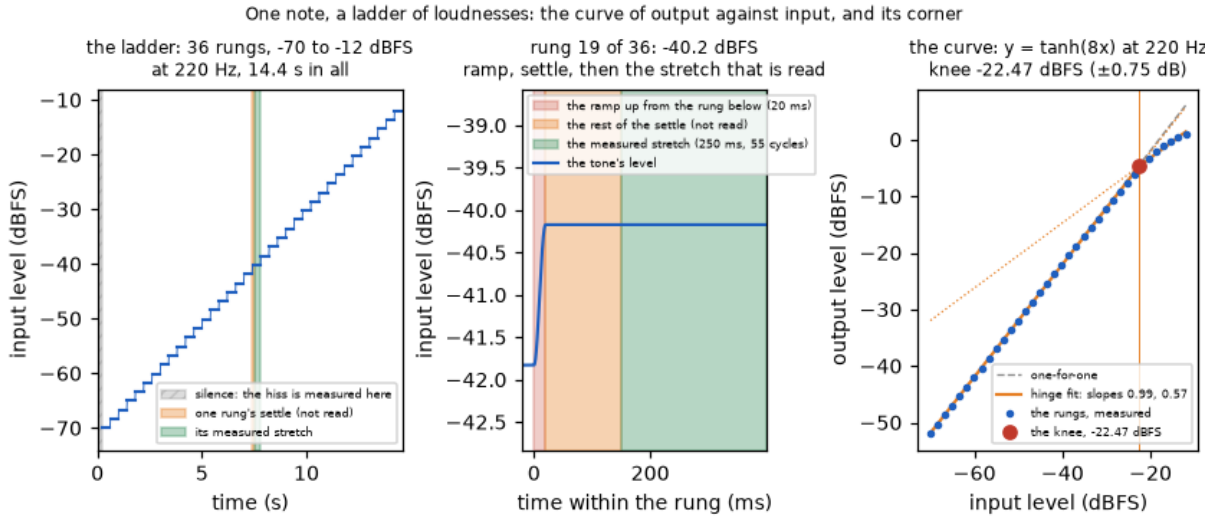



Figure 15: One note, a ladder of loudnesses. Left: the test note's level over time — the silence before it, where the hiss is measured, then the rungs climbing, with one rung's settle (orange) and measured stretch (green) marked. Middle: that rung close up — the short ramp from the rung below, the rest of the settle, then the stretch that is read. Right: output level against input level at the rungs for the steep soft clipper named in the title, the one-for-one line dashed, the two fitted lines of the hinge (dotted where each is extended past the corner) and the corner, which is the knee.

When a record is opened, the app states three things beside the curve, each worked out afresh from the stored rungs: the knee — a loudness, or "at or below" a loudness when the test could not see it; the cleanup reading — where, going up the ladder, the distortion first crosses the line

the chart draws, or that it never does, or that it always is over it; and whether the note itself came through well enough for any of that to be quoted. Each has its own part of the section on reading a record back.

**What it cannot tell you.** Stated once.

1. **Only one note.** The curve is one pitch. A pedal with a tone control ahead of its clipper compresses differently at another pitch, and the Gain Map (Section 5) is where loudness meets pitch.
2. **Nothing between two rungs.** The corner is placed by a fit through all the rungs and may land between two of them, but a feature narrower than a rung — a straight dead region against a very steep soft onset — looks the same when the whole of it falls between two rungs. The fine window below exists for that.
3. **A knee the ladder did not reach.** A pedal already compressing at the quietest rung has its knee somewhere below it, and the honest answer is a bound — "at or below the quietest rung" — never a number.
4. **A distortion reading near the hiss.** Toward the quiet end the note shrinks and the rig's hiss does not, so a distortion figure there can be the hiss. The hiss line described later says where, and each rung is labeled.
5. **The second crossing.** The cleanup level is the first crossing from the quiet end. A pedal whose distortion rises, falls and rises again as it is driven harder — some soft-knee overdrives do this — is quoted at its first crossing only.
6. **The last rung is read under the test's own fade.** The note fades out inside the last rung's measured stretch, so that rung reads a few thousandths of a decibel low; the limits section gives the figure.
7. **The loudness axis is the signal, never a knob.** How loud the note goes in is how hard the circuit is driven, as your pick and your guitar's volume knob drive it. The pedal's own knobs are not touched while a curve is measured; to see what a knob does, you measure one curve per position.
8. **Nothing about listening.** The curve says how the output's level and its distortion follow the input's level. It says nothing about what the distortion sounds like.

### 4.2 The test signal: a ladder of loudnesses

The ladder first. On a hardware rig the note is played at 36 rungs from -70 to -12 dBFS, a span of 58 dB climbed in equal steps of 1.657 dB. Equal steps in decibels are equal *ratios*: each rung is the same proportion louder than the one below, not the same amount. That is the probe grid. Its top rung is held back from full scale on purpose: an interface's input is set so that the pedal's output cannot clip the converter at the loudest drive, and with the input staged that way the top rung is the loudest drive that cannot clip it. Its bottom rung sits far enough under the top that a high-gain pedal's knee is reached rather than bounded — the pedal's own hiss, not the interface, is what limits it, and the gate below measures exactly that. A plugin has no converter to protect, so its ladder runs to the very top, from -80 to 0 dBFS in 48 rungs of 1.702 dB. The quietest rung, the loudest and the count are the user's to change for a run, and a record's ladder is read from its own rungs, never assumed.

**The fine window.** Where a feature is too narrow for rungs 1.657 dB apart to show, the measure sheet adds a fine window: extra rungs 0.5 dB apart over a span you choose, capped at 24 of them, so that a wide span spreads its extra rungs out rather than growing the run. The extra rungs are merged into the ladder and the whole list sorted, so they interleave the ordinary ones; a fine rung that lands on top of an ordinary one is dropped. Over a span from -40 to -34 dBFS the hardware ladder delivers 48 rungs, the ordinary 36 plus 12.

**Within each rung.** The stepped tone is one sine at the probe note — 220 Hz, the A3 an octave above the open A string (the figures in this chapter are at 96 kHz; a run uses its calibration's rate) — whose loudness climbs the ladder without a break in the wave: the same sine runs through the whole test, only its size changes. Each rung has two parts, the settle region and measurement window: first 150 ms in which the tone ramps smoothly up from the rung below over its first 20 ms and the pedal is then left to settle at the new loudness, unread; then the 250 ms that is actually measured. That measured stretch is trimmed to a whole number of cycles of the note — 55 cycles at the probe note, never fewer than 4 — the cycle snap, and the next section says what it buys. A rung is therefore 0.4 s, the hardware ladder's tone 14.4 s, and the middle panel of Figure 15 is one rung as it is played. The tone's final moments fade to nothing over one ramp, and that fade lies *inside* the last rung's measured stretch — the limits section returns to it.

**The silence before, and where the assembly lives.** A hardware or plugin run plays 0.2 s of silence before the tone and 0.2 s after it, 14.8 s in all. The silence before is not padding: the app measures the loudness of whatever comes back during it — the loop's hiss and the pedal's own, in the same recording — and keeps that figure with the record as the noise stamp. Everything below that says "the hiss" means that number. The technical reference notes that this arrangement — the silence, the tone, the stamp, and the delay the recording is cut by — is assembled by the app's measurement service rather than by a single shared function, and that Simulation Mode assembles it differently (no silence, no stamp, a shorter settle); the last section says what that costs.

### 4.3 How the curve is built

Five steps, each in words; the technical reference gives the exact rule for each.

**Cutting the recording to size.** Every signal through a rig comes back a little late — the round trip through the converters and cables. The calibration measured that delay, and each rung's measured stretch is cut out of the recording that much later than it was played, after the silence before the tone. A plugin measures its own delay on every render and hands back a recording with none, so its cut is the silence alone. The delay is never guessed. A rung whose stretch would run past the end of the recording is left out.

**Asking for one exact pitch.** Each cut stretch is read for the note and for its overtones by the one-bin estimator, which does one thing: it asks how much of exactly one pitch the stretch contains. The cycle snap is what makes that question clean. A stretch that holds a whole number of cycles of the note also holds a whole number of cycles of every one of its overtones, so each of them fits the stretch exactly, and a pitch that fits exactly is read at its full size while every other pitch that fits exactly adds up to nothing — the note's own presence does not spill into the reading of its second overtone, or its third. The stretch's two ends are tapered smoothly to nothing before the reading is taken, which costs almost nothing on a pitch that fits and makes the reading forgiving of the two things the snap cannot promise: the tail of the ramp from the rung below, and a cut that lands a sample off. The check at the end of the chapter measures

both. (The Chord IMD measurement reads a whole spectrum the same way, one slot per pitch, and Section 6 explains slots in general; here only a handful of pitches are ever asked for.)

**The gate.** Before a rung is read, its measured stretch's overall level is compared with the hiss. A rung within 6 dB of the hiss — about twice its size — is left out of the curve and counted: the per-step noise gate. Its "output level" would be the hiss and not the note, and a run of such rungs at the quiet end would draw as a flat line, which is what a compressed pedal looks like. The quietest rung that survives is then the bottom of the curve, and any bound reads from there. The record says how many rungs were dropped.

**The note and its overtones.** For each surviving rung the estimator reads the note itself — the fundamental, whose size is the rung's output level — and up to 8 overtones above it, each harmonic order in turn, stopping at the first that would lie above the audible band (20 to 20 000 Hz) or too close to half the sample rate; at the probe note every rate the app offers reads all 8, and only above about 2222 Hz (C#7) does the band's top begin to take the highest away. The rung's total harmonic distortion is the size of those overtones together, compared with the note — the same figure as chapter one's, THD and not THD+N: the hiss is deliberately not in the sum, which is why the hiss line described later is needed.

**What is stored, and what is not.** The rungs are sorted by input level (a fine window's rungs interleave the ordinary ones) and the record keeps the note, the sample rate, each rung's input level, output level and distortion, the noise stamp and the dropped count. Nothing described in the next section is stored: the knee, the cleanup reading and every label are worked out afresh each time the record is opened, so a better rule corrects every curve ever measured.

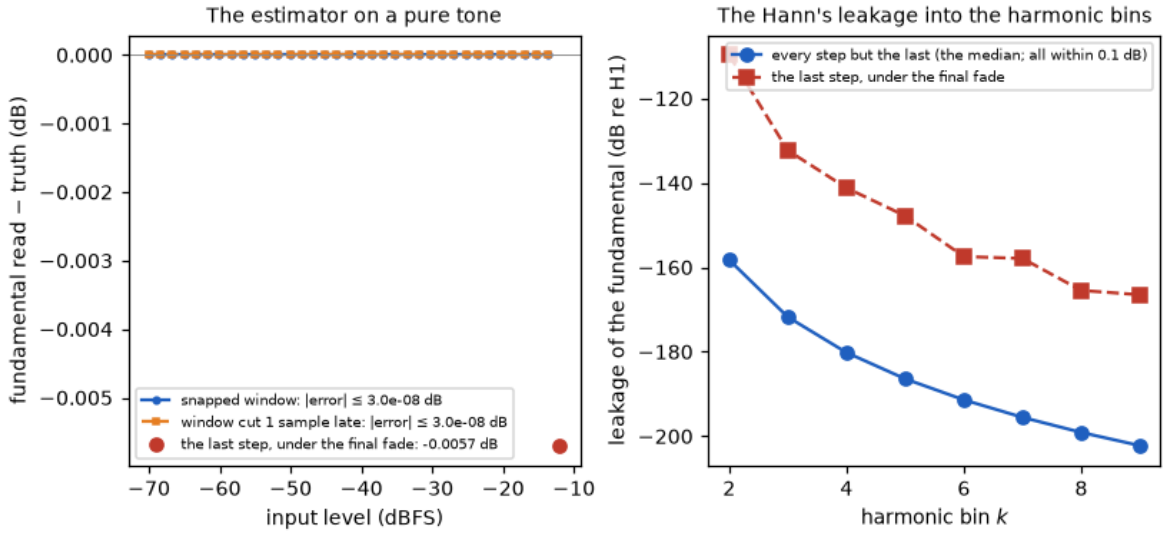



Figure 16: The estimator on a pure tone with nothing else in it. Left: how far its reading of the note sits from the truth at every rung, on the stretch cut exactly and on a stretch cut one sample late — both at the arithmetic's own rounding — and the last rung, which holds the final fade. Right: how much of the note leaks into the reading of each overtone, by overtone, on the cut stretch and under the fade.

## 4.4 What the app decides when it reads a record back

Everything in this section is computed from the stored rungs every time a record is opened.

**The knee.** Between each pair of neighboring rungs the app takes the incremental slope: how many dB louder the output got for that step louder in. One dB out for a dB in is clean; no dB out for a dB in is a wall. Those slopes are the local picture; the knee comes from a fit through the whole curve. The hinge fit draws the curve as two straight lines joined at a corner, and slides the corner along the input axis — through 1 025 candidate positions between the second rung and the third from the top, so that each line keeps at least two rungs — solving for the best pair of lines at each position and keeping the corner where the two lines miss the rungs least. Against that stands the plainest alternative: one straight line through everything. The right-hand panel of Figure 15 shows the fit on the worked example.

*Two kinds of answer.* The corner is reported as a *number* only when the quiet end of the curve is clean, in a stated sense, and otherwise as a *bound*. The app decides in a fixed order.

1. If bending the line does not cut the miss to a half of the single line's or less — if the bend buys almost nothing — the curve is one line. A line at nine tenths of one-for-one or steeper is no knee at all: the pedal is clean throughout. A shallower line is a pedal compressing mildly everywhere, and its knee is "at or below the quietest rung".

2. If the corner's upper line still climbs at three quarters of one-for-one or more, the pedal does not compress in this range: no knee.

3. The unity guard: if the lower line is shallower than nine tenths of one-for-one, no clean stretch anchors the fit. The pedal was already compressing at the quietest rung, the corner the fit found is an artifact of a curve bent everywhere, and the answer is a bound at the quietest rung, with the fitted slope quoted beside it as the evidence. The test is against one-for-one and not against the compression threshold of the previous step, and the difference matters: every slope between the two is a pedal that is compressing, gently, at its quietest rung, and the guard used to let those through as confident numbers.

4. The floor-bound zone: a corner within one rung of the bottom of the ladder cannot be anchored from below — it might lie lower still — and is a bound at that position.

Otherwise the corner is a number, with a resolution band: the range of corner positions that fit the rungs within 5 % of the best. Half that range is the knee's resolution, never finer than the spacing between candidate positions, and it sets the decimals the level is shown with — a tenth of a decibel when the resolution is 0.5 dB or better, a whole decibel when it is not. A bound's resolution is the ladder's own step, because nothing finer is knowable there, and its label reads "≤". Figure 17 shows the two cases as the fit sees them: for each candidate corner, how badly the two lines miss the rungs. A hard clipper whose threshold lies inside the ladder makes a sharp dip with a narrow band around it, and the corner is a number. The same clipper with its threshold below the quietest rung makes a curve that falls toward the bottom with no dip to trust, and the verdict is a bound at the quietest rung by the unity guard. And "no knee" is a real answer, not a failure: the soft clipper of the earlier chapters at its gentler setting bends so little inside the hardware ladder that no corner is found, which is why this chapter's worked example is the steeper one.

**The hiss line.** A distortion reading is a ratio: the overtones compared with the note. The hiss in the recording is a fixed amount, while the note shrinks as the ladder goes down, so at the quiet end the hiss is a larger and larger share of the note — and some of it lands where the overtones are read. The noise bound is how much distortion the hiss alone could fake at each rung, worked out as a deliberately generous worst case: as if every bit of the hiss had landed on the overtones. Anything under that line cannot be told from the rig, and the line rises toward the quiet end. The stamp it is built from is this capture's own, pedal included — a calibration made with nothing in the loop cannot carry the pedal's hiss, so no calibration figure enters here, unlike the sweep

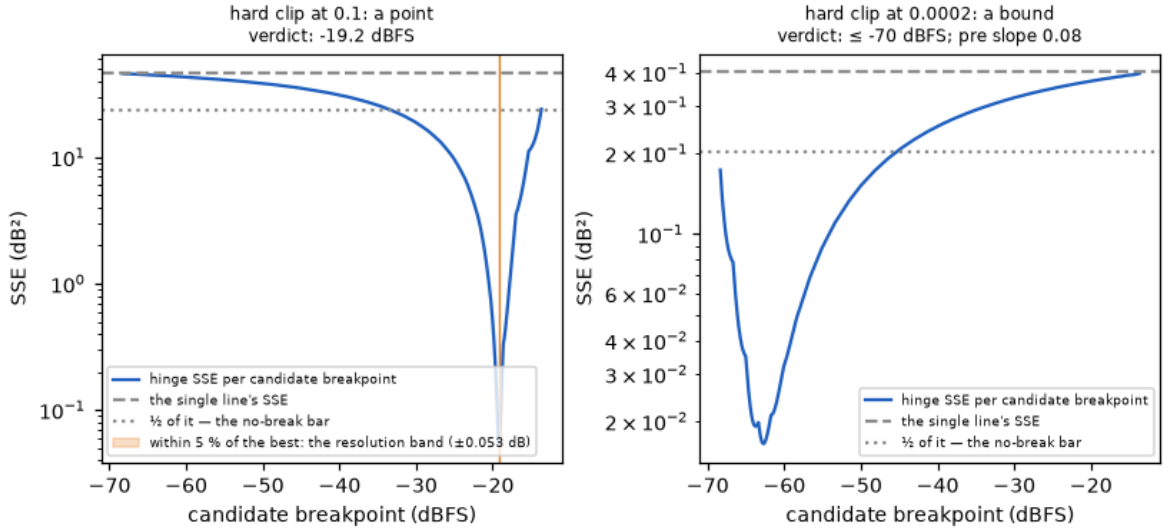



Figure 17: How badly the two lines miss the rungs, for every candidate position of the corner. Left: a hard clipper whose threshold lies inside the ladder — a sharp dip, a narrow band of positions that fit about as well, a number. Right: the same clipper with its threshold under the quietest rung — the curve falls toward the bottom with no dip to trust, and the verdict is a bound at the quietest rung.

measurements of the earlier chapters. This floor is gathered with the other measurements' floors in Section 9.

The hiss line has two possible companions. If the library holds a null-run reference — the loop measured with nothing in it, by this same measurement, under exactly this record's calibration — the app reads from it how much distortion the rig itself *adds* at each loudness. That contribution is an absolute amount, so it is scaled to how loud this pedal's note actually came out: a pedal with gain stands further above the same rig floor, and one that pads the signal stands closer to it. It is read only between the null run's own quietest and loudest rungs, never guessed beyond them. And a plugin has no rig hiss at all — its calibration is numerically silent — so instead the app measured the plugin's own noise before the run, and that operative floor stands in for the hiss. The line drawn on the chart is the *largest* of the bounds that apply, since each one bounds something that is not this pedal's distortion and taking the largest can only be more careful; the floor source label names a companion only where it actually sets the line somewhere, so that a null run sitting under this capture's own hiss everywhere is not what the reader is told they are looking at.

With the line drawn, every rung gets a floor class. At or under the line (within 0.1 dB), the reading *is* the rig. Above it but within 6 dB, the reading is a mix of the pedal and the hiss. Beyond that it is clear: the pedal. The trace is then drawn in floor runs: solid where it is clear, dotted where it is a mix, faint where it is the hiss, each stretch between two rungs taking the more cautious of its two ends' classes; and where the trace would fall under the chart's own bottom edge it is left as a gap, never drawn as a line along the edge that would look like a reading. Figure 18 shows all of it on a plain wire through a little noise and a loop with a little distortion of its own.

**Does it clean up?** Read the distortion from the quietest rung to the loudest. The cleanup reading says where, going up, it first crosses the line of 1 % — a convention for reading the curve, drawn where the app's own code draws it; the value is read out of the app by probing the shipped rule rather than typed. The reading has three states, and the ends of the ladder are never dressed up as a crossing. If the quietest rung is already over the line, the pedal is *never clean* — within

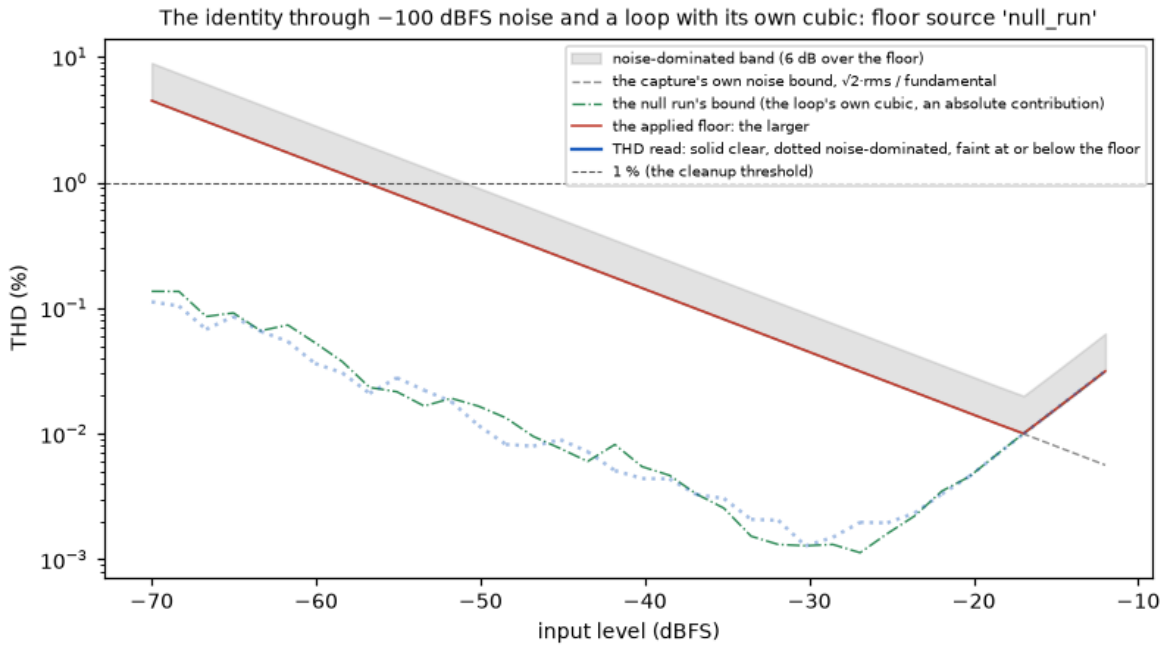



Figure 18: The hiss line on a plain wire through a little noise and a loop that adds a little distortion of its own: the capture's own noise bound rising toward the quiet end, the null run's bound (the loop's own distortion, an absolute amount, defined only inside the rungs it measured), the line drawn as the larger, the band above it where a reading is a mix, and the trace in the chart's three styles — here at or under the line everywhere, since a wire adds nothing of its own.

this ladder's range, the only range it can speak for. If some rung is the first over the line, the pedal *cleans up*, and the level is placed between that rung and the one below in proportion to how far each sits from the line. If no rung reaches the line, the pedal is *always clean*, as far as this ladder goes. Only a genuine crossing quotes a level or draws a marker. Figure 19 shows the three. The Gain Map chapter that follows reads the same three states off each column of its map, by the same rule.

**Did the note come through?** Every distortion reading is a ratio to the note, and a ratio to a note that is not there means nothing. A few devices — an octave-up effect at full tilt, a rectifier — remove so much of the note that what the estimator reads at the note's own pitch is leakage from the louder overtones. Chapter one measured how far below the loudest thing in a reading the note can sit before that happens (76 dB), and each rung is judged by that rule from the one distortion figure it stores: the summed presence reading, which compares the note with all its overtones together rather than with the loudest one, and so is at most 9.03 dB quicker to call the note absent — careful in the direction of quoting less. If most rungs fail, no knee, no cleanup level and no percentage is quoted for the record; the curve's shape is still shown. The check at the end of the chapter runs a made-up device with no note through it.

**The summaries.** The generated summaries take four things from the curve — the knee's level, with a confidence that stays low for a bound; how sharp the bend is, from how many dB of input the slope takes to fall between two levels; the average slope over the top of the ladder; and the cleanup reading, in the same three states — and they are a later chapter's subject. One more app-level fact belongs here: a hardware record whose calibration carries the measured volts factor draws its input axis and quotes its knee and its cleanup level in volts at the pedal's input, and a plugin record's axis is digital full scale — a different physical quantity — so no gap between the two is ever a single number.

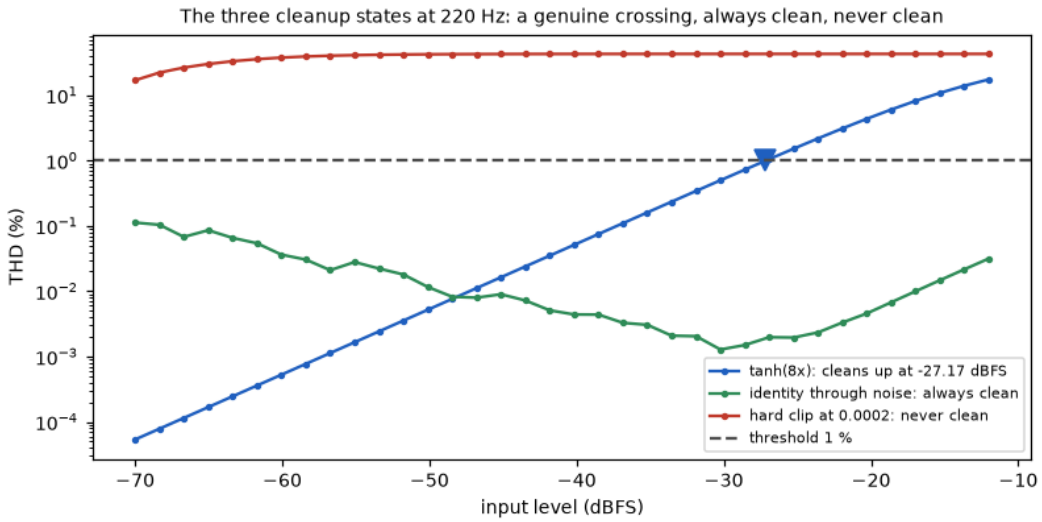



Figure 19: The three cleanup states: the steep soft clipper crossing the line between two rungs (the triangle marks the crossing); a plain wire through noise that never reaches it; and a hard clipper whose threshold lies under the quietest rung, over the line at every rung.

### 4.5 The settings

Every setting is in the technical reference's Compression chapter, in one table read from the manifest. This chapter repeats the handful a reader acts on. The hardware ladder runs from -70 to -12 dBFS in 36 rungs (1.657 dB apart) and the plugin ladder from -80 to 0 dBFS in 48 (1.702 dB apart); a fine window adds rungs 0.5 dB apart, at most 24 of them; the note is 220 Hz; each rung settles for 0.15 s (the first 0.02 s of it the ramp) and is measured for 0.25 s, trimmed to 55 cycles; the silence before the tone is 0.2 s; a rung within 6 dB of the hiss is dropped; the corner's upper line must fall under 0.75 of one-for-one to count as a knee and its lower line must reach 0.9 for the corner to be a number; the cleanup line is 1 %; and a reading is clear of the hiss line at 6 dB above it.

### 4.6 What the result looks like

The app's own chart, rendered from its chart code on its simulated pedal (never a screenshot), is Figure 20. Across the top run three tiles: the probe note, with its frequency; the knee, as a level in dBFS with the app's short gloss under it; and the cleanup level, with its own gloss. Under them the Touch Response chart puts the input level along the bottom — labeled as how hard you hit the strings — and the output level up the side. The measured curve follows the dashed one-for-one reference at the quiet end and then flattens; a dashed vertical line marks where the pedal cleans up and another where the knee sits, each labeled; and a shaded band across the right of the plot is left unlabeled by the chart itself — the user guide's Compression page says what it marks. Below is the companion chart, Distortion vs. Level: the distortion in percent against the same input axis, with the 1 % line dashed across it and labeled. On this record the trace begins part-way across and the line under the chart says why: where the measured distortion falls under the chart's own lowest mark it is left as a gap, because the true values there are lower than anything the axis can draw. This particular pedal is a simulated one, so it carries no noise stamp and the chart draws no hiss line; on a hardware record the line, the band above it and the three trace styles of the previous section appear on this lower chart, and a legend names the floor's source.




Figure 20: The Compression view as the app draws it, for its simulated pedal: the three tiles, the Touch Response chart with its one-for-one reference and its two markers, and the Distortion vs. Level chart with its gap under the axis floor.

## 4.7 How you can check that the app does what this says

**The second program.** As in the earlier chapters, a second program was written from the description — in Python, independently, not by translating the app's code — and the parity test runs both on the same made-up devices and demands the same answer. It runs 24 cases, and two of the device cases a second time as their phantom capture twins (2 twins). Every case is a known device through a known loop: a delay, a stream of noise added at the interface's input, a gain applied after them as an input knob would apply it, and, in some cases, a little distortion of the loop's own, the way a converter adds some. The Python borrows chapter one's devices and noise and the Transfer Curve chapter's tone filters, and owns everything else: this instrument has no sweep in it.

**The known answer.** It is chapter one's again. For a device with no memory, the exact size of the note and of every overtone at any loudness can be computed from the device's formula, so beside every rung the second program prints the exact ladder of sizes at that rung's drive and the exact distortion it implies; and for the read-time half it applies the app's own rules — the hinge fit, the cleanup reading, the hiss line — to those exact rungs, so the knee, the cleanup and every label have exact answers too. A knee has no truth beyond that: it is a summary of a curve, not a property of the device, and the check asks whether the app's summary of the measured curve matches its summary of the exact one.

**The leakage allowance.** One idea is new here: the *leakage allowance*. The estimator's reading of one overtone picks up a trace of its neighbors — a little of the note, a little of the overtone next door — even on a stretch cut exactly, because the taper at the stretch's ends is not perfect. How much depends only on how far apart the two pitches are: -158.2 dB below the leaking pitch at one overtone's distance, -171.8 at two, -202.2 at eight, the same at every loudness. That table is measured first, on a made-up recording containing the note and nothing else, and it is allowed for before any real case is judged: an overtone whose true size sits close to what its neighbors leak into it can read a little off for the estimator's own reasons, that much is subtracted from the disagreement, and what remains is held to the bar. Under the final fade the leakage rises to -109.7 dB, which is why the last rung is left out of every comparison and stated as a limit instead.

**The demands.** The numbers here are written into this document by the pipeline. The two programs must agree at every rung and every overtone to within a ten-millionth of a decibel (1e-07 dB) with no filter in the loop — on the run that built this document the worst disagreement was about three hundred-millionths of a decibel (3.1e-08) — and to within a millionth (1e-06) through a filter (measured about two hundred-millionths, 1.6e-08); every yes-or-no reading — each rung's gate verdict, every floor class, every run, the knee's kind and level and label, the cleanup state, the floor's label with and without a null run, the presence verdict — must be identical. On every noiseless case, at every rung but the last, the app's reading of the note must sit within a millionth of a decibel (1e-06) of the exact size (measured nine ten-millionths, 9e-07); the overtones, once the leakage is allowed for, within two ten-thousandths (0.0002) for the smooth devices (measured about seven hundred-thousandths, 6.9e-05) and for the hard clipper (about eight hundred-thousandths, 8.2e-05 — at the probe note and rate no folded overtone of a computed device lands on a pitch that is read, so the clipper with a kink reads like a smooth one here), and within five thousandths (0.005) through a filter (about two thousandths, 0.0018); and the distortion figure, summed over exactly the overtones the app sums, within a ten-thousandth (0.0001; measured about eight millionths, 7.8e-06). The knee must land within a tenth of a decibel (0.1) of the exact curve's knee (measured about five hundredths, 0.0534 — every knee that is a number lands on the same candidate position on both sides, and the two that differ do so by one candidate step), and the cleanup crossing within a thousandth (0.001; measured about three

ten-thousandths, 0.00034). The noise stamp must read the injected noise within two hundredths of a decibel (0.02; measured about seven thousandths, 0.00744, the estimate's own scatter over a short silence).

**Checks on the rig.** Four checks are about the rig rather than the device. The same device and the same noise through a loop turned up by a stated amount must move every output level by exactly that amount — the departure measured under a ten-billionth of a decibel ($\leq$ 1e-10) — and move the knee's input level by under a ten-billionth ($\leq$ 1e-10; the bar is a billionth, 1e-09), with every floor class unchanged: turning the interface up does not move the knee, which is the point of reading the knee on the input axis. A recording 100 samples late, with the app told the silence plus that delay, must read every rung as the undelayed recording did, to under a ten-billionth of a decibel ($\leq$ 1e-10; bar a billionth). A stretch cut 1 sample late must move the note's reading by no more than a millionth of a decibel (1e-06; measured three hundred-millionths, 3e-08) — the taper makes a one-sample miss invisible, which is what it is there for. And a phantom with no note at all — its note 79.97 dB under its own second overtone — must read absent on most of its rungs (measured 0.833 of them) and get no knee, no cleanup level and no floor label; the same made-up recording measures how well the estimator separates a note from its overtones on a cut stretch, 158.2 dB against the 76 dB chapter one's bound allows, which is why that borrowed bound is safe here by a wide margin. Part B's parity is established: on every read-time case — the identity through noise with and without a null run, the null run through a loop with its own distortion, the exact sub-unity lines, the clippers at and under the floor, the fine window, the plugin window, the stated floor, and the device with no fundamental — the two implementations agree on every bound, every class, every run, every knee field, every cleanup state and level, every floor label and every verdict.

**Two results in words.** Two results are worth stating in words. The worked example, the steep soft clipper with no noise on the hardware ladder: its knee is a number at -22.469 dBFS, resolution $\pm$0.748 dB, displayed as "-22 dBFS", with a lower line of slope 0.9936 and an upper line of 0.5749, against the exact curve's -22.469 (a difference of under a ten-billionth of a decibel, $\leq$ 1e-10); its cleanup crossing is -27.175 dBFS against the exact curve's -27.175 (a difference of about a millionth, 1.3e-06). And the hard clipper: its flat top begins at -20 dBFS, and the fitted corner sits at -19.16 dBFS, about eight tenths of a decibel above where the clipping starts (0.842), with an upper slope of 0.157. That gap is a measured relation between a fit's corner and a clipper's onset, not a claim about either: two straight lines through a curve that turns a corner gradually will place their join a little past the first departure from one-for-one. With a fine window laid over that knee the ladder delivers 52 rungs, the knee reads -19.21 dBFS and moves by about five hundredths of a decibel (0.0534), its resolution unchanged, because on both ladders the band of corners that fit about as well is narrower than one candidate step, and the candidate spacing is then what the fit resolves.

**Cases built for the reading rules.** Three cases are constructed to be exactly what the reading rules are for. An exact line that compresses a little everywhere and then breaks reads "$\leq$ -70 dBFS" with fitted slopes 0.85 and 0.2: the unity guard, doing its job. Simulation Mode's own assembly on the worked example — fewer rungs, shorter settles, no silence before the tone — reads its knee at -21.19 dBFS, resolution $\pm$0.388, 1.28 dB from the hardware assembly's knee on the same device: what interactive quality costs. And the exact line that compresses a little everywhere and never breaks is the finding the limits section states.

**When it runs.** The test runs automatically on every change to the app, and the tables and figures in both documents are regenerated on every change and compared with the committed

copies; the scripts and the instructions for running them are in the repository named on the first page.

### 4.8 Where it stops being a measurement

- **The last rung is read under the test's own fade.** The note fades out over one ramp, and the last rung's measured stretch ends where the note ends, so it holds the fade: the last rung's note reads -0.0057 to -0.0017 dB relative to its exact size on the check's devices — a few thousandths low — and the estimator's leakage there rises from -158.2 to -109.7 dB. That is a fact about the test signal, not about any pedal. Moving the fade past the stretch would change the signal every existing record was measured with, and is reported rather than made.

- **The no-bend test can be decided by that fade.** The first of the four decisions compares how badly two lines miss the rungs with how badly one line does, as a ratio, and the ratio has no scale of its own. On a curve that is one exact straight line, compressing a little everywhere, the only rung off the line is the last one under the fade; the two-line fit absorbs that one rung, the ratio reads "a bend", and the second decision then reads "no knee" — where the same exact line with no fade reads "at or below the quietest rung". Measured on this build: the exact line reads "none" on both programs while its exact curve reads "$\leq$ -70 dBFS" with a slope of 0.85. A measured pedal rarely draws an exact line, and any real curvature outweighs one bent rung, so the case is stated as one the fit decides on structure the test cannot see, and left as a finding.

- **A rung skipped because its stretch ran past the recording is silent.** It leaves no point and no count, unlike a rung the gate dropped; a short recording simply yields a shorter curve.

- **Simulation Mode builds its test differently.** No silence before the note, so no noise stamp and no hiss line; fewer rungs and shorter settles at its quick quality (24 rungs, 0.05 s to settle and 0.1 s measured, at 48 kHz, against the hardware assembly's 36, 0.15 s and 0.25 s, at 96 kHz in the check); its full quality uses the hardware assembly's rungs and settles. On the worked example the quick quality's knee sits 1.28 dB from the hardware assembly's, so a simulated record and a measured one agree on the shape of a curve and not on the last decibel of its knee, and a comparison between the two is read that way.

- **A soft onset is not disclosed.** The unity guard tells a clean anchor from a missing one; it does not tell a wide, gradual bend from a sharp corner. A pedal whose bottom decade is clean and whose bend then spreads over many dB gets a knee that is a number and is accurate as far as the fit goes; how wide the bend is would be a new reading, not a tighter one.

- **The quiet end is the rig's.** Under the hiss line a reading is the rig, and the rungs say so one by one; a rise in distortion toward the quiet end is the hiss's growing share of a shrinking note, not the pedal, unless it stands clear of the band.

- **The loudness axis is the signal, never a knob.** How loud the note goes in is how hard the circuit is driven; the pedal's knobs stay where you set them, and a knob is measured one curve per position.

- **One note.** The curve is one pitch; a pedal with a tone network ahead of its clipper compresses differently at another, and the Gain Map (Section 5) is the cross-check across notes.

- **The simulated pedal folds its own overtones.** A device computed sample by sample makes overtones above half the sample rate that fold back into the band. At the probe note and rate none of them lands on a pitch that is read, and the check's phantom twins are the fold-free control; at another note or rate the highest overtones of a computed device with a kink in it would not be the exact answer there.

# 5 Gain Map

## 5.1 What this measurement tells you, and what it cannot

The Harmonic Distortion measurement of Section 2 asks one question at one loudness: play a note into the pedal at a stated level, and list the harmonics that come out. But a distortion pedal is not one thing at all loudnesses. Play it softly and it may pass the note almost untouched; dig in and it clips; somewhere between the two it changes. A gain map asks chapter one's question again at each rung of a ladder of loudnesses — one sweep per rung, each analyzed on its own — and draws the answers as a map: how hard the pedal is driven up one side, which note you play along the other, and one square per loudness and note holding the distortion the pedal made there.

**Loudness as an axis.** The new axis needs a sentence of its own, because it is easy to mistake for a knob. The drive level of a row is how hard the app drives the pedal — the level of the test signal going into it, as your pick and your guitar's volume knob would drive it. The pedal's own knobs are never touched while a map is measured; they stay wherever you set them, and that setting is recorded with the measurement. So the vertical axis is not the drive knob. It is the thing the drive knob is turned to suit: the signal arriving at the circuit. To map what a knob does, you measure one map per knob position. The rungs of the ladder are the level lattice, and the quietest and loudest rungs are its drive window.

Figure 21 is the whole idea in one picture, computed from the worked example of the earlier chapters — the soft clipper of the tanh curve — through the pipeline described later. On the left is one cell: one note, one loudness, and the ladder of harmonics the analysis reads there, exactly chapter one's picture at one point of the range. In the middle is the same note at every rung of the ladder, the ladder of harmonics growing rung by rung as the pedal is driven harder. On the right is the whole map, with that column and that cell outlined. A cell is one measurement; a column is the rungs of one note stacked; a map is the columns side by side. Everything the rest of this chapter says is about how those cells are filled in and which of them may be believed.

When a map is opened, the app states three things beside it, each worked out afresh from the stored squares: the peak tile, the most distortion the pedal clearly made anywhere on the map, or a bound when nothing clearly stands out; the cleanup reading, where, going up the ladder at one note, the pedal starts to distort; and the fundamental presence verdict, whether the note itself survived well enough for any ratio to it to mean anything. Each has its own section below.

**What it cannot tell you.** Stated once.

1. **Nothing about the pedal's own volume or gain.** Every square is a ratio: the harmonics compared with the note itself, in that square. A pedal that is loud and clean and one that is quiet and clean draw the same square. And the vertical axis is the signal driving the circuit, never a knob on it.

2. **Nothing between two rungs.** Each rung is its own capture, seconds apart from the next. A pedal whose behavior drifts or sags between them — a dying battery, a warming transistor — shows the drift as a wobble from row to row, which the map draws and does not name.

3. **Whether a square near the floor is the pedal or the rig.** Below the margin described later, a square is quoted as a bound — "no more than this" — never as a figure.

4. **Nothing in the two fade columns.** The first and last notes of the map sit inside the sweep's fade-in and fade-out, and what they show is the test signal, not the pedal.

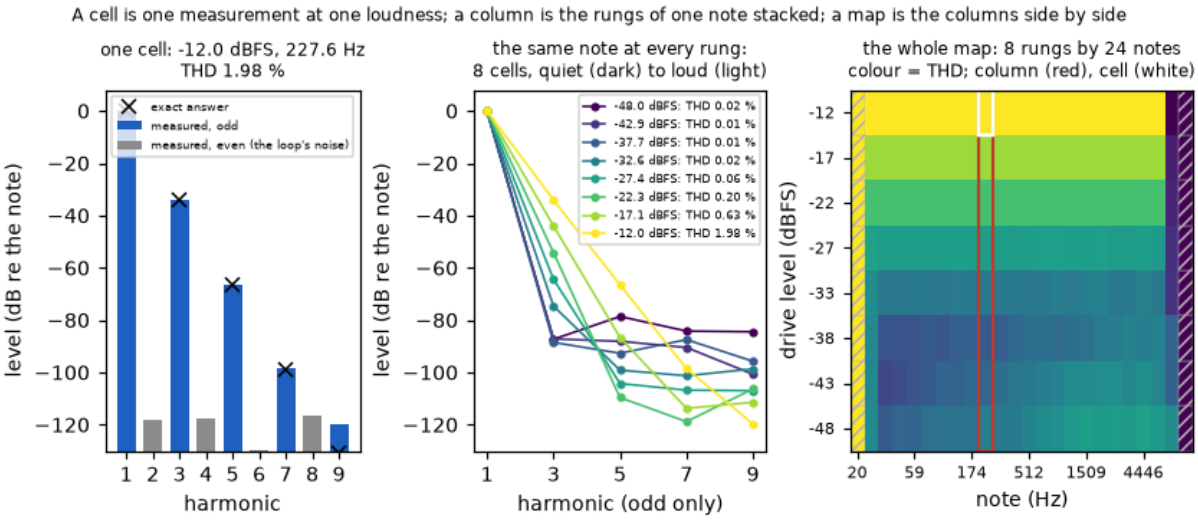



Figure 21: A map is a stack of single-loudness measurements. Left: one cell of the worked example — the soft clipper at the loudest rung of the hardware ladder, at the note the cleanup tile reads — as the measured level of each harmonic relative to the note (blue for the odd harmonics, gray for the even ones, which this clipper does not make and which read the loop's own noise), with the exact answer marked by a cross; the ninth harmonic's exact answer lies under the loop's noise, just below the pane. Middle: the same note at all 8 rungs, quiet to loud, each rung's THD in the legend. Right: the whole map, THD as color, the fade columns hatched, the column outlined in red and the cell in white.

5. **Nothing a short column lost.** Near the top of the map some of a note's harmonics lie too high to be counted reliably, and that column's distortion figure is known to be short.

6. **Nothing about listening.** The map says how much distortion there is at each loudness and note. It says nothing about what the distortion sounds like.

## 5.2 The test signal, and the ladder of loudnesses

The ladder first. On a hardware rig the map is measured at 8 rungs, from -48 to -12 dBFS, a span of 36 dB climbed in equal steps of 5.143 dB. Equal steps in decibels are equal *ratios*, not equal amounts: each rung is the same proportion louder than the one below. The top rung is held back from full scale on purpose. An interface's input has to be set so that the pedal's output never clips the converter at the loudest drive; with the input staged that way, the top rung is the loudest drive that cannot clip it, and the ladder reaches down a fixed span below. A plugin has no converter to protect — it is arithmetic on numbers — so its ladder runs to the very top, from -60 to 0 dBFS in steps of 8.571 dB, its quietest rung lower still, down where high-gain plugin amplifiers were measured to start bending.

At each rung the test signal is the same gliding tone as chapter one's — the synchronized swept sine, from 20 to 10 000 Hz, with the same fades — at that rung's loudness. There is one difference that matters later: by default each rung's sweep lasts 5 seconds, half the 10 seconds of chapter one's, because 8 sweeps take 8 times as long; the measure sheet offers 2, 5 or 10 seconds, and the length chosen is not written into the record. Each sweep is played after 0.3 s of silence and followed by 0.5 s more, and the analysis that reads it is also half as fine as chapter one's. A shorter sweep is a coarser instrument: the analysis's own small residue — the distortion it reports with nothing in the loop — is larger for it, and a later section says by how much.

The notes the map is drawn at are chosen once, as a grid: 24 of them, spaced by equal musical intervals — 4.7 semitones, 0.39 of an octave — from the lowest note the sweep plays to the

highest. That is the export grid. Its first and last columns sit exactly at the sweep's two ends, inside the fades, which is why those two columns are masked, as the next section explains.

The rungs are played and recorded one after another, and each recording is analyzed entirely on its own, as if it were the only sweep. Nothing about one rung leaks into another. Figure 22 is what the measurement produces: the worked example's map, computed by the pipeline through a loop with a little noise in it, colored by distortion and marked with the tiles the later sections explain.

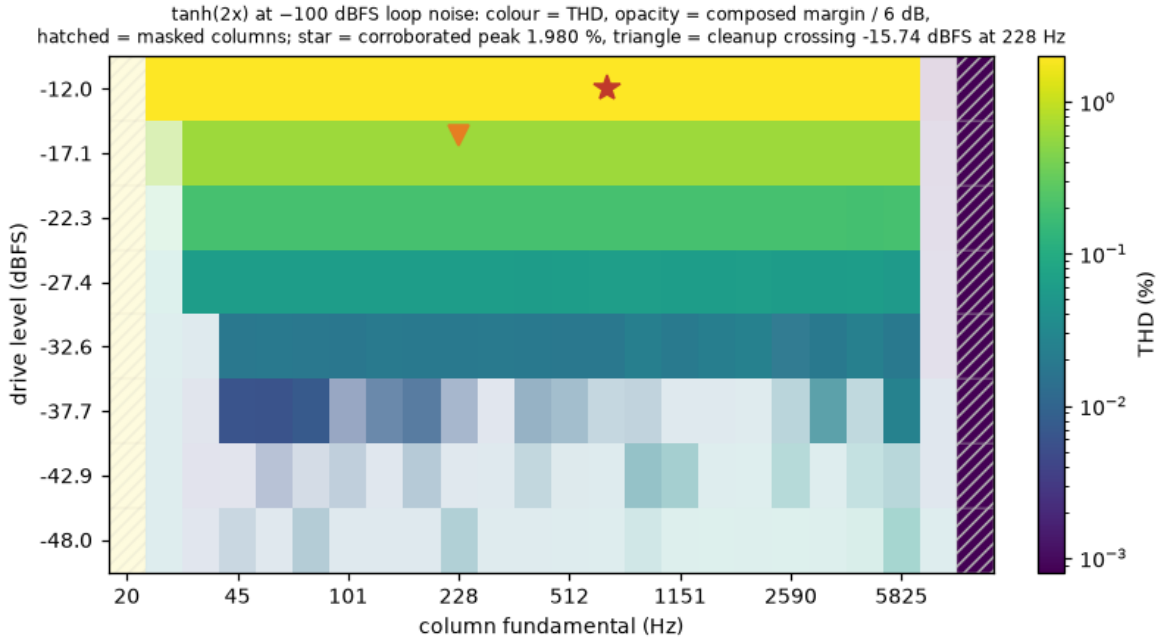



Figure 22: The worked example's map: the soft clipper on the hardware ladder through a loop with a little noise. Color is THD; a square is drawn fainter the less clearly it stands above the floor described in the next section; the two fade columns are hatched; the star is the peak the tile quotes and the triangle is where the cleanup reading places the crossing at the tile's note.

## 5.3 How the map is built

Five steps, each in words; the technical reference gives the exact rule for each.

**Cutting each recording to size.** Every signal through a rig comes back a little late — the round trip through the converters and cables. The calibration measured that delay, and each rung's recording is cut at the moment the sweep is known to have arrived: the silence before it dropped, the sweep and the silence after it kept, and never a sample past the recording's end. A plugin measures its own delay on every render and hands back a recording with none. The delay is never guessed.

**Reading each rung as chapter one does.** Each cut recording goes through exactly the analysis of Section 2: the glide undone, each harmonic's blip cut out with its own window, and every harmonic read against frequency. The app reads 9 harmonics at each rung, and the note's level along with them. Nothing is corrected for the rig here: the loop's own gain and tone stay in every reading, so a square's note level is the level the note actually reached at the interface, with everything the loop added — its delivered fundamental. That number is a level, not a reference, and the floor's arithmetic in the next section leans on it.

**Filling the squares.** For each rung and each note of the grid, the app reads the total harmonic distortion over the audible band (20 to 20 000 Hz), exactly as chapter one defines it, and the level of each harmonic order at that note. A harmonic that would lie above half the sample rate, or a note at which the fundamental reads nothing, leaves its square empty rather than wrong.

**Masking what the test made.** Two columns are marked as the test's own: the first and the last, which sit within a quarter tone (50 cents) of the sweep's two ends and were therefore excited while the tone was still fading in or already fading out. A plain wire shows the same readings there, so they describe the measurement and not the pedal. They wear an edge mask: drawn faint, kept out of every headline number, and never allowed to set the color scale. A third kind of column is marked near the top of the map: a truncated column, where some harmonic that the audible band would count lies past its validity bound, so the column's distortion figure is known to be short. It is flagged and kept out of the tiles, never corrected.

**What is stored, and what is not.** The record keeps the grid of numbers — every square's THD and harmonic levels — the rungs, the harmonics read, the sample rate and the band, and beside them one audition model per rung, a separate model of the pedal at that loudness that only the listening panels use. Nothing described in the next section is stored: every verdict below is worked out afresh each time the record is opened, so a better rule corrects every map ever made. A record from before the sample rate and the band were stored derives no masks and no residue, and its squares summed every harmonic below half the sample rate; the app treats it as the older reading it is.

### 5.4 What the app decides when it reads a record back

**A floor, and a margin.** Everything the app says about a map rests on one question, asked of every square: how far above the largest thing that could fake it does this square's distortion stand? A square is called a measurement only while it stands clear of *every* fake at once; below that it is a bound. The distance is the square's composed margin, and the things that could fake it are three. This floor is gathered with the other measurements' floors in Section 9.

The first is the loop's own noise. During calibration the app played its sweep through the rig with a cable in place of the pedal and kept a note of how much distortion and noise the bare loop showed at each pitch — the calibration stamp, stored inside every record made with that calibration. It is a single reading at each pitch, and a single reading can be unlucky, so the map reads it through a median read: at each pitch the stamp is replaced by the middle value of 7 neighboring points on its own grid, 3 to either side, a quarter of an octave (0.25) each way. That is wide enough that one unlucky trough cannot make the loop look quieter than it is, and about as wide in all as one column of the map, so a feature a column can resolve is not smoothed away. Then comes an allowance, the drive term, for the fact that the calibration was checked at one loudness (-26 dBFS) and the map's squares are at many. Noise is a fixed amount, not a fixed share: the same hiss is a bigger share of a quieter note. So a square whose note arrived quieter than the calibration's did is given a floor that much higher. A square whose note arrived louder gets no allowance in the other direction — the floor is never pushed lower than the calibration measured, because the rig's own distortion grows with drive and the calibration measured noise and distortion together at one level. The comparison uses the delivered fundamental, the level the note actually reached, and the calibration's own level plus the gain the loop was measured to add; leave that gain out and every margin comes out better than it is.

The second, on a plugin, is the plugin's own noise. A plugin's calibration is a loop with nothing in it and no converters, numerically silent, so it measures nothing the plugin does; instead the

app measured the plugin's own noise before the run, and that operative floor stands in for the loop noise a hardware rig would have.

The third is the one a noise floor cannot see: the analyzer residue, the distortion the analysis itself reports with nothing at all in the loop. Take the sweep, hand it straight back as its own recording, and analyze it: a perfectly clean signal should read no distortion, and almost everywhere it does. But at the first note after the masked column, the sweep's fade-in ramp leaks into the window that reads the second harmonic, and the analysis reports a little distortion that is entirely its own. It is a fixed share of the note, not a fixed amount — it is the same at every rung — which is exactly why a noise floor cannot bound it. It shrinks as the sweep gets longer: at 26.2 Hz it reads -70.11 dB below the note on the default sweep, -46.32 on the shortest and -86.67 on the longest, and one column higher it has already fallen to -103.6, under any real rig's noise. The app works the residue out afresh for each map's own grid. Because the sweep's length is not written in the record, it assumes the worst — the shortest sweep the sheet offers, so the first unmasked column is floored at -46.32 dB on every record — a claim it can make about any record, at the price the last section states. The left panel of Figure 23 shows the residue at every column for the three sweep lengths; the right panel is the parity's own instrument, explained later.

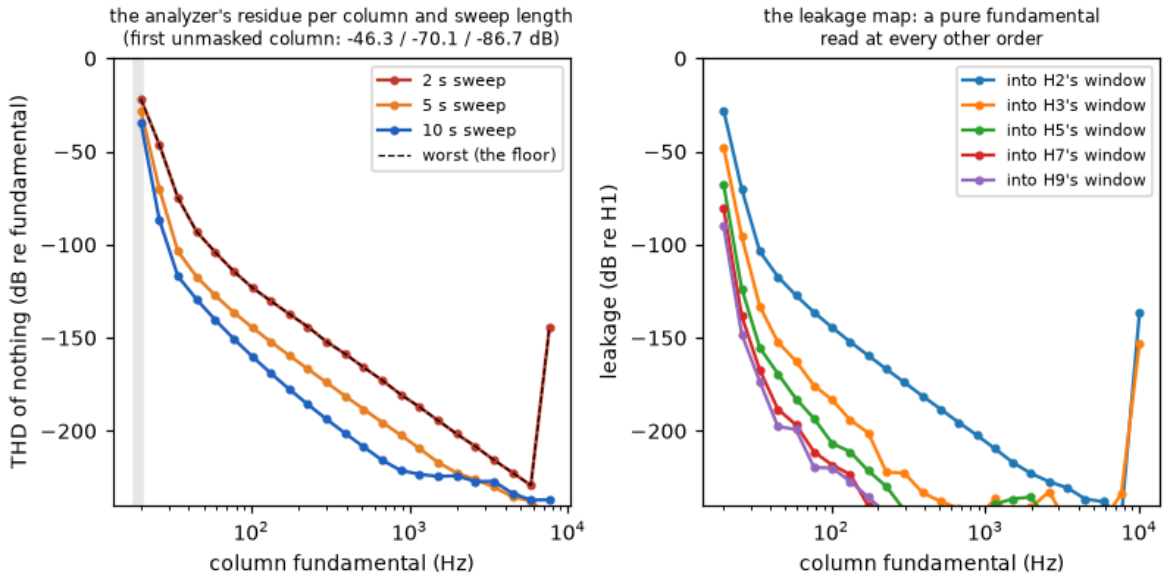



Figure 23: Left: the analysis's own residue at every column of the map for the three sweep lengths — the fade-in leak at the first unmasked column, largest on the shortest sweep — and the worst case the record assumes (dashed). Right: the leakage map used by the check described later: a signal containing only its note, read at every other harmonic.

The composed margin is the square's distance above the *largest* of the three, because the largest fake is the one that binds. A square clears when that distance is at least 6 dB. Figure 24 is the picture of it at one note of the worked example. On the left, against loudness: the square's own THD falling as the drive is turned down, the loop's floor read through the median (with the single-reading version beside it), rising at the quiet rungs by the drive term and flat above the calibration's level, the analysis's residue far below at this note, and, on the right-hand scale, the margin — how far the square stands above whichever floor is higher — against the minimum. On the right, the same at the loudest rung across every note.

**Is the note there?** Every square is a ratio to its own note, and a ratio to a note that is not there means nothing. A few devices — an octave-up effect at full tilt, a rectifier — remove so much of the note that what the analysis reads at the note's own frequency is its own leakage from the louder harmonics. Chapter one measured how far below the loudest thing in a reading the note

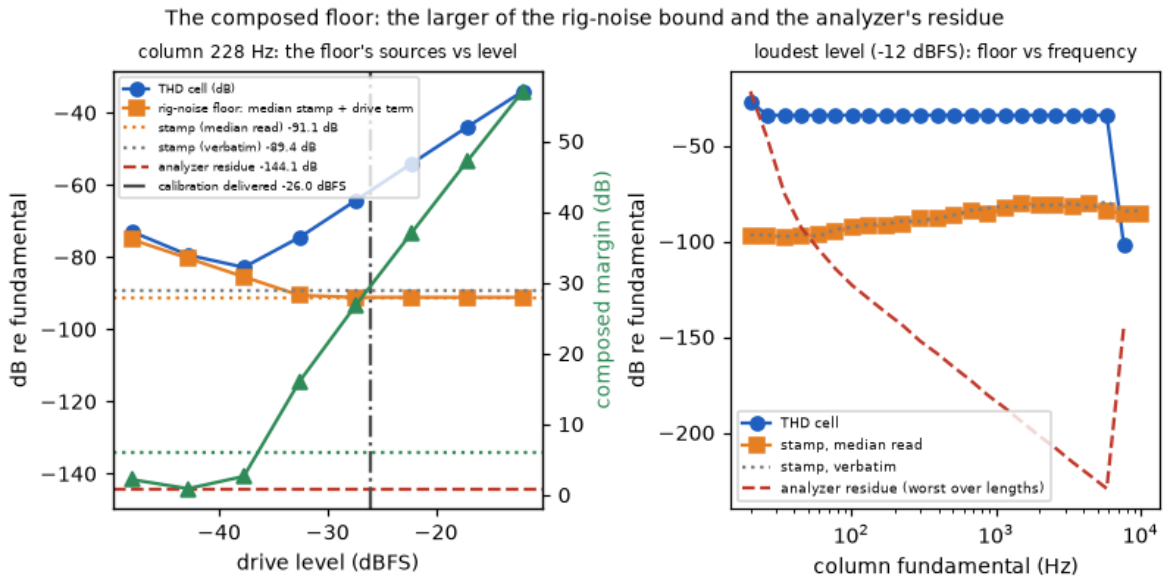



Figure 24: The floor, composed from its sources, at one column of the worked example (left): the square's THD against loudness, the calibration's floor read verbatim and through the median, the drive term rising below the calibration's own level and flat above it, the analysis's residue, and the margin (green, right-hand scale) against its minimum. Right: the same at the loudest rung against note.

can sit before that happens (76 dB), and each square is judged by that rule on its own: present, absent, or undetermined when the note read nothing. The map's verdict is then a majority over every square outside the two fade columns, and it outranks the peak tile: when most squares have no note, no percentage is quoted at all, only how deep the note sits. The verdict is a majority for a reason the check at the end of this chapter shows. On a device that has no note at all, a few squares still read "present" accurately — at the quietest rungs the note's window holds the loop's noise, and beside the low edge it holds the second harmonic's own leakage — and one square's truth must not overrule the map's.

**The peak tile.** The headline number is the loudest square that can be believed, and belief takes two steps. First, the fade columns and the truncated columns are set aside, and so is every square whose note is absent; among what remains, a square is *clear* when its composed margin reaches the minimum. Second — and this is the step that keeps a clean pedal from reading a number — a clear square counts only when the square one column beside it, the same rung one note away, is also clear. That is a corroborated peak. Real distortion spills onto the neighboring notes: a circuit that clips at one drive clips the note next door too, and distortion changes slowly from one column to the next. A noise reading happens to one square alone. The neighbor is asked for along the note axis and never along the loudness axis, because distortion can change a great deal from one rung to the next — a genuine onset at the loudest rung may have no clearing rung beneath it — while two noisy squares in adjacent rungs of one column are as likely as any other pair.

The tile can therefore say one of three things. If any corroborated square exists, the loudest of them is the peak: the most distortion the pedal clearly made. If none does but some unmasked square has a figure, the loudest of those is quoted as a bound, written with $\leq$: the pedal made no more than this, and the rig could not see under it. If the note is absent over the map, no percentage at all. The bound is what a very clean pedal earns, and it is not a small number: a plain wire through a quiet loop reads a bound of 0.133 % on the default sweep, because that is what the loop's noise adds up to at the loudest of 192 squares, and nothing under it can be told

from the rig. Figure 25 shows the rule deciding: on the left, a plain wire through noise, where no square clears and the tile is a bound; on the right, the soft clipper through the same loop, where every clear square has a clear neighbor and the tile is a peak. One case is left open on purpose: a map with no floor information at all — Simulation Mode, or a record made with no calibration — has nothing to judge its squares against, so every square counts as clear and corroborated, and the tile states the loudest one as a peak. That is reported rather than changed, and the last section returns to it.

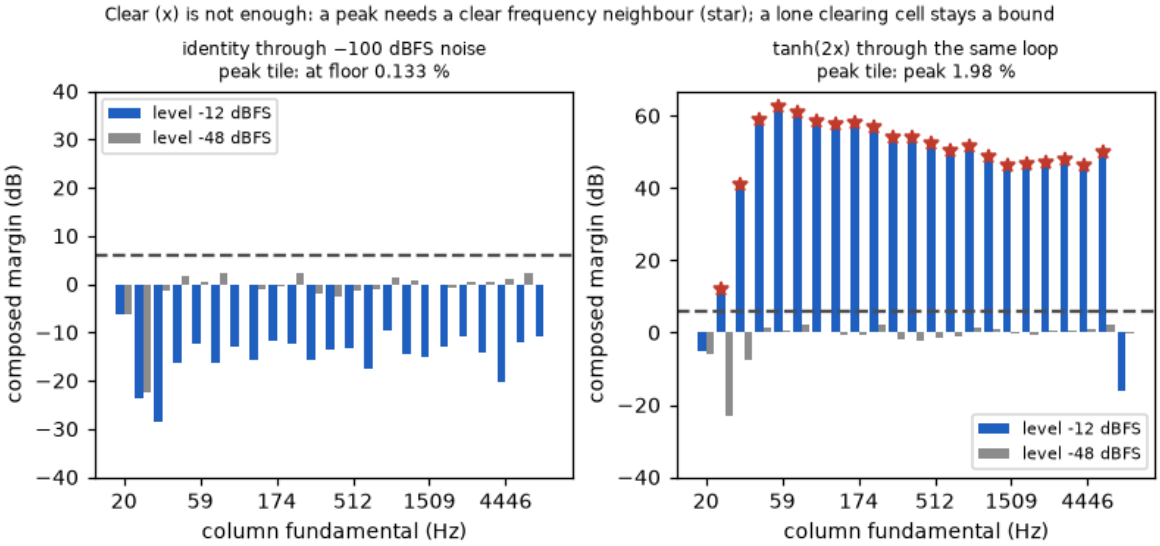



Figure 25: The corroboration rule. Each panel shows the composed margin at every column for the loudest rung (blue) and the quietest (gray) against the minimum (dashed); a cross marks a square that clears on its own, a star one whose neighboring note clears too. Left: a plain wire through a little noise — nothing clears, and the tile is a bound. Right: the soft clipper through the same loop — every clear square has a clear neighbor, and the tile is a peak.

**Does it clean up?** Take one column of the map and read its THD from the quietest rung to the loudest. The cleanup reading says where, going up, the pedal crosses the line of 1 % distortion — a convention for reading the map, drawn where the app's own code draws it; the value is read out of the app rather than typed, by probing the shipped rule until it answers. The reading has three states, and the endpoints are never dressed up as a crossing. If the quietest rung is already over the line, the pedal is *never clean* — within this map's range, which is the only range the map can speak for. If some rung is the first over the line, the pedal *cleans up*, and the level is placed between that rung and the one below in proportion to how far each sits from the line. If no rung reaches the line, the pedal is *always clean* — again, as far as this map goes. The tile beside the map reads the column nearest 220 Hz (A3), the note the Compression measurement probes, so the two records' cleanup figures are about one pitch; every other column carries the same three-state reading too, and the chart's markers draw the genuine crossings. When the tile says never clean, its subtitle says which range it means, and when the pedal's own Compression record — which probes further down — finds a genuine crossing below the map's floor, the subtitle points there instead. Figure 26 shows the three states at the tile's note: the soft clipper crossing the line between two rungs, a plain wire never reaching it, and a clipper whose threshold sits under the ladder's quietest rung.

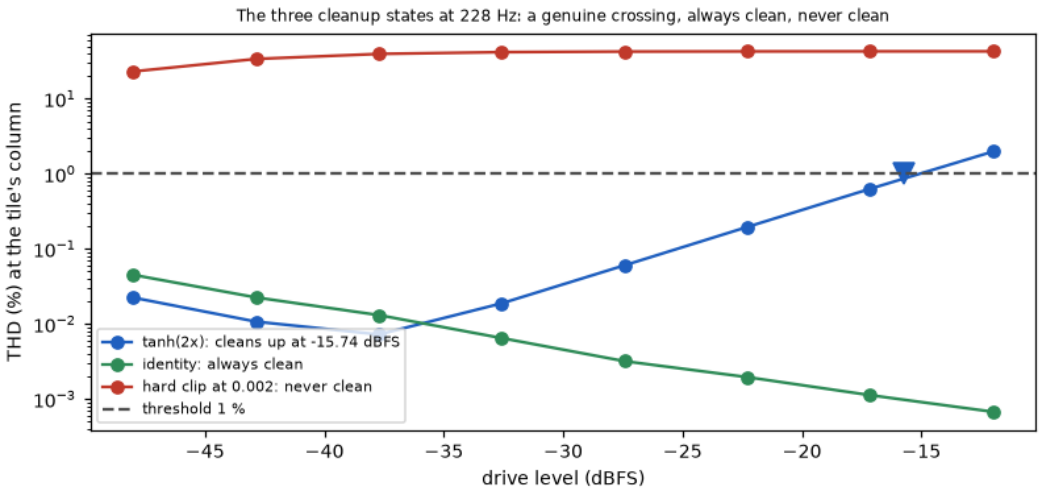



Figure 26: THD against loudness at the tile's column for the three cleanup states: the soft clipper crosses the line between two rungs (the triangle marks the crossing); a plain wire stays under it throughout; a hard clipper is over it at every rung.

## 5.5 The settings

Every setting is in the technical reference's Gain Map chapter, in one table read from the manifest. This chapter repeats the handful a reader acts on. The hardware ladder runs from -48 to -12 dBFS in 8 rungs (5.143 dB apart) and the plugin ladder from -60 to 0 dBFS (8.571 dB apart); the sweep lasts 5 s by default, with 2, 5 or 10 s offered and the choice not stored; the map has 24 note columns; the calibration's floor is read through a median of 7 points; a square clears at a margin of 6 dB; and the cleanup line is 1 %, read at the column nearest 220 Hz.

## 5.6 What the result looks like

The app's own chart, rendered from its chart code on its simulated pedal (never a screenshot), is Figure 27. Across the top run three tiles: the drive range, with the number of rungs measured; the cleanup reading at A3, with a line saying which range the verdict covers; and the peak distortion, with a line saying it is the loudest THD anywhere on the map. The chart itself puts the drive level up the side, labeled so that up is digging in, and the note you play along the bottom, named with its frequency; each square's color is its THD, with the scale under the chart running from the map's own faintest square to its strongest, in percent, and the legend saying that a faded column is a band edge — the outermost columns are drawn faded. This particular map is the app's simulated pedal at a setting that is over the cleanup line at every rung, as its middle tile says. Figure 22 above is the same kind of map as the pipeline draws it, with the margins, the masks and the tiles' marks made visible.

## 5.7 How you can check that the app does what this says

**The second program.** As in the earlier chapters, a second program was written from the description — in Python, independently, not by translating the app's code — and the parity test runs both on the same made-up devices and demands the same answer. It runs 15 cases, and two of the device cases a second time as their phantom capture twins (2 twins). Every case is a known device through a known loop: a delay, a stream of noise added at the interface's input, and a gain applied after them as an input knob would apply it; the calibration is the app's own sweep through the same loop with nothing in it, analyzed by the app's own calibrator, whose measured delay is what cuts every rung — the app's own flow, end to end.

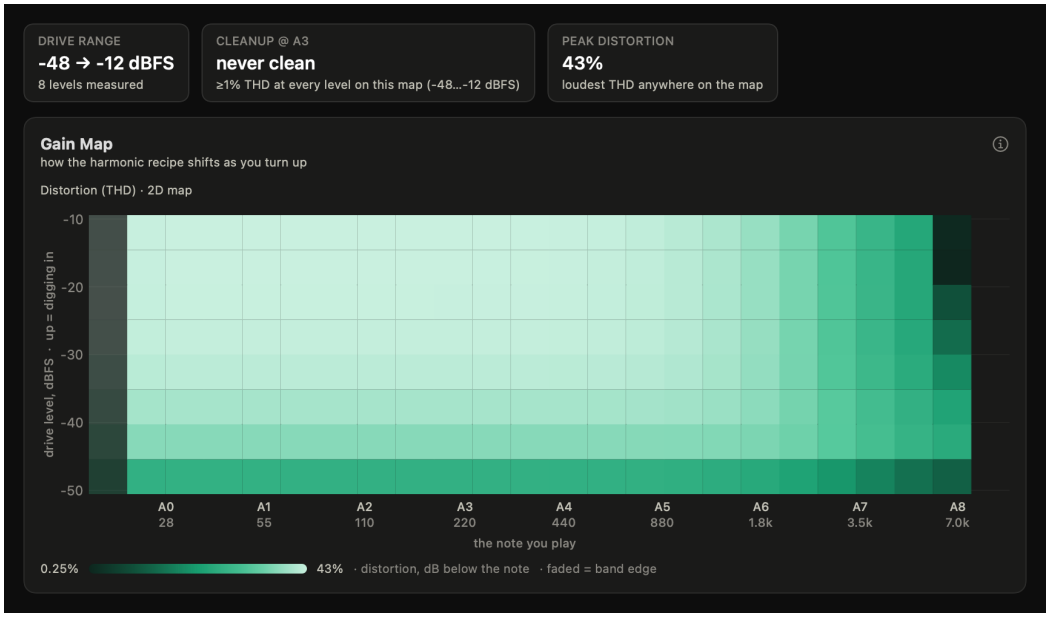



Figure 27: The Gain Map view as the app draws it, for its simulated pedal: the three tiles, the THD map over note and drive level, and the outermost columns drawn faded.

**The known answer.** It is chapter one's. For a device with no memory, the exact size of every harmonic at any loudness can be computed from the device's formula, so the second program prints, beside every square, the exact ladder at that rung's drive and the exact THD it implies; and for the read-time half it applies the app's own rules to those exact squares, so the peak, the cleanup and the verdict have exact answers too.

**The leakage map.** One idea is new here: the leakage map. A signal containing only its note and nothing else — a phantom with one entry — is run through the same analysis, and whatever it reads at every other harmonic is what the analysis itself invents at that note and harmonic. The right panel of Figure 23 is that map. It is measured first and allowed for before any real case is judged: a square whose true content sits close to the leakage landing in its window can read a little off for the analysis's own reasons, that much is subtracted from the disagreement, and what remains is held to the bar. Chapter one's bound on how far one harmonic can leak into another does not carry over, because a shorter sweep leaks worse at the low edge; the leakage map states it here, column by column.

**The cases.** The case families, in a sentence each. The soft clipper and a gentle polynomial curve, each again as its phantom twin. A plain wire through noise at each of the three sweep lengths. A phantom whose note sits far under its own second harmonic, so that the map has no note. A hard clipper over the line at every rung. The soft clipper through the same loop twice, once with the interface's gain turned up. A plugin whose own noise floor stands in for the calibration's. A hard clipper on the plugin ladder, driven to the very top. The worked example recorded late, so that the calibrator must find the delay. And the soft clipper behind a tone filter and the polynomial ahead of one.

**The demands.** The numbers here are written into this document by the pipeline. The two programs must agree, at every square and every harmonic, to within a millionth of a decibel (1e-06 dB) with no filter in the loop — on the run that built this document the worst disagreement was about five hundred-millionths of a decibel (5.1e-08) — and to within a ten-thousandth (0.0001)

through a filter (measured about two millionths, 1.8e-06); every yes-or-no reading — each mask, each square's presence, clear and corroborated, every tile's state and every column's cleanup state — must be identical, the calibration's stamp identical point for point, and the residue identical at every column and length.

**Against the truth.** The app's reading of every square must match the exact ladder, once the leakage is allowed for, within five hundredths of a decibel (0.05) for the smooth devices (measured 0.012), within five hundredths (0.05) on the twins (0.04), and within three tenths (0.3) through a filter (0.15 — the exact answer treats each note as a steady tone through the filter, and a sweep is not steady); the THD squares the same way, within a thousandth (0.001) and two tenths (0.2). Every cleanup crossing must land within three tenths of a decibel (0.3 dB) of the exact curve's crossing (measured 0.197, at the first unmasked column in every case, where the residue's leak sits inside the sum).

**The plain wire's residue.** The plain wire's residue at the first unmasked column must reproduce the three figures the app derives — -46.32, -70.11 and -86.67 dB from the second program against the app's -46.3, -70.1 and -86.7 — within five hundredths of a decibel (0.05), and its tile must be a bound of 0.133 % at the default length with every column always clean.

**Checks on the rig.** Turning the interface's gain up by 12 dB must move the note's level by exactly that much, the calibration's stamped gain by exactly that much (12), and any square's margin by under a ten-billionth of a decibel ($\leq$ 1e-10; the bar is a millionth, 1e-06) — the app's reading of the pedal does not move when the interface is turned up, which is the whole point of the drive term's gain. A recording 100 samples late must be found by the calibrator and cut so that every square moves by $\leq$ 1e-10 dB — nothing the arithmetic can tell from no move at all. Part B's parity is established: on every read-time case — the identity through noise at the three sweep lengths, the device with no fundamental, the never-clean clipper, the loop 12 dB louder, the operative floor on a silent stamp, and the plugin window — the two implementations agree on every margin, every flag, every tile and every verdict, and on the corpus grids the reimplementation reproduces the kernel's own tiles to the digit.

**Two results in words.** Two of the results are worth stating in words. The device with no note — the phantom whose note sits 80.03 dB under its second harmonic — reads absent on about eight tenths of its squares (0.79) and gets a tile with no percentage; the squares that still read present are at the quietest rungs and beside the low edge, exactly where the previous section said the note's window holds noise and leakage, which is why the verdict is a majority. And the worked example itself, the soft clipper on the hardware ladder with no noise: 426 squares compared, the worst raw disagreement with the exact ladder 0.348 dB and 0.012 dB once the leakage is allowed for; its peak tile reads 1.979 % against the exact grid's 1.979 %, and its cleanup tile places the crossing at -15.74 dBFS at the column at 227.6 Hz, against the exact curve's -15.74. The hard clipper on the plugin ladder reads a peak of 41.57 % and cleans up at -25.05 dBFS; the clipper over the line everywhere reads 43.69 % at every rung of its tile column and never clean.

**Stored records.** One last check starts from stored records rather than from a made-up device, and it is the only one that does: the read-time rules — the floor, the margins, the corroboration and the tile — were run by the second program on the stored grids and calibration stamps of 9 real records from the project's own library, and reproduced the app's tiles on every one, 2 peaks within 0.1 dB and 7 bounds to within a billionth of their value (1e-09). A stored record holds results rather than a recording, so for this half of the check the stored grid is the input, and those grids are the same figures the app's own tests pin.

**When it runs.** The test runs automatically on every change to the app, and the tables and figures in both documents are regenerated on every change and compared with the committed copies; the scripts and the instructions for running them are in the repository named on the first page.

### 5.8 Where it stops being a measurement

- **The sweep's length is not on the record, and the floor pays for it.** The residue is taken at the shortest sweep the sheet offers, whatever length was used, so a record made at the default length has its first unmasked column judged 23.8 dB harder than it deserves — the shortest sweep's -46.32 dB against the default's own -70.11 — and a record made at the longest length 40.3 dB harder. That is a bound in the honest direction, never a wrong number, and storing the length would tighten it; that is a change to what a record holds, not to how one is read.

- **The note's level is a level, and your interface is part of the circuit.** Nothing in a square is corrected for the rig, so the note's level in each square is the level the pedal delivered through the interface, its gain and all. On a pedal with a resistor ahead of its clipper that level is also a reading of how hard the interface's input loads it: the same pedal at the same setting into two interfaces with different input impedances is driven differently, and can legitimately make different amounts of distortion. Compare maps taken on different rigs by their shape, never by their absolute figures, and read a difference between two interfaces as a difference between two pedal-and-interface pairs.

- **A shorter sweep leaks more.** The map's sweep and its analysis are each half as fine as chapter one's, so a loud harmonic leaks more into its neighbors' windows at the low edge than chapter one's measured bound allows; the leakage map states the amount per column and harmonic, and a square whose content sits inside it is the analysis's, not the pedal's — which is what the residue and the margins say, and why a device with no note reads "present" beside the low edge.

- **A map with no floor information quotes its loudest square as a peak.** Without a calibration or a plugin's own floor there is nothing to judge the squares against; every one counts as clear and corroborated, and the tile states the raw maximum. Simulation Mode and every record made without a calibration are in that state. Reported, not changed.

- **The models beside the map are for listening.** The record carries one model of the pedal per rung, played by the listening panels; none of the map's numbers come from them. A rendering between two rungs blends two models, and the quieter one clips the signal in a way the pedal does not; that is a property of what the models play, measured in its own chapter, and never of the map.

- **The fade columns and the short column.** The first and last columns are the sweep's ramps; a column near the top may have lost a harmonic above the band. All are drawn faint and kept out of the tiles, and none is corrected.

- **Nothing between rungs.** Each rung is its own sweep, seconds apart from the next; a pedal that drifts between them shows it as row-to-row wobble, which the map draws and does not name.

- **The drive is the signal, never a knob.** The vertical axis is how hard the pedal is driven; the pedal's knobs are wherever you left them, and a knob is mapped one map per position.

# 6 Slots: how a spectrum is read

The next chapter reads a recording by cutting one long, steady stretch out of it and asking how much of each frequency that stretch contains. The answer comes back as a row of *slots*, one per frequency, evenly spaced from zero upward, each holding the strength of the recording at that one frequency. The slots are not infinitely fine: how far apart they sit is set by how long the stretch is, and by nothing else. A stretch one second long gives slots one hertz apart; a stretch twenty times longer gives slots twenty times finer. The stretch is the analysis window, and its length is the price of the resolution.

A tone that completes a whole number of cycles inside the stretch is described by exactly one slot: it lights that slot and no other, and the slots around it hold only whatever noise the recording carried. A tone that does not fit — one that ends a fraction of a cycle short of the stretch's edge — cannot be described by one slot. Its energy spills into every slot: most into the two beside it, less into the next, and a little, never nothing, into every slot further off. That spill is the tone's *skirt*, and its shape is the same whatever the tone. Figure 28 shows both cases, computed through the same code the next chapter uses: two notes that fit the window exactly, over a floor that is flat because nothing is spilling; then the same two notes with the high one moved a quarter of a slot off its center, and the whole floor lifted by its skirt. On the right is the neighborhood of that note, where the spill lands first and hardest — the slots the chapter's coherence check reads.

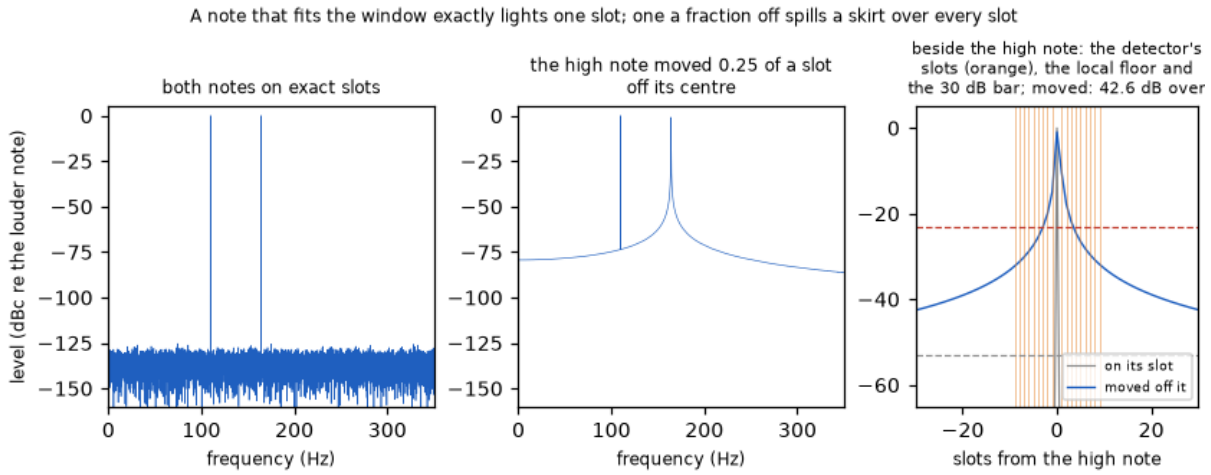



Figure 28: Two notes at the standard drive through a little noise, read through the analysis window the next chapter uses. Left: both notes on exact slots — two lines and a flat floor. Middle: the high note moved a quarter of a slot off its center, as the panel's title states; its skirt covers the whole span. Right: the slots either side of the high note in both cases, the nine the coherence check reads marked, with the local floor beside the note and the bar above it for the moved case.

The stretch is cut with a plain edge — no fade at either end of the cut — which is what makes the exact-fit case perfectly clean and the near-miss case spill. Many analyses soften the edges of their window to blunt that spill, at the cost of blurring every slot into its neighbors. The next chapter does not, because it can arrange for every tone it cares about to fit exactly: it tunes the notes to the window before it plays them. That arrangement is what the word *coherent* means there, and everything the chapter can do follows from it — as does the one thing it must check, which is that the notes really did stay put.

# 7 Chord IMD

## 7.1 What this measurement tells you, and what it cannot

Play two notes at once into a distortion pedal and more comes out than the two notes and their harmonics. The circuit does not treat the notes separately: it multiplies them together, and the output carries tones at the sum of their frequencies, at the difference, and at every combination of multiples of the two — twice the low note less the high, three times the high less twice the low, and so on. None of those tones is in either note alone. This measurement — the two-tone measurement — plays a musical interval, a power chord's fifth by default, and lists every tone the pedal invents from the pair.

Each invented tone is an intermodulation product, named by the recipe that makes it: so many of the low note plus or minus so many of the high. Its product order is how many multiples went into it — the sum-and-difference tones are order two, "twice the low less the high" is order three — and the app lists every recipe up to order 7, 54 of them on the standard probe. Each is read at one slot of one long spectrum and quoted in dBc: decibels below the louder of the two notes you played, so "−40 dBc" means a hundredth of the note's size, whatever the volume was.

The record holds the level of every one of those recipes; the two notes as the pedal delivered them; three percentage figures that earlier versions of the app showed as tiles, kept for the record as those tiles' frozen definitions; and the raw spectrum in the slots beside each played note, kept for the check described later. Everything else — what each tone's reading is, how deep the recording could see, which musical job each tone does, and how hard the pedal was driven — is worked out afresh from those stored facts every time the record is opened, so a corrected rule corrects every record ever made.

**What it cannot tell you.** Stated once.

1. **Only one loudness.** One capture is one drive level, and a clipper's invented tones grow with drive faster than anything else about it. Two records are comparable only at the same drive, which is why the record states its drive on the page.

2. **Whether a tone is the pedal's or the interface's.** An interface's converters invent tones of their own. A tone that stands above this recording's noise is real; the measurement does not say what made it. Only a capture with nothing in the loop — a null run — separates the two, and the floors described below neither do that job nor claim to.

3. **Which stage moved the notes.** When the coherence check fails, the record says the notes did not stay still and stops there. A pedal with a dying battery moves a note as surely as an interface's clock does; the verdict has no culprit, by design.

4. **Nothing below the recording's own floor** — and, for the three old percentage figures, nothing about a floor at all: they are sums made at capture time over the tones that passed the presence test, and they carry no bound.

5. **Nothing above order 7.** The list of recipes is a small-signal census. A hard clipper driven into its bends has real energy at orders the list does not reach, so an empty row high in the list means measured and nothing found at this drive, never that the pedal makes nothing there.

6. **Nothing about listening.** The chart says how much of each invented tone there is. It says nothing about what the distortion sounds like.

## 7.2 The test signal, and the lattice

The signal is two sines of equal size played together, faded in over 50 ms and out again at the end so that nothing bangs. The standard loudness is -26 dBFS for the pair combined — each note -32 dBFS on its own — the same drive the other measurements use, so that they all describe the pedal at one operating point; the worked example later in the chapter uses the loudest drive the hardware sheet offers, -6 dBFS, where the reference clippers actually clip. The default pair is A2 and E3, a fifth on the low strings, named in ordinary equal temperament (110 Hz and 164.8 Hz). The measure sheet offers one kind of probe, the interval: you choose the two notes.

**The premise, stated once.** Both notes are tuned, before anything plays, so that each completes a whole number of cycles inside the analysis window (Section 6). That is coherent sampling, and it has a consequence that makes the whole measurement possible: slots add the way frequencies do. If the low note sits on slot number $k_1$ and the high note on slot $k_2$, then "twice the low less the high" sits on slot $2k_1 - k_2$ — exactly, not approximately — and so does every other recipe, on its own slot, leaking into no neighbor. The set of slots the two notes and all their invented tones occupy is the analysis lattice, and the analyzer reads each recipe at its one slot and nowhere else.

**Two numbers describe the lattice.** Write the two slot numbers as a common factor times a simplest fraction — the low note's slot is $g$ times $a$, the high note's $g$ times $b$, with $a$ and $b$ sharing no factor. The two parts answer two different questions.

The fraction $a : b$ is the reduced tone ratio, and it decides whether two different recipes can ever land on the *same* slot. They can only when the two numbers are small: for recipes up to order 7, a fraction whose two numbers add to 15 or more keeps every recipe on its own slot, and the bound is exact — the fraction 1 : 13, whose numbers add to 14, still has a collision, and every fraction one higher is clean. The common factor $g$ is the lattice spacing, and it decides how *close* two invented tones can get: every recipe's slot is a whole multiple of $g$, so no invented tone can sit nearer than $g$ slots to another, or to either played note. A lattice can pass the first test and fail the second. The probe the app used until its window was lengthened shared no slots and still put four recipes two slots from the played notes, where a note's own skirt lives; the records made on it carry that mark, as a later section explains.

That is also why a consonant interval is an awkward one to measure. A fifth is, in its simplest form, a ratio of three to two, and three-to-two is a fraction with tiny numbers — the most crowded lattice there is, with recipes landing on top of each other and on the notes themselves. The measurement has to move the notes off that exact ratio, and the question is how far, and in which direction.

**The snap.** The lattice snap is done in two steps, so that the interval you ask for is one interval at every sample rate. First, the fraction is chosen *once*, against every sample rate the app offers at the same time: no two recipes may share a slot; every recipe must sit at least 13 slots from the played notes at every rate; both notes must stay within 15 cents of the pitch you asked for; and among the fractions that manage all three (26 do, for the fifth), the one whose interval is nearest the one you asked for wins. The default fifth lands on 109 : 163 — an interval of 696.7 cents against the 700 of equal temperament, 3.3 cents flat, and the same at every rate. Second, on the rate at hand, that fraction is scaled up by whatever common factor fits the slot grid: at 96 and 48 kHz the notes land on slots 2 398 and 3 586, spaced 22 apart, with the low note -3.6 cents and the high -6.9 cents from equal temperament; at 88.2 and 44.1 kHz on slots 2 616 and 3 912, spaced 24, at 0.33 and -3 cents. The interval tile on every record prints the offsets that capture was played at. Keeping the interval outranks everything else: a window too short to place the

fraction at a good spacing — Simulation Mode's quick preview — still asks the same musical question, and the reading rules carry the crowding accurately.

Some pairs the app refuses. If a requested interval cannot escape an exact small fraction within the cents budget at all, the plan stops before anything plays and says which lattice it found and why it fails. It never falls back to the naive tuning, because on an exactly degenerate lattice several recipes share each slot, the enumeration keeps one label per slot by a tie-break, and the record would come out looking cleaner than the pedal is — a table whose labels are not facts.

**The window is a duration.** The analysis window is 2 097 152 samples at 96 kHz — two to the power 21, about 21.8 s of steady tone, with slots 0.046 Hz apart. What that fixes is the *duration*, not the sample count: at each of the other rates the app takes the power of two nearest the same length of time, so the 44.1, 48, 88.2 and 96 kHz rates form two pairs that share a slot grid. The two-tone signal is then made long enough to hold the window, plus the fades and a short settle at each end — 22.3 s at 96 and 48 kHz, 24.3 s at 88.2 and 44.1 kHz — never set by hand. That rule exists because of a mistake it corrected. An earlier version fixed the window as a sample count and set the signal's length by hand, and at two of the four rates the window ran into the fade-out: the notes no longer fitted the window, their skirts flooded the low slots, and the deepest products could not be told from the window's own leakage. Figure 30 shows the window inside the signal as the app now sizes it, and the old case beneath it. Figure 29 shows the lattice itself: every recipe as a stem at its slot and order on the shipped fifth, the same on the old crowded probe with its four recipes beside the notes in red, and a close look at the slots beside the low note.

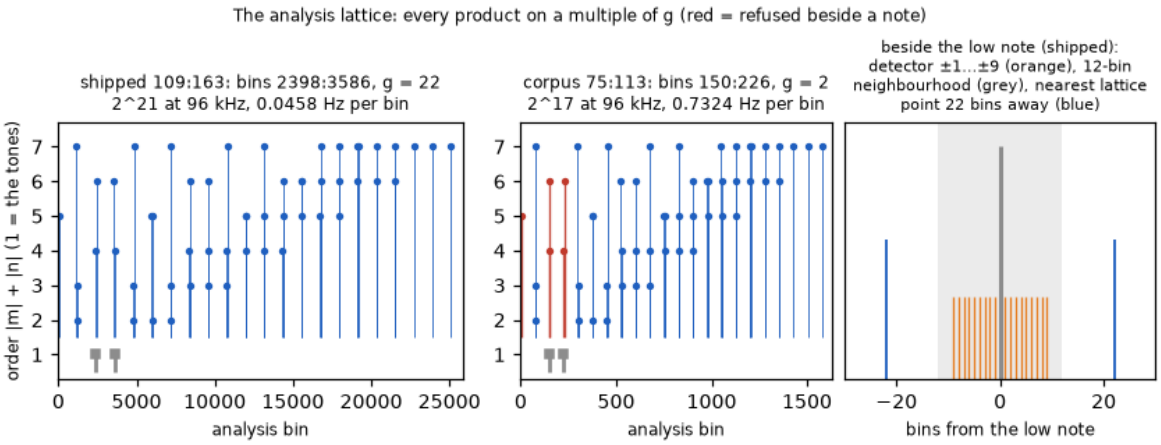



Figure 29: The analysis lattice. Left: every recipe as a stem at its slot and order on the shipped fifth, the two played notes as squares at order one; every stem sits on a multiple of the spacing. Middle: the same on the crowded probe every earlier record was made on — the four recipes two slots from the notes and the one two slots from zero are red, refused on position. Right: the slots beside the low note on the shipped lattice — the nine the coherence check reads, the neighborhood the reading rule guards, and the nearest recipe a spacing away.

## 7.3 How the answer is read

Eight steps, each in words; the technical reference gives the exact rule for each.

**The window.** The app measures the delay through the rig on every capture and lines the recording up with the signal. The window then starts after the fade-in and the settle and runs for exactly its length; the moment the fade-out begins is the end of the steady state, and the signal

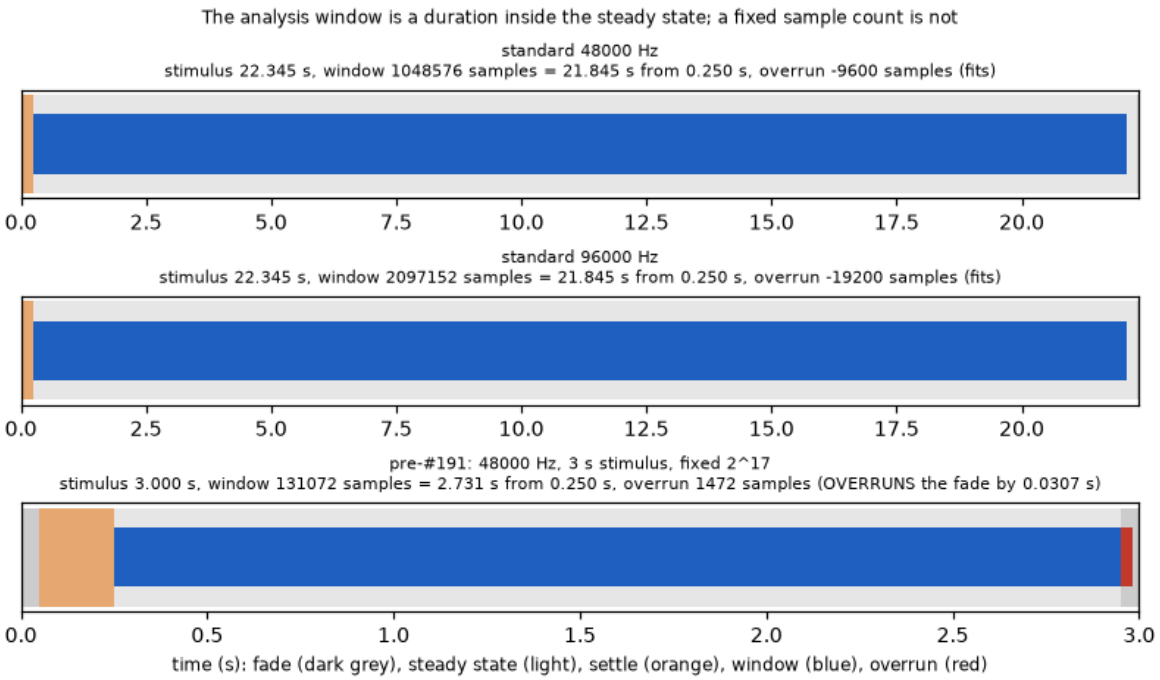



Figure 30: The analysis window inside the two-tone signal at two rates as the app now sizes it (top two), and the earlier fixed-length window on a hand-set signal, which ran into the fade-out (bottom, the overrun in red).

is sized so that the window ends before it. The analyzer never shortens a window it is handed: the notes were tuned to fit that length, and a shorter cut would spill far worse than the fade it avoided.

**The spectrum.** One long transform over the window, cut with plain edges and no taper (Section 6), scaled so that a note which fits exactly reads as its own size.

**The walk.** Starting with the two notes' slots and slot zero already claimed, the analyzer walks every recipe from order two up to order 7, in order, and admits each one whose frequency lies between one slot's width and just under half the sample rate — unless its slot is already taken, in which case the recipe is dropped. So where a crowded lattice puts two recipes on one slot, the lower order keeps it, and on the shipped fifth, where nothing collides, exactly 54 recipes are admitted. The whole lattice is collected before any noise is measured, so that the noise can be measured away from it.

**The reads.** Each recipe's size is the value of its one slot, exactly; so are the two notes'. An earlier version took the largest of the slot and its two neighbors, "to tolerate drift" there was none of. That left a strong tone untouched and lifted every reading at the noise by about 4.2 dB on average; records made that way still carry those reads, and their floor was measured the same way, so their margins stand.

**The local floor, and the gate.** Each invented tone is judged against the noise measured right beside it: the local floor is the middle value of the slots from 3 out to about 8.8 Hz out on either side (192 slots at the standard rate, and never fewer than 12), with every slot the lattice claims cut out by name — every other recipe, both notes, slot zero, and the detector's slots beside each note described below. The three nearest slots are left out because they are the tone's own: if

coherence is ever lost, a skirt lands there first, and a floor must never be read from the skirt of the thing it judges. The span is held in hertz rather than in slots because noise is local in frequency, and a fixed slot count would have shrunk sixteenfold when the window grew. The presence gate is then a fixed bar: a tone counts as *present* when it stands more than 4 times (12 dB) above its local floor. That verdict is decided once, at capture, and written into the record; nothing decided later can change it. The right-hand panel of Figure 33 is the picture: the gray span either side of one recipe, the lattice slots cut out of it in red, the floor through the middle and the gate four times above it.

**The three percentages.** Over the invented tones inside the audible band (20 Hz to 20 000 Hz) the analyzer adds up three sums as percentages of the notes: two borrowed from laboratory two-tone standards, and one over every tone that passed the gate. They were the tiles of earlier builds and survive as those tiles' frozen definitions; the roles described later replaced them.

**The detector.** For each played note the analyzer keeps the raw spectrum at the 9 slots either side of it, and the local floor beside that note, in the record: the coherence detector. Nothing else keeps those slots, and they are what the coherence check reads.

**What is stored.** The invented tones, loudest first, each with its recipe, frequency, size, order and gate verdict; the signal as delivered; the two notes' sizes; the three sums; the band; and the detector. Nothing worked out at reading time is stored.

### 7.4 What the app decides when it reads a record back

Everything in this section is computed from the stored record every time it is opened. A reader that knows the window's length says so; one that does not places nothing on the lattice and makes the smaller claim.

**The capture floor.** The record states how deep this particular recording could see: the capture floor. It is the middle value of the levels of every recipe the gate did *not* pass — leaving out any the rules below refuse for a reason of their own: a recipe inside a note's skirt, one under the band's low edge, one beside a louder tone — with at least 3 such empty slots needed, and fewer than that honestly reported as no floor. Slots that lie under the loudest tone's own skirt deliberately stay in, because on a record like a rectifier's every empty slot is under it, and a vanished floor would read as "every tone resolved". The two floors run one way: the gate is decided at capture against the raw spectrum, the capture floor is a summary made afterwards over slots already judged, so leaving a refused slot out of the floor changes which judged slots are averaged and never which slots were judged. This floor is gathered with the other measurements' floors in Section 9.

**One reading per tone.** Every invented tone gets one product reading, and the app prints four kinds of answer: measured; nothing found down to a stated level; could not be read, and why; and out of band. The rules are applied in a fixed order, stopping at the first that fits, and position comes before level throughout.

- **Beside a played note, or beside zero** — within 12 slots of either note's slot or of slot zero: could not be read. A slot that close holds the note's own skirt, and a slot beside zero holds the recording's own offset and slow drift. This is decided on position alone, before any level is looked at, because the gate samples its floor from slots that lie mostly outside the skirt the slot is sitting in and would pass such a slot with a confident margin.

- **Under the band's low edge** — below 20 Hz: out of band. A fact about frequency, not a limit of the instrument, so its sentence never merges with a lattice's or the analyzer's. On the shipped fifth one recipe lands there: $3f1-2f2$, about 1 Hz up, where nothing a pedal does can be told from the recording's own drift.
- **Beside a louder invented tone** — within the same 12 slots of a louder tone that itself passed the gate: could not be read. At that separation the quieter slot holds the louder tone's skirt. The louder of the pair is read; the quieter is not. A neighbor that is only noise casts no skirt and refuses nothing. On the shipped fifth nothing is ever closer than 22 slots, so this rule marks nothing; on the old crowded probe it marks many.
- **Inside the analyzer's own leakage** — on an old record whose window ran into the fade-out, a tone too deep below the loudest component to be separated from the window's skirt: could not be read. No record the app makes now gets this one.
- **Under the loudest tone's skirt** — on a record whose loudest invented tone stands above the played notes and whose coherence check could not be read, a tone deeper below that loudest one than a skirt reaches at the lattice spacing (36.8 dB at 22 slots): unread, never present and never absent.
- **The gate passed** — measured, with its margin over the capture floor when there is one.
- **No floor to read against** — could not be read, carrying the count of empty slots it had.
- **Otherwise** — nothing found down to the slot's own level: a bound, never a value, and never a rank.

A refusal is never quietly filed as "absent", and "absent" is never a number. The record counts the four kinds and states its floor by its reason: a floor, or "every tone resolved" only when nothing was refused, or "too few empty slots" with the counts.

**Shared slots, and the second-order cross-check.** On a record made on an exact small fraction — the original fifth, an exact three to two — several recipes share a slot, and the label the record shows is the walk's tie-break. Such a record marks those rows as shared, from the stored notes alone, and the cross-check leaves them out. The cross-check itself is a fact about second-order tones: a single squared term makes the sum and the difference of the notes at equal size, and twice each note at sizes that differ by twice the notes' own gap in the output, so a tone standing more than 6 dB above what its partner licenses is not a clean second-order reading — the slot holds something else too, and its label and rank are not to be trusted. Only excess flags; the quieter partner stays trustworthy.

**The coherence check.** This is the section the reader most needs, because everything above rests on one premise, and the premise has no margin.

*What "coherent" means here.* The window is cut with plain edges, so a note that fits it exactly is perfectly clean and a note that does not spills a skirt over every slot (Section 6). The notes were tuned to fit. But the tuning is a fact about the signal the app sent; whether the notes *stayed* on their slots for the whole window is a fact about the chain, and anything in it that drifts during the capture — an interface clock, a sagging supply, a circuit warming up — breaks the fit. When it breaks it does not degrade gently: the note's skirt lands across the whole lattice and buries the invented tones, and every level on the record, floor included, is a reading of the skirt.

*Where to look.* The spill lands first and hardest in the slots immediately beside each note, so those are the slots the coherence check reads: the 9 either side of each note, kept in the record by the detector. Every one of the nine is read, not only the odd ones, and that is a measured

fact rather than a choice: a note that wobbles an even number of times across the window puts its spill in the even slots only, and a detector that read the odd ones alone saw a clean floor on the very capture whose stored recipes two slots from the notes read far above it.

*The floor and the bar.* The worst of those slots is compared with the local floor measured beside that same note, and the excess is read against a bar of 30 dB over it. The bar is relative to the floor, not an absolute level, because a spilled note's energy does not shrink when the slots get finer while noise does; an absolute bar was right at one window length only. Beside the excess the record prints what noise alone would put the worst of that many slots over the floor — about 7.8 dB for the 36 slots the detector reads, about 4.8 dB for the 4 that older records offer — a reference so that a reading over four slots and one over thirty-six can be laid side by side, never a bar.

*Distance is only a stand-in for level.* Being far from every invented tone is necessary and not enough. Under lost coherence a tone spills a skirt too, and a slot beside a note reads the note's skirt only while the loudest invented tone, sitting a spacing away on the other side, is quiet enough for its skirt to sit under the note's there by a stated margin (10 dB). That gives each slot a ceiling on how loud the loudest invented tone may be: the nearest slot tolerates one up to 16.4 dBc, louder than the notes themselves, and the ninth only one below -6.81 dBc, so the nearest slots are the last to survive. The check reads only the slots whose ceiling the record's loudest tone clears. A device whose invented tones stand above its own notes — a full-wave rectifier does this — clears no ceiling, and then the check reads nothing and the line says why: the slots beside the notes would be reading the products' coherence, not the notes'.

*What the verdict is, and is not.* When the check reads and the excess is over the bar, the record says the played notes did not stay stationary across the window, from some source in the chain that the reading cannot name. Nothing is refused, discarded or re-run: the capture is saved as measured with that sentence on it, and the measure sheet announces it the moment the number exists. The sentence never names a stage — a pedal with a dying battery moves a note over twenty seconds as surely as a rig — and it is a statement about this capture, not about the pedal. Under the bar it says the premise held.

*What it means for each invented tone.* A note that moved spills a skirt over the slots, and the skirt thins with distance in a known way: what reaches a slot twenty-two steps from a note is a small fraction of what the nine slots beside it read. So the record does not write off every invented tone at once. It works out, for each one, how much of the notes' spill — and of the spill of any loud invented tone, which carries the same wobble — lands on its slot, and reads the tone if it stands at least a stated margin above that. The tones that do not are marked unread: not present, not absent, their tile a lower bound, and a quiet slot under the spill is not counted toward the floor either. The margin was set by running the check's own wander fixture through several pedals and measuring how far a read tone's level could be off; at the margin it is under two decibels. The sentence then says how many tones were read clear of the spill and how many were not. On records made before the detector existed, and on the short-window lattice, the record keeps the older sentence — that the whole spectrum is smeared — since those readings interpret only a loop with nothing in it.

*A shape, never a verdict.* The record also quotes the sideband pair statistic: the average of the two innermost slots beside each note, and the difference between the high note's pair and the low note's. A shared wander in frequency puts the high note's pair above the low note's by a fixed amount the record prints beside it; a wobble in level, or a wobble in timing that treats both notes alike, leaves the two pairs equal. The line states the numbers and draws nothing from them, because equal pairs are consistent with two different causes and the shape has not earned a verdict. Figure 31 shows the detector's slots on a clean loop and on the pipeline's wander fixture, and the ceilings.

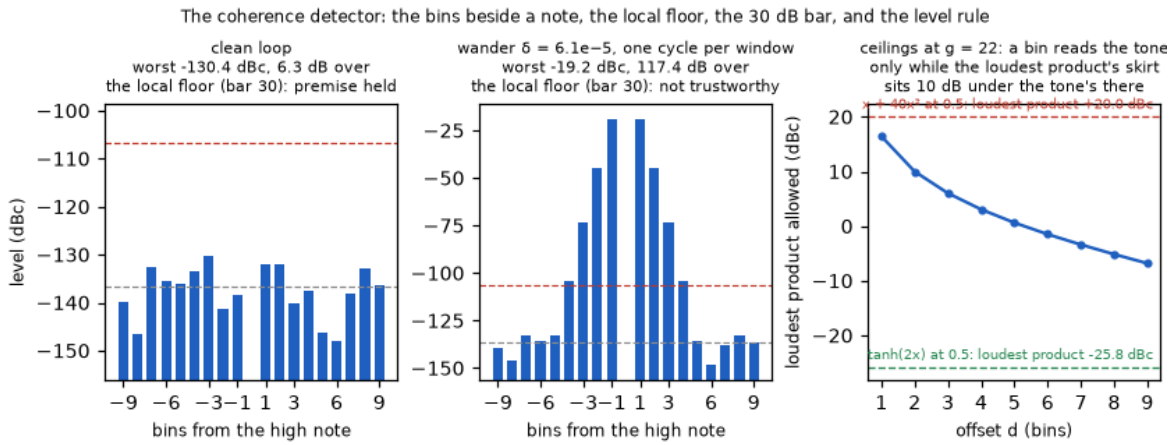



Figure 31: The coherence check. Left: the nine slots either side of the high note on a clean loop through a little noise, with the local floor beside the note (gray) and the bar above it (red) — within a few decibels of the floor, premise held. Middle: the same slots on the pipeline's wander fixture, a shared drift of both notes at one cycle per window — far over the bar: the notes did not stay stationary. Right: the ceiling each slot sets on the loudest invented tone at the shipped spacing, with the worked example's loudest tone far below every ceiling and a squarer driven above its own notes clearing none.

**The harmonic roles.** The tiles at the top of the record sort the invented tones by what each does to the chord you *asked for*. That last phrase is the point. To keep the tones separable the app moved the notes a few cents onto the lattice, so a sorter that read the lattice's own frequencies would call the octave below the root "twenty cents flat" when it is exactly the chord tone the lattice displaced. So the roles are classified at the interval you requested and measured at the lattice: the harmonic role is decided from the recipe alone, and the level is read from the slot the tone actually landed on.

The request is not written on the record, so the app recovers the two notes as the nearest equal-tempered pitches to the delivered tones, which is unambiguous inside the cents budget — the one place the delivered frequencies are consulted, and only to name the notes. It then takes the interval's just ratio: the simplest whole-number fraction, using only the factors two, three and five, within 20 cents of the equal-tempered interval — 3 : 2 for the fifth, so the grid is built on the note a fifth's ratio implies, a fundamental of 55 Hz for A2 and E3 — and it lays a grid of multiples of that fundamental. Every recipe lands on that grid at a whole-number step, exactly, and its pitch class is read off by octave reduction: halve or double the step until it sits inside one octave above the root, then ask which note that is and how far from it. Figure 32 shows where each of the 54 recipes lands on the fifth's grid, colored by the role assigned to the step; the role table in the technical reference lists each step's pitch and role.

The five roles, named as the app names them. **Thickening**: a tone on a played note's pitch class, at or above the low note — it reinforces the chord. **Growl (sub-octave)**: the same pitch classes, below the low note — weight under the chord. **Added colour**: a tone on a pitch that is not a chord tone but is in tune with the chord (a second (or a ninth), a minor third, a major third, a fourth, a fifth, a minor sixth and a major sixth above the root). **Sourness (clash)**: everything else — any other pitch class, and any step that lands more than 20 cents from every fret. **All products**: the total, how much rather than what kind. The tolerance is not chosen: the widest gap between a just interval built from those three factors and its equal-tempered fret is under it, and the nearest point a two-tone grid can produce from a larger factor is beyond it, so the bar sits in an empty band. A role's figure is the power sum, in dBc, of the measured tones in it, counting each slot once; a tone the app found nothing for adds nothing and is bounded by the record's floor; a tone the app could not read was never looked at, so a role that holds one is

marked as a lower bound, and the line under the tiles names the recipes left out. The record also states what it enumerated — the recipe count and the orders — so a tone missing from the list can be told from one that was measured and silent.

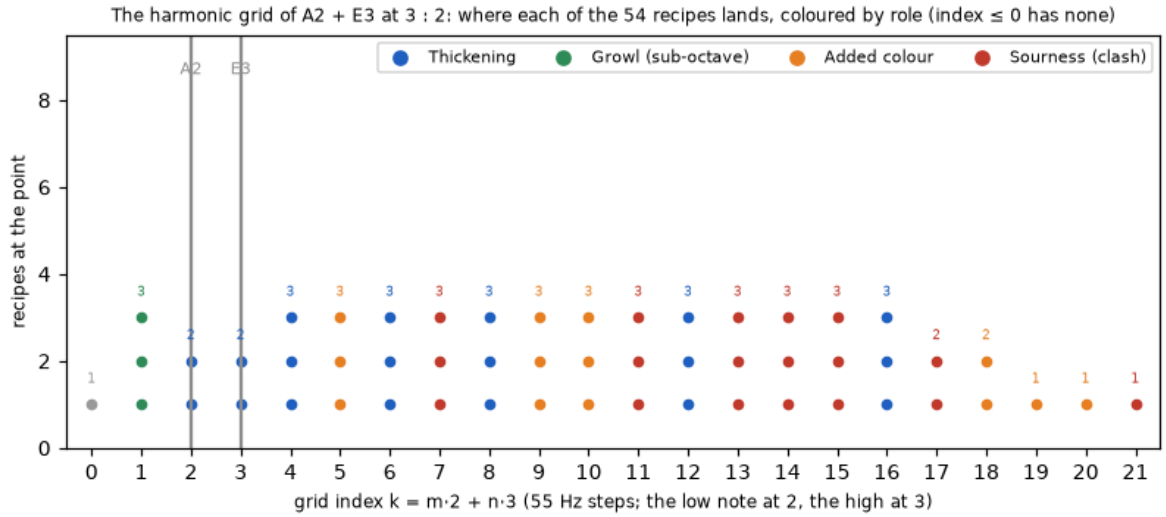



Figure 32: The harmonic grid of the requested fifth at its just ratio: every recipe at its grid step, stacked where several share a step and colored by the role assigned to that step; the two played notes marked. Step zero and below has no role.

**The drive line.** The record states how hard the pedal was driven, read from the stored signal: the two notes' combined peak and each note's own level in dBFS, in volts at the pedal input as well when the record's own calibration measured the interface, and a note that the two tones are equal — every order relation above assumes it.

## 7.5 The settings

Every setting is in the technical reference's Chord IMD chapter, in three tables read from the manifest: the parameters, the lattice the app chooses on every sample rate, and the role grid of the default fifth. This chapter repeats only the handful a reader acts on. The default pair is A2 and E3, delivered -3.6 and -6.9 cents from equal temperament at 96 and 48 kHz (0.33 and -3 at 88.2 and 44.1 kHz), within a budget of 15 cents; the window is about 21.8 s and the signal 22.3 s or 24.3 s long; the walk covers 54 recipes up to order 7; a tone is present at 4 times its local floor; the coherence bar is 30 dB over the floor; and a grid step counts as on a fret within 20 cents.

## 7.6 What the result looks like

Figure 33 is the worked example — the soft clipper of the earlier chapters, at the loudest hardware drive, on the shipped fifth through a little noise — as the pipeline reads it. On the left is the whole spectrum below the notes' first few multiples: every recipe marked measured (blue), nothing found (gray) or refused (red), the two notes as squares at zero, and the capture floor as the dashed rule with its slot count. The clipper is symmetric, so every even-order recipe reads at the floor and every odd-order one stands well above it, in a ladder that falls away with order. On the right is the local-floor neighborhood of one recipe: the gray span either side of it, every lattice slot cut out of it in red, the floor as the middle value of what remains, and the gate four times above it.

The app's own chart, rendered from its chart code on its simulated pedal (never a screenshot), is Figure 34. Across the top run the tiles: the interval played, with the cents each note was moved

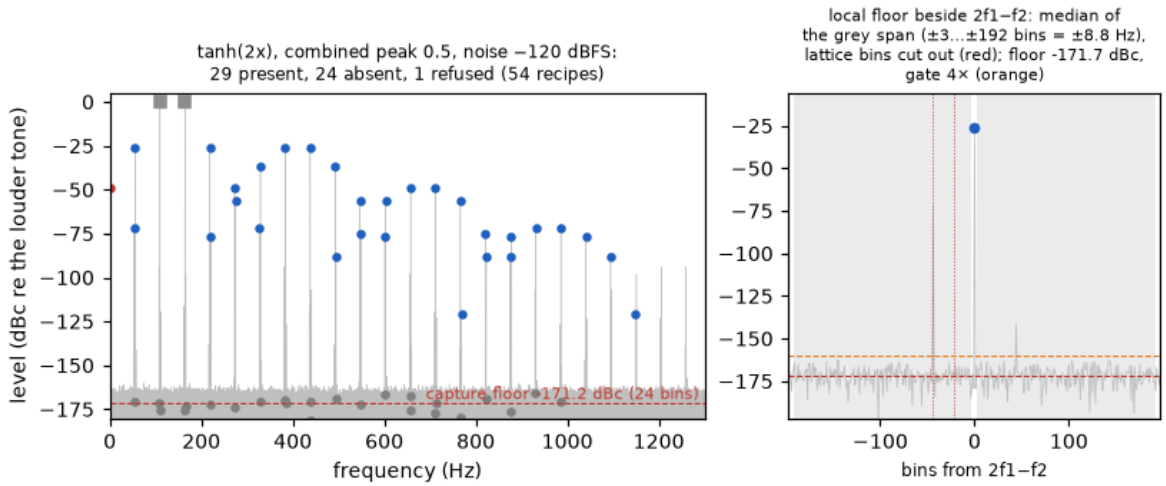



Figure 33: The worked example, a soft clipper at the loudest hardware drive on the shipped fifth through seeded noise. Left: the spectrum with every recipe marked measured (blue), nothing found (gray) or refused (red), the two notes as squares, and the capture floor as the dashed rule. Right: the local-floor neighborhood of one recipe, with the lattice slots cut out (red), the floor (red, dashed) and the gate (orange, dashed).

and the frequencies delivered; then the five roles, each with its figure in dBc and a one-line gloss. Under them, lines state the recipe count and orders enumerated, that the roles are assigned at the requested interval and the levels read at the lattice, and the drive level. The chart itself sits on a musical pitch axis, labeled with note names and their frequencies: the two played notes in one color and every invented tone as a stem in one of two others, one for the even orders and one for the odd, with the legend saying that a dotted stem is not a reading and a faint one was read against the floor and not ranked. Below the chart the invented tones are ranked loudest first, each with its recipe, the pitch it lands on and its deviation in cents, its frequency, its level in dBc, its margin over the floor, and its order; under the table, one line names the recipe under the band, one gives the floor — or says why there is none — and one is the coherence check, quoting the worst slot, its excess over the floor beside the note, what noise alone would put there, its distance from the bar, and the pair statistic.

## 7.7 How you can check that the app does what this says

**The second program.** As in the earlier chapters, a second program was written from the description — in Python, independently, not by translating the app's code — and the parity test runs both on the same made-up devices and demands the same answer. It runs 28 cases, and each device case a second time as its phantom capture twin (9 twins), on the signal the app's own tuning produces, so the choice of lattice is exercised rather than assumed.

**The known answer.** Here it is what makes the check strong. For a device with no memory — one whose output at each instant depends only on its input at that instant — two notes on exact slots produce an output that repeats along two directions at once: advance the low note by one cycle and the output repeats, advance the high note by one cycle and it repeats again. A signal that repeats on a grid like that is made of exactly the recipes the chapter lists, and each recipe's size is a definite number that can be computed by integrating the device's formula over one cycle of each note. The second program does that integration for every case and prints the exact size beside the app's reading of it; for the formula devices it also works the size out a second way, by expanding the formula, and holds the two methods to each other.

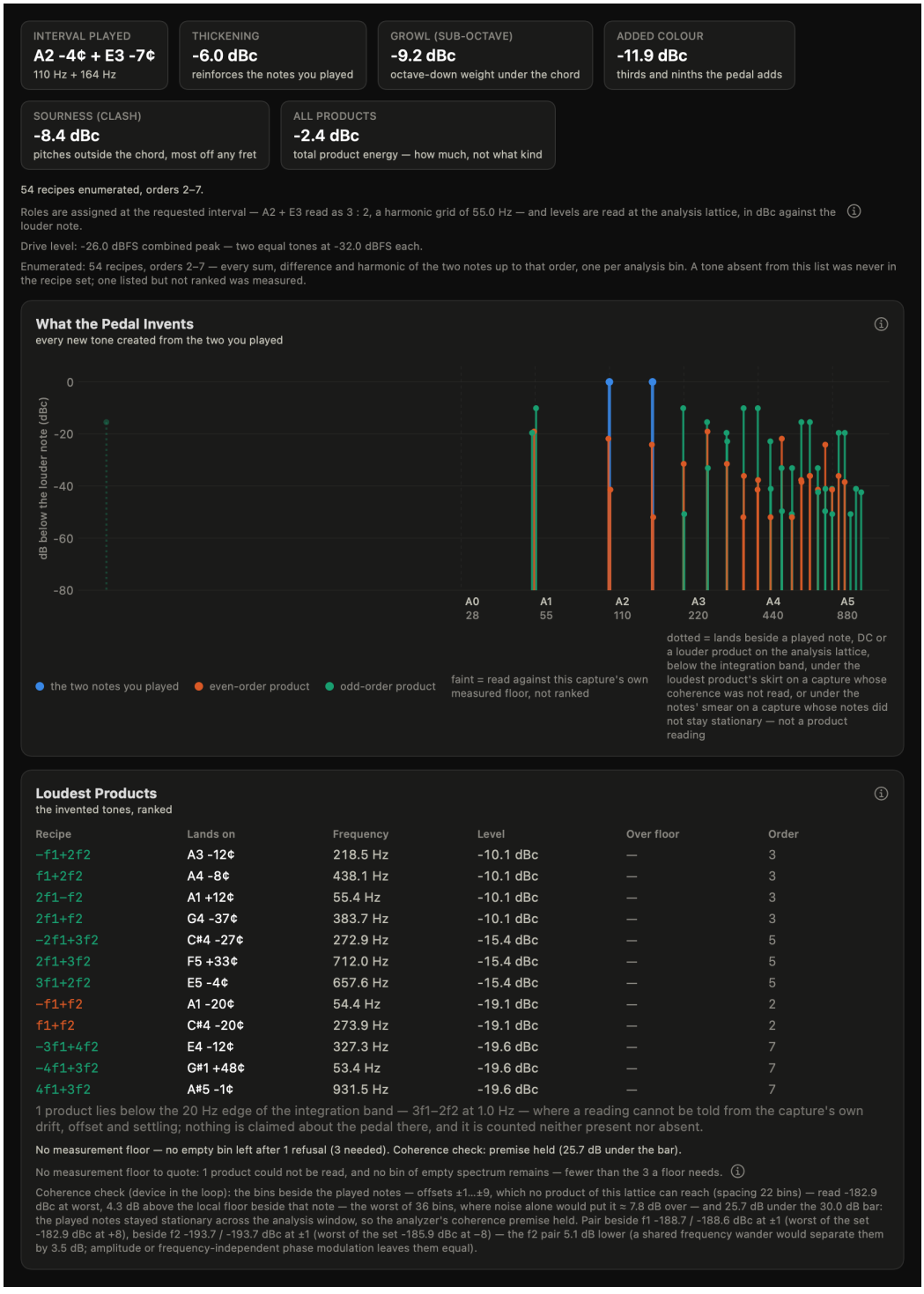



Figure 34: The Chord Roughness view as the app draws it, for its simulated pedal fed the default fifth at the standard drive.

**The cases.** The case families, in a sentence each. The lattice alone, at every sample rate and both short windows, for the fifth, an octave, a major third and a pair that must be refused. Three formula devices — a gentle curve, a plain squarer and a lopsided clipper — and the soft and hard clippers of the earlier chapters, all at the loudest hardware drive. A phantom that plants every one of the 54 recipes at a ladder of sizes. A plain wire through noise, at two levels and two window lengths. A squarer behind a tone filter and the soft clipper with a filter before it. The old crowded lattice with a squarer and a cubic. The wander fixture, read the old way and through the detector. A squarer driven until its invented tones stand above the notes. And the worked example analyzed 100 samples late.

**The demands.** The numbers here are written into this document by the pipeline. The two programs must agree on every level to within a ten-millionth of a decibel (1e-07 dB) on a capture with no filter in it — on the run that built this document the worst disagreement was about two hundred-millionths (1.8e-08 dB) — and within a thousandth (0.001 dB) through a filter, whose arithmetic rounds differently in the two languages over two million samples (measured about two ten-thousandths, 0.00016); every yes-or-no reading — slot, gate, reading and its reason, grid step, role, shared-slot mark, cross-check — must be identical, and every sentence the record prints identical.

**Against the truth.** The two exact answers must agree with each other to a hundred-millionth of their size (1e-08), and the expanded formula with the integration to a billionth (1e-09). Every recipe that a device cannot produce must read at or below -200 dBc on a noiseless capture (worst measured -273.8) — the depth at which nothing can be told from the loudest tone is set by the arithmetic's precision, not by the window. The app's reading of every recipe a device does produce must match the exact answer, once the noise in the slot is allowed for, within a thousandth of a decibel (0.001) for the smooth devices, within five hundredths (0.05) for the two with a kink in them, and within a hundred-millionth (1e-08) on every phantom.

**The lattice.** The lattice each case played must be the one the manifest recorded by running the app's tuning on that grid, field for field, and the refused pair refused with the same numbers on both sides.

**Noise and latency.** The wire through noise must read every recipe as nothing found, with its capture floor within 3 dB of the value the noise's own statistics predict (measured about two tenths and about a tenth of a decibel short at the two lengths, 0.21 and 0.14) and its coherence excess within 6 dB of what noise alone puts over the floor (measured 6.3 against 7.8). And the late analysis must move the window by exactly its offset, keep the same overrun, and move no recipe's level by more than a hundred-thousandth of a decibel (1e-05; measured about two millionths, 2.2e-06). Part B's parity is established: on every predicate case — the corpus lattice's refusals by position and beside a louder product, the wander fixture through the legacy path and through the detector, and the device whose products stand above its tones — the two implementations agree on every reading, every count and both registers' sentences.

**Folding, and the phantom.** Two of those results are worth stating in words. The two devices with a kink in them — the hard clipper, and the lopsided one — are applied to the signal sample by sample, so what they make above half the sample rate folds back down, and on a two-tone lattice a folded tone lands *exactly* on a low-order slot: about a hundredth of a decibel (0.0128) at f1+4f2 and three hundredths (0.0298) at 7f1, against under a ten-billionth and about two ten-billionths on their phantom twins. That is the earlier chapters' lesson again: a computed device is never the exact answer near the top of the band, and the phantom is. And the phantom of

every recipe reads back 53 present and 1 out of band — the recipe the shipped fifth puts under the band's low edge, whose level still reads back to about three ten-billionths of a decibel and whose reading is a refusal on frequency anyway, by the band rule that comes before the gate.

**The worked example.** The worked example itself, the soft clipper at the loudest hardware drive through noise: 29 recipes measured, 24 with nothing found, a capture floor of -171.2 dBc from 24 empty slots, a coherence excess of 6.3 dB over the local floor against the 7.8 noise alone predicts, and a loudest invented tone at -25.83 dBc. Its two third-order difference tones — twice the low less the high, twice the high less the low — read -25.832 and -25.832 dBc against exact answers of -25.832 and -25.832; three times the low note reads -37.078 against -37.078; and the sum of the two notes, which a symmetric clipper cannot make, reads -173.79 dBc — the noise — and is reported as nothing found, with that level as its bound.

**Noise per slot, drift, and a loud squarer.** The wire through noise also settles a question the limits below rely on: how far a single slot's noise sits below the same noise measured across the whole band. At the standard window it is 58.79 dB (46.75 at the old short one), which is why no broadband floor can be turned into a per-slot decision without an assumption worth tens of decibels. And the wander fixture — a shared drift of both notes, the generator the app's own test uses, which reproduces the one real fault the reference bench has captured — reads the old way at -49.25 dBc, 75.6 dB over the capture floor, with the pair difference 3.56 dB against the 3.56 a shared frequency wander predicts, and through the detector on the shipped lattice at -19.23 dBc, 117 dB over the local floor — far over the bar either way; the squarer driven above its notes has a loudest tone of 20 dBc, clears no ceiling, is not read, and has 49 recipes marked unread under a skirt of 36.8 dB.

**When it runs.** The test runs automatically on every change to the app, and the tables and figures in both documents are regenerated on every change and compared with the committed copies; the scripts and the instructions for running them are in the repository named on the first page.

### 7.8 Where it stops being a measurement

- **The floor cannot say whose tone it is.** A measured tone stands above this recording's noise; whether the pedal or the interface made it, the record does not say. An interface's converters invent a handful of second-order tones of their own. The only thing that separates the two is a capture with nothing in the loop — a null run — and no floor measured across the whole band can stand in for it, because a single slot's noise sits 58.79 dB below the broadband figure at the standard window.
- **The percentage figures carry no floor.** The three sums are made at capture over the tones that passed the gate and are bounded by nothing, so a percentage can be quoted below what the rig as a whole could support. The roles replaced them on the page; the sums remain in the record as they were.
- **A coherence verdict names no culprit, and reads nothing beside a loud tone.** "The notes did not stay stationary" is a statement about this capture; a pedal's own drift earns it as surely as a rig's. And a device whose invented tones stand above its notes is not read at all, because every slot beside its notes is the products'. A device whose slots beside the notes stay at a fixed size while its output rises — falling in dBc as the drive goes up — is consistent with something after its level control wobbling the notes; the record reports

the numbers and the shape, and whether it should say more when that pattern is clean is an open question.

- **The spill is extrapolated, and a fast wobble escapes it.** The per-tone reading extends the outer slots' reads outward along the window's known thinning; a note wobbling faster than about twice per window puts its spill beyond those slots, and the extrapolation under-reads it. Where the model is unsure which kind of wobble it is, it takes the larger spill, which refuses more tones, never fewer.

- **What a wander looks like, and what the shape cannot separate.** A shared drift in frequency puts the high note's pair over the low note's by the amount the record prints; a wobble in level and a timing wobble that treats both notes alike leave the pairs equal, and the shape does not tell those two apart. Slots that fall away steadily with distance from the note are the skirt of a slow, one-way drift; a lift that does not fall with distance is broadband noise.

- **A simulated pedal folds its own tones.** The app's virtual pedal applies its clipping sample by sample, so what it makes above half the sample rate folds back exactly onto low-order slots; the parity numbers above measure how much. A real pedal through a real interface does not do this.

- **The list stops at order 7.** At the standard drive the reference clippers' content sits in the low orders, but a hard clipper driven into its bends has real energy at orders the walk never reaches. An empty row high in the list means measured and nothing found at this drive, never that the pedal makes nothing there.

- **Records made on the old lattice carry its marks.** On the short window the fifth's spacing was 2 slots: 4 recipes sat 2 slots from the played notes and one 2 slots from zero, all refused on position, and tones clustered two slots apart so that the quieter of a pair is refused beside the louder. The older three-slot read applies to those records too. Records made on the original exact fraction share slots outright, marked as shared. None of them can be measured again onto a clean lattice; the marks are the permanent honest reading.

- **The recipe under the band.** On the shipped fifth one recipe, $3f1-2f2$, lands about 1 Hz up — below the band's low edge, where nothing a pedal does can be told from the recording's own drift. It reads out of band on every record, named in its own line, never in a sum and never in a tile; the phantom check above shows the level under it still reads back exactly.

- **The gate is frozen.** A tone's presence and the three sums are written at capture. Changing the gate's factor or its neighborhood moves future captures only, and a reading rule can change which judged slots go into a floor but never which slots were judged. The window's length is not stored either: a reader derives it from the stored signal by the plan's own rule, and an old record whose window ran into its fade answers the question with no bound, never with a certificate that its window was clean.

- **The interval is a few cents off, on purpose.** Keeping the invented tones apart costs pitch inside the stated budget, printed on every record. A consonant interval is a simple fraction and a simple fraction is a crowded lattice, so a request that cannot escape one inside the budget is refused before anything plays rather than measured on a lattice whose labels are not facts.

# 8 Waveform Matrix

## 8.1 What this measurement tells you, and what it cannot

Every earlier chapter turned a recording into an estimate: a ladder of harmonics, a curve with a corner, a map of squares. This one keeps the recording. The waveform matrix — the app's "scope view" — is a grid of small pictures. Each picture, a tile, is the wave that went into the pedal and the wave that came out, over two cycles of one note at one loudness, drawn from the samples the interface delivered and from nothing else. Nothing is fitted, modeled, resampled or cleaned up. Across the grid one way are the notes, from the low E string to the E four octaves above it by default; across it the other way are the loudnesses, from a small signal to the loudest drive the rig allows. So the grid shows how the pedal changes the *shape* of a note across the neck and across a range of playing loudness, as a set of recordings a reader can check any model-drawn wave against.

Because there is no estimate, the method here is not arithmetic but a set of choices about samples: which notes and which loudnesses are measured (the note lattice and the amplitude lattice); what is played (one stepped tone per note — the climbing ladder of Section 4); where in each recording a tile is cut (the zero-crossing start); how the samples are written down (the sample array (encoding)); and the one number kept beside each row, the level of the silence before it (the row noise stamp), so that a quiet tile can be judged against the row's own silence and badged when it does not stand clear of it (the noise margin and badge).

**What it cannot tell you.** Stated once.

1. **Nothing between the grid's points.** A tile is one note at one loudness, and the grid is deliberately sparse: the default grid has 5 notes by 5 loudnesses and takes about 12 s of test signal in 5 rows. What the pedal does a semitone away, or a decibel louder, is not in the record.

2. **Nothing a picture cannot carry.** There is no harmonic level here, no distortion percentage, no knee — only the wave. The chapters that estimate those quantities are where to read them.

3. **A picture's scale is its own.** Every tile is drawn scaled to its own loudest sample, so that a small signal and a large one are both readable. Shapes are compared across the grid; sizes are read from the labels, never from the pictures.

4. **Whether two tiles that look different are different.** A shift in time alone redraws a wave. As the app says beside every grid, a visibly different wave is not automatically an audibly different one.

5. **Whether a quiet tile's fuzz is the pedal's.** The stamp says how far a tile's output stands above the silence measured before its row, and the badge says when it does not. That is a statement about this recording's silence, not about the pedal, and the section on reading a record back says what else that silence can hold.

6. **That a louder tile should look more distorted.** Reading along a row and expecting ever more shaping as the drive rises is natural and often wrong. A soft-knee overdrive's distortion, as a share of the note, peaks at a moderate drive and *falls* toward the loudest, because its gain compresses while its harmonics saturate; a loud tile that looks cleaner than the one beside it can be telling the truth, and the Gain Map (Section 5) is the cross-check.

7. **Nothing about listening.** The grid shows what the circuit did to the wave's shape. It says nothing about what the distortion sounds like.

Everything in this document describes a pedal that has settled; how a pedal changes over time is being measured next, and gets its own chapters when it arrives.

### 8.2 The test signal: a grid of notes and loudnesses

**The notes.** The grid's notes are chosen so that grids of different sizes share their points, and that takes a rule worth a paragraph. The obvious way to place five notes between a low one and a high one is to space them evenly. But five evenly spaced notes and three evenly spaced notes share only their two ends, so a small grid and a large grid of the same pedal could never be laid over each other point for point — measured when the rule was chosen: an even divider's four-note grid lost E4, and its five-note grid lost G♯3 and C5. The app instead builds every grid by refinement (nesting): start from the two ends, and add each further note by splitting the widest remaining gap in the middle, so that a bigger grid always contains the smaller one exactly — three notes inside five inside seven. Every candidate is snapped to the nearest equal-tempered pitch, with one exception the guide's tuning page states: the standard note keeps the app's E2 (82.4 Hz) rather than equal temperament's 82.407 Hz. Two notes are placed before any gap is split. The standard note itself, the note anchor, joins whenever the span holds it, so that every grid shares a note with the Transfer Curve measurement of Section 3 — the next section says what that buys — and the span's middle note, E4 on the default span, is the center the splitting works outward from, one side at a time. On the default E2–E6 span the result is the octave ladder, with two more notes joining at seven:

| $N$ | notes |
|---|---|
| 3 | E2 · E4 · E6 |
| 5 | E2 · E3 · E4 · E5 · E6 |
| 7 | E2 · A♯2 · E3 · E4 · A♯4 · E5 · E6 |

**The loudnesses.** The loudness ladder needs the same guarantee and one more, because a loudness on the computer and a loudness at the pedal are not the same thing. The app's levels are in dBFS, decibels below the interface's full scale — a number on the computer — and two interfaces set to the same dBFS put different voltages on the cable. The standard loudness that the app's other measurements share is therefore fixed as a *voltage* at the pedal's input, 168 mV peak: the voltage anchor. On a rig whose output was measured with a meter during calibration (the volts factor of the earlier chapters), the app works out which dBFS delivers that voltage on *this* rig and puts the anchor there; where no such measurement exists — a plugin, or a rig that was never metered — the anchor sits at -26 dBFS by convention. The quietest loudness is a fixed distance below the anchor, the floor span of 22 dB: 13.3 mV peak at the pedal on any metered rig, and -48 dBFS only on the convention. It was set where it is because the quiet end is where a kind of distortion lives that a louder test cannot see — a dead zone around zero, which takes a large bite out of a small wave and an invisible sliver out of a large one — and the project's reference crossover stage hands over at about 55 mV; the floor sits 12 dB under it. The loudest loudness is the reverse: -12 dBFS on hardware whatever the rig, because it is the interface's own converters that set how loud a probe may be, while a plugin, with no converter to protect, runs to 0 dBFS. Between the floor and the ceiling the anchor joins, and the remaining loudnesses are added by the same splitting rule, one side of the anchor at a time — the side with fewer points first, ties to the quiet side — so that the anchor sits at the exact middle of every grid and the steps are uneven on purpose. The default five are -48, -37, -26, -19, -12 dBFS, the three -48, -26, -12, the seven -48, -42.5, -37, -26, -22.5, -19, -12. That is also why the default is 5 and not 3: the three-point ladder's only quiet point is the floor itself, with nothing in the 22 dB between the floor and the anchor, where dead zones and the onset of clipping live; the five-point ladder puts a point at -37 dBFS, in the middle of that gap.

The evidence that the anchor and the floor are voltages is in the delivered ladders themselves, read from the manifest. On the reference rig, whose full scale is 3.35 V, the quietest column lands at -47.99 dBFS; on a rig whose full scale is 4.827 V it lands at -51.17 dBFS — different numbers on the computer, the same 13.3 mV at the pedal. Figure 35 draws both lattices at every size.

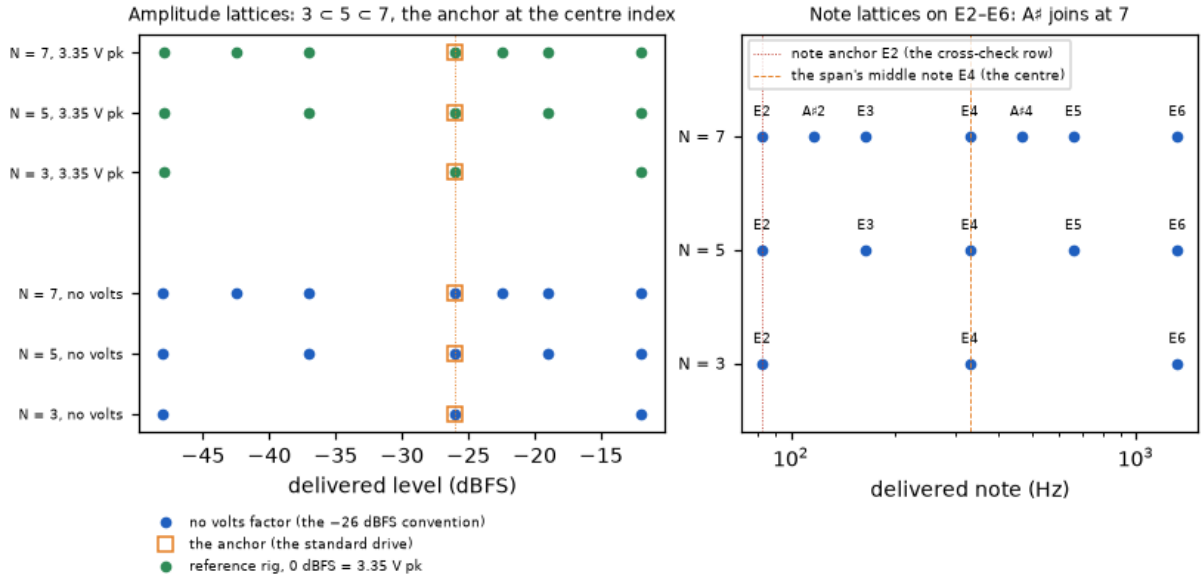



Figure 35: The two lattices at three, five and seven points. Left: the loudness ladder on the no-volts convention and on the reference rig's metered factor (the same ladder, by construction), the anchor squared at the middle of every size, the quietest column a fixed span below it and the loudest column fixed; every smaller ladder's points are among the larger one's. Right: the notes on the default span — the octave ladder at five, two more joining at seven — with E2, the cross-check row (dotted), and E4, the center (dashed): two different rows.

**Each row is one climbing note.** The grid is recorded one note at a time. For each note the app plays 0.2 s of silence, then the stepped tone of Section 4 — the same sine held steady and climbing the loudness ladder quiet to loud, each rung with its settle region and measurement window (150 ms to settle, the first 20 ms of it the ramp up from the rung below, then 250 ms that may be read, trimmed to whole cycles by the cycle snap) and a final fade — and then 0.2 s more of silence, so that whatever the pedal is still doing ends inside the recording. The silence before is the row's own noise window. The 5 rows of the default grid take about 12 s of signal in all, the lowest note first, and the first thing every row plays is its quietest step. Figure 36 shows one row.

## 8.3 How the pictures are cut

Six steps, each in words; the technical reference gives each as a rule or an equation.

**The silence is measured first.** Before any tile is cut, the app reads the level of the row's opening silence — everything that came back while nothing was being played: the loop's own hiss and the pedal's — and keeps it as the row noise stamp. It is the one number stored beside the tiles that is not a sample, and the next section and the last both turn on it.

**A tile is a whole number of cycles.** Each tile holds 2 cycles of the row's note, counted at the recording's sample rate and rounded to the nearest sample — 2 330 samples at E2 and 146

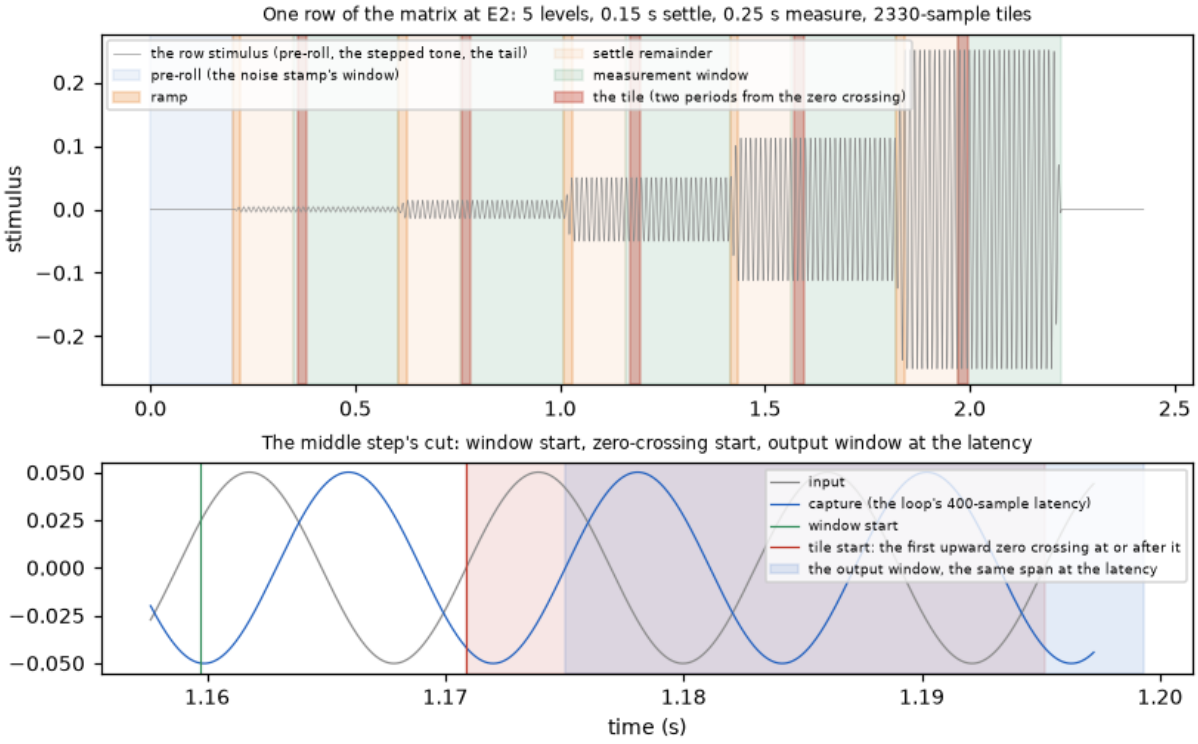



Figure 36: One row of the grid at E2 on the default ladder (top): the opening silence, each rung's ramp, the rest of its settle, its measured stretch, and the two-cycle tile cut from each stretch. Below, the middle rung's cut, close up: the start of the measured stretch, the first upward zero crossing at or after it where the tile begins, and the output's window — the same span, later by the loop's delay (400 samples in this rendering), so that the delay shows as a shift rather than being lined up away.

at E6 at 96 kHz — never stretched to a fixed length. The count is written into the record beside the tile.

**It starts where the input rises through zero.** Inside each rung's measured stretch, the tile begins at the first moment, at or after the stretch's start, where the test tone crosses zero on its way up: the zero-crossing start. So every tile in the grid begins on the same phase of the input, and the pictures line up with each other. It is a choice of where to start reading, not a change to the samples.

**The output is the same stretch of time, after the delay.** Every signal through a rig comes back a little late, and the calibration measured by how much. The output tile is the recording over exactly the input tile's span, that much later. A plugin measures its own delay on every render and hands back a recording with none. Nothing is lined up beyond that: if the pedal itself shifts the wave in time, as every tone filter does, the shift stays in the picture as the output sitting a little behind the input — it is part of what the pedal did.

**A tile that would not fit is refused.** A tile can fail to fit in two ways, and in both the app stops with a message rather than cut a shorter one. If the tile would run into the next rung's ramp it does not fit its stretch — which cannot happen with 2 cycles per tile, since the measured stretch holds at least 21 cycles even at the lowest note, but would with a cycle count no stretch can hold. And if the recording ends before the tile does, the recording was too short, and the message names the calibration's delay as the thing to check. A quietly shortened tile would be a fabricated measurement.

**What is stored.** The samples themselves. Each tile's two arrays — input and output — are written as sample array (encoding)s: the 32-bit numbers the converter delivered, 4 bytes each, with the encoding's name, the channel and the sample count stated beside the data, so that a program reading the record later never has to guess the layout, and refuses one it does not recognize rather than guess. The record also keeps the notes and loudnesses actually delivered — never the ranges asked for — the sample rate, the cycle count, and the row stamps.

## 8.4 What the app decides when it reads a record back

Everything here is worked out from the stored samples each time a record is opened; nothing below is stored.

**A tile's numbers, and the badge.** From each tile's output the app takes three figures: its loudest sample (the peak the tile is drawn to), its average size, and its average value — the last of which is the evidence the final section reads. The noise margin and badge compares the output's average size with the row's stamp. A tile whose output stands less than 6 dB above the stamp — less than about twice the size of the silence — wears the badge, and its caption says, in the app's own words, that most of what the tile shows is the rig's and the device's own noise, not shaping of the tone. It never says the tile is below a floor, because the wave is drawn: what is drawn is a real recording, and the badge says what most of it is. A record with no stamp — one made before the stamp was kept — gets no badge either way, rather than a guess.

**What the silence can hold.** The badge is only as good as its yardstick, and the yardstick is whatever came back during the row's opening silence. Usually that is hiss. It is not always, and the reason is a part every pedal has. Section 3 met the coupling capacitor on a pedal's output,

the part that removes a wave's average level. It does not remove it instantly: a capacitor holds a charge and lets it out over time, so when a wave that carried an average level stops, the capacitor is left holding that level and drains it away over a fraction of a second — a slow, one-way slide back to zero, not a switch. Now recall that the rows are recorded one after another, and that each row ends with its loudest step. A pedal that clips one half of the wave harder than the other — the lopsided clipper of Section 2, or a rectifier outright — makes a wave with an average level, and its loudest step leaves the capacitor charged. The row's own closing silence and the next row's opening silence together last well under a second, and the capacitor is still draining when the next row's silence is measured. The stamp reads that drain, the DC tail, as if it were noise — and it can be tens of decibels louder than the hiss. The stamp, as the floor it is, is gathered with the other measurements' floors in Section 9.

Figure 37 shows it, computed through the check described later on the same made-up rectifier the technical chapter uses: a lopsided clipper behind a slow output capacitor, with a little noise added at the interface. The left panel is the boundary between the first two rows in time. The first row's loudest step ends, its closing silence follows, then the second row's opening silence — the stamp's window — and the start of the second row's quietest step. Drawn twice: once with the device's state carried from one row into the next, the carried state, which is how a real pedal through an interface is measured, because a real capacitor does not reset between rows; and once with every row started fresh, which is how Simulation Mode renders its rows. In the carried rendering the capacitor's drain runs right through both silences; in the fresh one the silences hold only the noise. The right panel is the same stretch as a level: the drain reads 68.4 dB above the injected noise in the second row's window, and the stamp reads it, while the fresh start reads the noise itself to within 0.12 dB. A quiet tile in that row is then judged against a yardstick tens of decibels too high, and a pedal with a longer drain or a smaller output would wear the badge on its quiet tiles for a slow slide that is not noise at all. The first row is never affected, because nothing came before it; the drain is a property of rows two onward, and the limits section returns to it with the numbers the library's own records show.

**The cross-check point.** Because the standard note is on every grid that spans it and the standard loudness is on every grid, one tile — E2 at the standard drive, the cross-check point — was recorded at exactly the note and loudness the Transfer Curve measurement of Section 3 uses, whenever that record used its default note. So the two pictures can be laid over each other: the transfer curve's averaged cycle, and this tile's raw one. They differ a little, and the check described later measures by how much: the transfer curve folds every cycle of its tone into one average, the tone's closing fade included, so its cycle sits a fraction of a percent under the raw wave. That is a property of the earlier instrument, not of this one, and the difference is stated rather than lined up away. The center tile is a different tile: the standard loudness at the span's middle note, E4 by default — the most musically representative cell, which sits at the exact middle of every grid whatever its size.

**Comparing two grids.** When two matrices are compared, tiles are paired by the notes and loudnesses the records actually store, never by where they sit in the grid — a note within 1.7 cents and a loudness within 0.05 dB count as the same point — so a three-point grid and a seven-point grid of the same pedal pair on the nine points they share, and "the same column" never passes for "the same drive". That is value alignment (app level). Two records from one rig pair on their stored dBFS; two metered rigs with different factors pair on the voltage actually delivered at the pedal, where the floor and the anchor line up by construction and the rig-side ceiling generally does not; a plugin record against a hardware record states two different physical quantities with no bridge between them, and no loudness is paired at all — the view says so and shows each grid on its own. Three or more records pair on the points every one of them delivered. And the view always says how many tiles lined up: a silent alignment would be the failure. Those rules live in

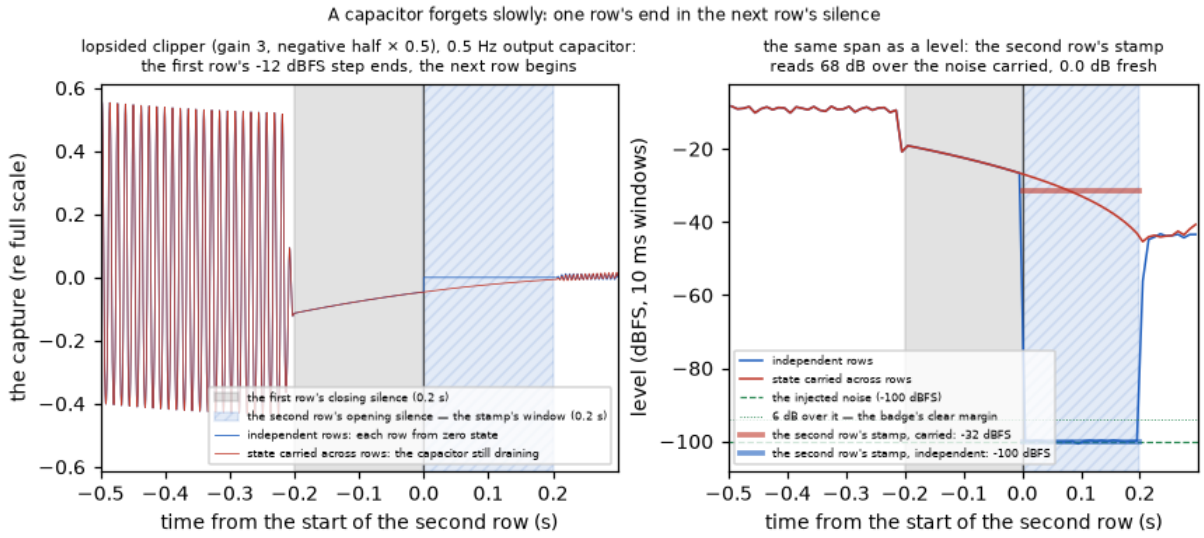



Figure 37: A capacitor forgets slowly. Left: the boundary between the first two rows for the lopsided clipper behind the slow output capacitor named in the title — the first row's loudest step ending, its closing silence (gray), then the second row's opening silence (hatched), the window the stamp is read from. Red: the device's state carried from row to row, as a real pedal experiences it — the capacitor is still draining through both silences. Blue: each row started fresh, Simulation Mode's shape — the silences hold only the noise. The two coincide until the first row's tone ends, since nothing came before the first row. Right: the same stretch as a level in short windows, with the injected noise (dashed), the badge's clear margin over it (dotted), and the second row's stamp under each rendering drawn across the window it is read from.

the app rather than in the measurement code, so the check described later states them and does not run them.

## 8.5 The settings

Every setting is in the technical reference's Waveform Matrix chapter, in one table read from the manifest. This chapter repeats the handful a reader acts on. The grid comes in 3 sizes, 3, 5, 7 points a side, and the default is 5; the default notes run from E2 (82.4 Hz) to E6 (1318.5 Hz); the standard loudness is 168 mV peak at the pedal, or -26 dBFS where the rig was not metered, and the quietest loudness 22 dB below it; the loudest is -12 dBFS on hardware and 0 dBFS on a plugin; each row opens with 0.2 s of silence and closes with 0.2 s, and each rung settles for 0.15 s and is measured for 0.25 s; a tile holds 2 cycles; and a tile wears the badge below 6 dB over its row's stamp.

## 8.6 What the result looks like

The user guide's page for this measurement carries no rendered chart, so this chapter shows no app rendering; Figure 38 is the check's own drawing of a grid, laid out the way the app lays out a tile: the input dashed, the output solid, each tile scaled to its own loudest sample. The device is the made-up crossover stage of the checks — a dead zone around zero of the size the project's reference stage was measured to have — on the default five-by-five grid. Read it a column at a time. In the two quietest columns the whole wave lies inside the dead zone, and the output is a flat line: a tile whose title reads a dead fraction of a hundred percent. At the standard drive, in the middle column, about a fifth of each cycle is dead — the output flattens across zero on every crossing, and the picture is a sine with its middle cut out. At the loudest column only a sliver

of each cycle is dead. Down a column the pictures are the same at every note, because a dead zone does not care what the pitch is; that is the physics the default floor was set to reach, and the reason the quiet columns exist.

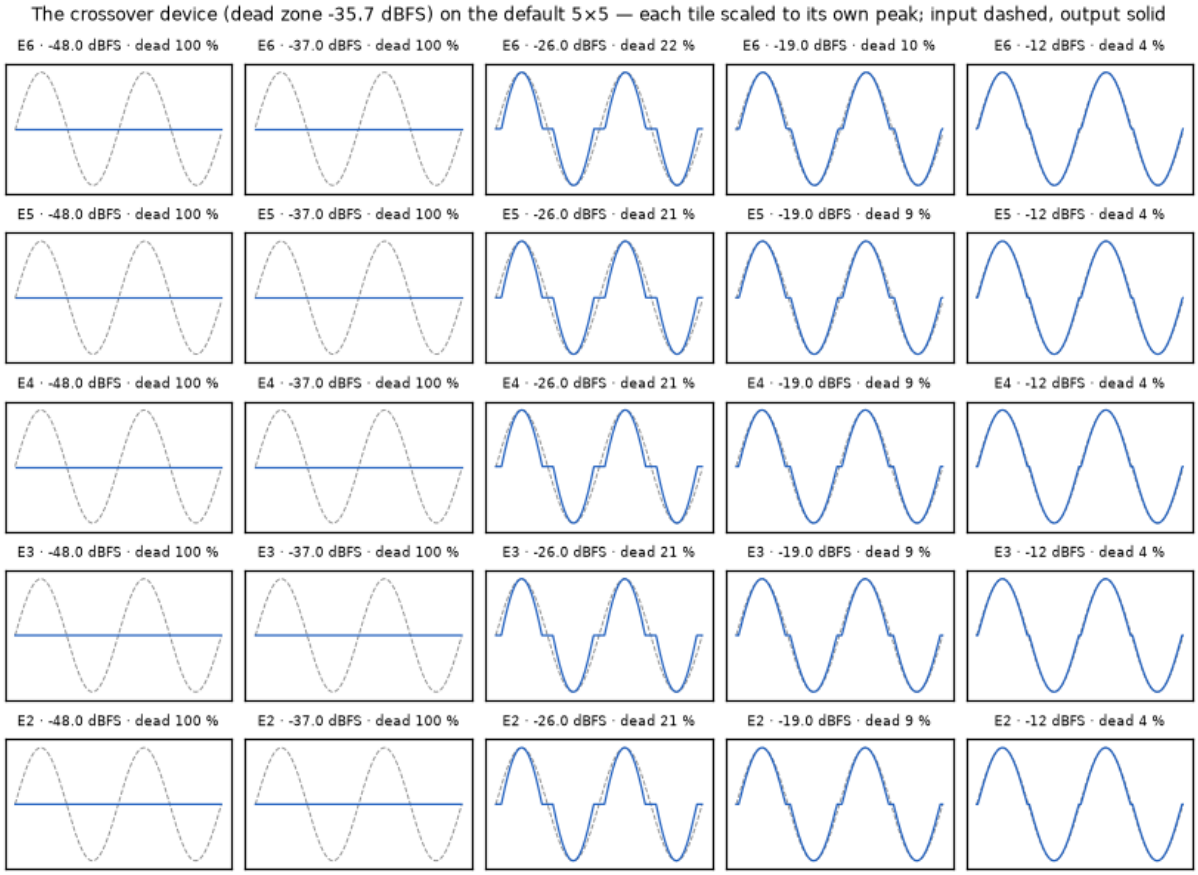



Figure 38: The crossover device — a dead zone around zero — on the default five-by-five grid, drawn the way the app draws a tile: input dashed, output solid, each tile scaled to its own loudest sample. Each tile's title gives its note, its loudness and how much of each cycle fell inside the dead zone. The two quietest columns lie entirely inside it; at the standard drive a fifth of each cycle does; at the loudest a sliver.

## 8.7 How you can check that the app does what this says

As in the earlier chapters, a second program was written from the description — in Python, independently, not by translating the app's code — and the parity test runs both on the same made-up devices through the same made-up loop and demands the same answer. It runs 26 cases. The second program borrows chapter one's devices and noise, the Transfer Curve chapter's tone filters and its exact steady-state answer, and the Compression chapter's climbing tone, and owns the rest: both lattices, the anchor, the row's signal, the cut, the rounding to 32 bits, the numbers read from a tile, the stamp, the badge, the crossover device, and the two ways of rendering the rows. The known answer is exact here in a way it could not be in the earlier chapters. For a device with no memory and no filter, the output tile *should* be nothing more than the device's formula applied to the test tone at the tile's own samples, and that can be computed to the last digit; behind a filter the known answer is the settled, steady-state wave the Transfer Curve chapter computes from the device's formula and the filter's response.

**Equality, not a tolerance.** Where a step is exact the check does not ask for agreement within a tolerance; it asks for equality. The two programs must deliver every lattice at every size and every metered factor — the tie described in the limits included — every note name, every starting sample in the tone and in the recording, every sample count, every badge and every dead fraction *identically*; on the run that built this document 0 of those differed. And on the 14 cases with no memory and no filter, the stored tile's departure from the exact answer must equal, sample for sample, nothing more than the rounding of the exact answer to 32 bits — about six hundred-millionths of the wave's peak (5.9e-08); the departure beyond that rounding was under a ten-billionth of the peak ($\leq$ 1e-10). A tile is the capture, and the check holds it to being exactly that.

**The demands with a tolerance.** The numbers read from a tile — its peak, its average size, its average value — are sums over the same 32-bit samples taken in a different order by the two programs, and must agree to within a trillionth of their value (1e-12) with no filter in the loop (the worst measured disagreement was under a ten-billionth, $\leq$ 1e-10) and to within a ten-millionth (1e-07) through a filter, whose arithmetic rounds differently in the two languages over a long recording (measured about a hundred-millionth, 1.2e-08). The stamp must read the injected noise to within three tenths of a decibel (0.3 dB; measured -0.12 to 0.0617 dB over the five rows of a plain wire through noise — the estimate's own scatter over 19 200 samples of silence), and no tile of that wire may wear the badge. The same device and the same noise through a loop turned up by 12 dB must leave every lattice value, every starting sample and every badge unchanged and move every level by exactly 12 dB (measured departure four hundred-millionths of a decibel, 4e-08): turning the interface up does not move a picture. A recording 100 samples late, with the tiles cut 100 samples later, must read every number identically (measured under a ten-billionth, $\leq$ 1e-10). The crossover device's quietest tile must lie entirely inside the dead zone (a dead fraction of 1) and its loudest mostly outside it (0.0412), the fraction falling steadily along the row (1, 1, 0.212, 0.0936, 0.0412 on the E2 row) and equal to the exact answer's on every row — the requirement the app's own test of this measurement was founded on, carried through the whole pipeline. The four refusals — an unrecognized encoding, a sample count that disagrees with the data, a recording one sample short of its last tile, and a cycle count no stretch can hold — must be refused by both programs with the same message. And the record's size: a seven-point grid on the default span encodes to 506 597 bytes and a five-point one to 248 648, inside the bound the app's own code pins; the two programs' encodings may differ by at most 8 bytes, the length of one number's last digits. Part B's parity is established: the delivered lattices at every dimension and factor (the tie included), every start index, the two decoder refusals and the two extraction refusals with their messages, the stamp, every margin and every badge agree between the two implementations on every case.

**The settle window.** Each rung's first 150 ms are not read, so that the pedal can settle at the new loudness before the tile is cut. The check measures what that window leaves behind. Behind a fast tone filter — a low-pass at a thousand hertz, or a high-pass just under the guitar's range — the tile's departure from the steady-state answer, once the 32-bit rounding is taken off, is under a ten-billionth of the peak ($\leq$ 1e-10): the transient is entirely absorbed. Behind a slow output capacitor it is not: at a corner of two hertz the tile still carries about eight hundred-thousandths of the peak (8.3e-05, -81.7 dB down) of the rung's start-up, and at half a hertz about a thousandth (0.0013, -57.6 dB). Figure 39 shows the three side by side at every loudness of the E2 row. The residue is stated, never removed: the tile is the capture.

**A one-sample mis-cut.** If the calibration's delay were wrong by one sample, the output window would be one sample late. The check plants exactly that error and measures the residue against the exact answer: 1.12 % of the peak at E2 and 17.9 % at E6, because one sample is a

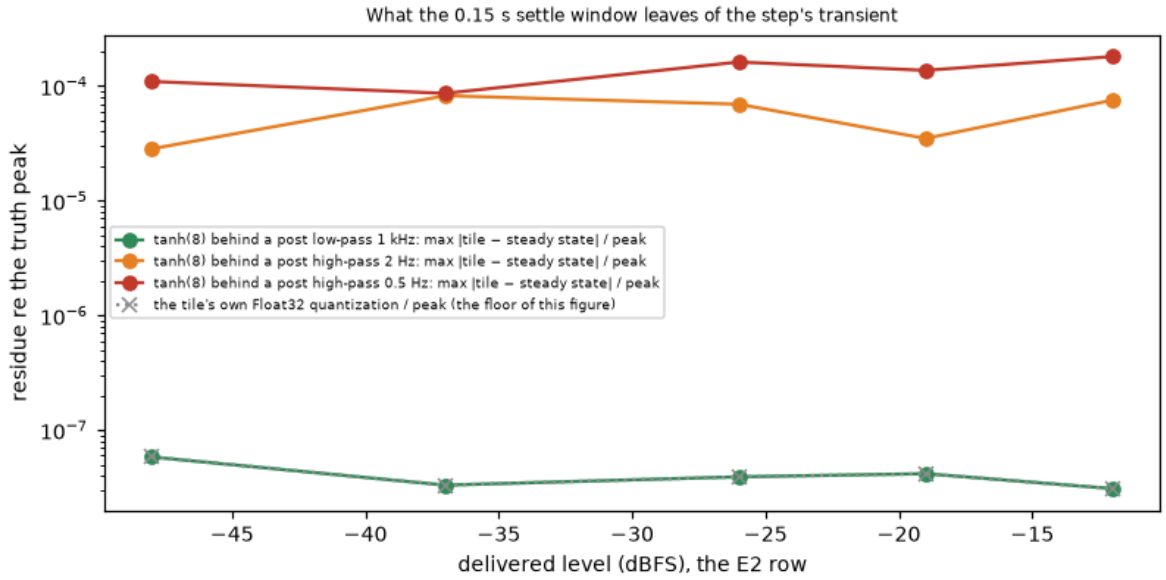



Figure 39: What the settle window leaves of a rung's start-up, at every loudness of the E2 row: the tile's largest departure from the steady-state answer as a share of the peak, for the steep soft clipper behind a fast low-pass (absorbed to the rounding, drawn as the floor) and behind a two-hertz and a half-hertz output capacitor, whose drain is as slow as the window or slower.

larger slice of a short cycle than of a long one, and on a flattened wave it moves the edges. That is why the starting sample is held to equality above, and why the delay is measured rather than guessed.

**The drain, measured.** The lopsided clipper behind the slow output capacitor of Figure 37, through a loop with a little noise, is the case that reproduces the finding the limits section states. With the device's state carried across the rows, the first row's stamp reads the injected noise to within about seven thousandths of a decibel (0.0074) and rows two to five read 68.4 to 68.6 dB *over* it — about 2 641 times the noise — while the same device on rows each started fresh reads every row within 0.12 dB of the noise. Behind a two-hertz capacitor the carried rows read 61.3 dB over, about 1 164 times the noise, and the fresh ones within 0.12. The quiet tiles tell the same story from the picture's side: their departure from the steady-state answer is 1.02 of the peak on the carried rows against 0.0919 on the fresh ones — the drain reaches the tiles as well as the stamp. Figure 40 is the row-by-row picture.

**The cross-check, measured.** A plain wire's tile at the cross-check point, read at the same positions in the cycle the transfer curve reads, differs from the transfer curve's averaged cycle by 1.18 % of the cycle's peak (-38.5 dB below it; the check requires under two hundredths of the peak, 0.02). A wire has nothing to differ by, so that is the frame: the transfer curve averages every cycle of its tone including the closing fade, and its cycle sits under the tone by the fade's share — -0.0324 dB on the input cycle. The steep soft clipper of the worked examples differs by 1.15 %, the same size. So at the one point the two measurements share, they agree to the difference between their frames, measured once, and that is the actual content of "the same operating point".

The test runs automatically on every change to the app, and the tables and figures in both documents are regenerated on every change and compared with the committed copies; the scripts and the instructions for running them are in the repository named on the first page.

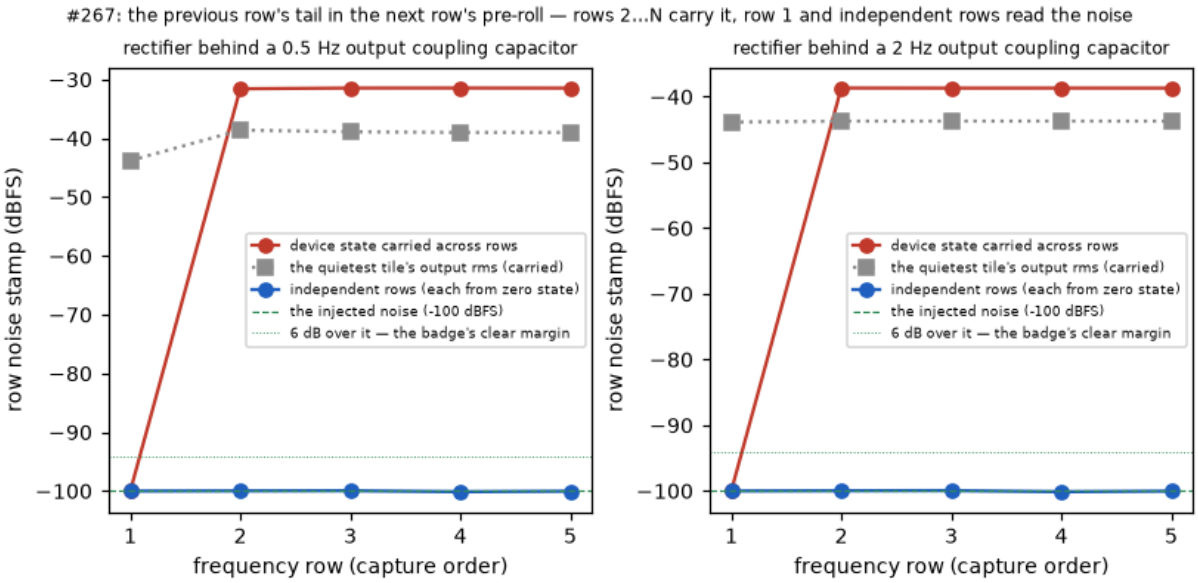



Figure 40: The drain, row by row: the stamp of every row for the lopsided clipper behind a half-hertz (left) and a two-hertz (right) output capacitor, with noise injected at the interface. With the device's state carried across the rows (red) the first row reads the noise and every later row reads the previous row's loudest step still draining, tens of decibels over it; with every row started fresh (blue) each row reads the noise. Gray: the quietest tile's own level on the carried rows, which sits under the contaminated stamp.

## 8.8 Where it stops being a measurement

- **The row's silence can hold the previous row's drain.** A stamp is a noise figure only when the silence holds noise. On a pedal whose output passes through a coupling capacitor and whose loudest step leaves a wave with an average level — a lopsided clipper, a rectifier — the previous row's last step is still draining into the next row's silence, and the stamp reads the drain. The project's own reference positions show it on two dates: rows two onward read 8–16 times the first row's stamp into an interface of about 30 kΩ, and 100–230 times into one of 1 MΩ, because a lighter load lets the capacitor drain more slowly. The quiet tiles of such rows are then judged against a yardstick far too high, and a pedal with a longer drain or a smaller output would wear the badge on its quiet tiles for a slide that is not noise. Simulation Mode starts each row fresh, so it cannot show this. Whether the stamp should remove the slide before it is read, the silence be made longer, or the slide be stated beside the stamp is undecided and filed; nothing about the capture was changed here.

- **The settle window is a duration, and a capacitor can be slower.** Behind a fast filter the window absorbs a rung's start-up to the rounding; behind a slow output capacitor it leaves the residue the check measures, and a rectifier's drain at every change of loudness runs through the whole tile. The residue is stated, never removed.

- **A wrong delay moves the output picture.** The output window sits where the calibration's delay puts it, and a delay wrong by one sample moves it by one sample — 1.12 % of the peak at E2, 17.9 % at E6. The delay is measured once per rig state, and the picture keeps the pedal's own shift precisely because it is not lined up.

- **The tie.** At the 7-point size the splitting rule has to choose between two gaps of exactly equal width, and until late September 2026 the choice fell to the last digits of the arithmetic: two very close metered factors on one interface (4.827 and 4.828 V at full scale, both in the

project's library) gave two different sets of quiet loudnesses. The rule now treats two gaps that differ by less than a billionth of a decibel as the same gap and splits the quieter one, so both factors give the same grid. Grids recorded before the change keep their loudnesses — a recorded grid never moves — and the paired comparison still finds the points an old grid and a new one share (five of the seven loudnesses, on one interface).

- **The cross-check reads the frame.** At the one point the two measurements share, they agree to the transfer curve's own closing-fade deficit; a reader comparing a tile with the transfer curve's picture by eye is comparing against a cycle a fraction of a percent low.
- **One point per tile, and the grid is sparse.** Nothing between two points is measured, and the three-point grid has no point at all in the 22 dB between the floor and the standard drive — which is why the default is 5. Simulation Mode's quick preview draws the 3-point grid, nested inside the default.
- **One capture per row, and nothing is averaged.** A row is recorded once. Noise in a tile is the noise of that recording, and a pedal that drifts between rows shows it as a difference between rows the grid draws and does not name.
- **A picture is not a listening test.** Every tile is drawn scaled to its own loudest sample, and a shift in time alone redraws a wave; as the app states beside the grid, a visibly different wave is not automatically an audibly different one.

# 9 Floors: how deep a reading can see

Every chapter before this one has met a floor: a level under which the number the app reads is no longer the pedal's. Each met it where its own measurement needed it and said what it had to in its own words. This chapter is where the one idea underneath those passages is stated once, and where a reader who has read only one of the measurement chapters can see how the others apply it. It re-explains none of their machinery; where a chapter's rule matters here it is pointed at, not restated at length.

## 9.1 What a floor is

Nothing quieter than the rig can be seen through the rig. A recording carries the interface's own hiss and its own small distortion, and the pedal's hiss on top; a reading that is no larger than what those alone could have produced cannot be told from them, however precisely it was read. The measurement floor is that level — the point below which a reading would describe something that is not the pedal — and the app's rule at or under it is the same everywhere: the reading is reported as a bound, "at most this much", never as a figure. A figure says what the pedal did; a bound says how far the instrument could see. Section 2 states this for the Harmonic Distortion sweep, and every other measurement inherits it.

A floor is not one number, and it is not a property of the pedal. It depends on which rig made the recording, how loud the note was, which note, and — for a swept measurement — how many sweeps were combined. That is why the app never carries a floor as a fixed setting: every floor it uses is measured, and the next section says from what.

## 9.2 Where floors come from

The six measurements draw their floors from three kinds of source, and a reader who knows which kind a floor is knows what it can and cannot see.

**The rig, measured with nothing in it.** During calibration the app plays its sweep through the interface with a cable in place of the pedal and keeps, inside every record made under that calibration, how much distortion and noise the bare loop showed at each pitch: the calibration stamp. The Harmonic Distortion sweep reads its floor from it (Section 2, "The floor"); the Gain Map reads it too, through a median read of 7 neighboring points, so that one unlucky reading cannot make the loop look quieter than it is (Section 5, "A floor, and a margin"). Two relatives belong to the same family. A null-run reference is the loop measured with nothing in it by the *same* measurement as the record — for Compression, the same ladder of loudnesses — under exactly the record's calibration; it says how much distortion the rig itself adds at each loudness, not only how much noise (Section 4, "The hiss line"). And a plugin has no rig at all: its calibration is a loop inside the computer, and it is numerically silent, so it measures nothing the plugin does. In its place the app measures the plugin's own noise before the run and lets that operative floor stand where the loop's hiss would have stood (Sections 4 and 5).

**This capture, from its own silence or its own empty slots.** A calibration made with nothing in the loop cannot carry the pedal's hiss. So the measurements that play a tone rather than a sweep measure the silence before the tone in the same recording — the loop's hiss and the pedal's own, together — and keep that as their floor. Compression's noise stamp is the loudness of the 0.2 s of silence before its ladder starts (Section 4, "The silence before"); the Waveform Matrix's row noise stamp is the same reading taken once per row (Section 8, "The silence is measured first"). Chord IMD has no silence to read, so it reads the empty slots of its

own spectrum instead: the local floor beside each invented tone, from slots that hold nothing the lattice claims, and the capture floor, a summary over every slot the gate did not pass (Section 7, "The local floor, and the gate" and "The capture floor"). A floor of this kind sees what the rig's calibration cannot; the price is that it sees only this capture, and whatever else the silence happened to hold — the last section returns to that.

**The analysis itself.** The third kind is the one a noise floor cannot see at all. An analysis can report a little distortion with nothing in the loop, because its own steps leave traces: the Gain Map's analyzer residue is what the sweep's opening ramp leaks into the window that reads the second harmonic, worked out afresh for each map by handing the sweep straight back as its own recording (Section 5, "A floor, and a margin"). It is a fixed share of the note, the same at every loudness, which is exactly why no amount of hiss can bound it. The same family holds the limit behind fundamental presence: a loud harmonic leaks a little into the window that reads the note, so the app has measured how far below the loudest harmonic a reading of the note can sit before it is the analysis's own leakage — 76 dB — and stops dividing by the note beyond it (Section 2). And in a spectrum, a note that does not fit its slot spills a skirt over its neighbors, so the slots nearest a played note are never read as a floor; that is Section 6's picture, and the Chord IMD chapter's reason for cutting the nearest slots out.

### 9.3 An amount, not a share

The idea that ties the families together is a difference in units. Hiss is an *amount*: so many microvolts, the same whether the note is loud or quiet. A distortion reading is a *share*: the overtones compared with the note, a ratio. So the same hiss is a larger share of a quieter note, and every floor that is stated as a share rises as the note gets quieter — at the quiet end of a ladder, at a quiet square of a map, for a sweep played more softly than the calibration was. The Harmonic Distortion floor carries an allowance for the sweep's level against the calibration's; the Gain Map's drive term raises a square's floor by however much quieter its note arrived than the calibration's did (and never lowers it for a louder one, because the rig's own distortion grows with drive and the stamp measured noise and distortion together at one level); Compression's noise bound is the stamp turned into the largest distortion reading the hiss alone could fake at each rung, climbing toward the quiet end. Three names, one arithmetic.

Figure 41 draws it. The device is a made-up one whose only distortion is a third harmonic at a fixed share of the note, 2 % (34 dB under it) at every loudness, played down the Compression ladder through white noise at -100 dBFS and read by the same code the Compression chapter's check uses. On the left everything is an amount: the note falls rung by rung, the harmonic falls with it a fixed distance below, and the hiss is a flat line. On the right the same rungs are shares: the reading sits at the device's true share, the noise bound rises as the note falls, and at about -63 dBFS the two meet. From there down the app reports a bound — the device really does add 2 %, and the picture shows the reading still saying so, but the rig could have faked that much, so the honest claim is "at most the bound", not the figure. The band above the line is where a reading is a mix of the two, and it is the same band the Compression chapter draws (Section 4, "The hiss line").

### 9.4 The largest binds

Where more than one floor applies to a reading, the app judges it against the largest. Compression draws its hiss line as the largest of the capture's own noise bound, the null run's bound and a plugin's operative floor; the Gain Map's composed margin is a square's distance above the larger of the rig's noise and the analysis's residue; the Harmonic Distortion floor takes the larger of the calibration's stamp and a plugin's operative floor. The reason is the same each time: every one of

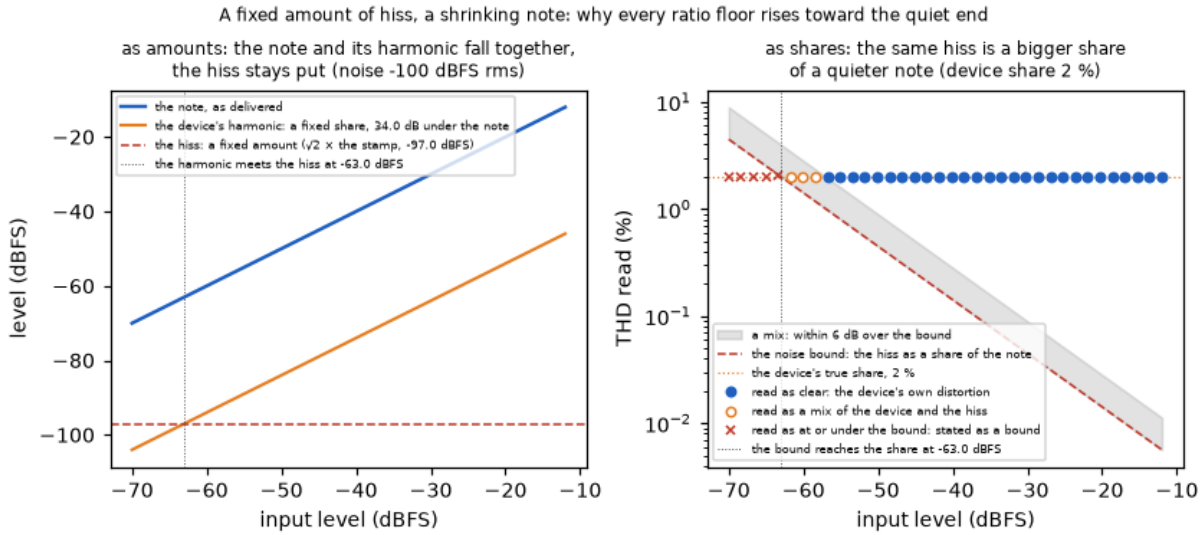



Figure 41: A fixed amount of hiss, a shrinking note. A made-up device whose only distortion is a third harmonic at the fixed share the title states, down the Compression ladder through the white noise the title states, read by the reimplementation's own Compression code. Left, as amounts: the note and its harmonic fall together and the hiss stays put; the dotted line is where the harmonic meets it. Right, as shares: the reading, flat at the device's true share; the noise bound, rising as the note falls; the band above it where a reading is a mix; and each rung marked by its floor class — clear, a mix, or at or under the bound and stated as a bound.

those floors bounds something that is *not* this pedal's distortion, so adding a floor can only move a reading toward "that is the rig", and taking the largest is the careful choice, never a lenient one. Figure 24 in the Gain Map chapter shows the composition at one note, and Figure 18 in the Compression chapter shows the line drawn as the larger of two.

The label follows the same logic. Compression's floor source names a companion floor only where it actually sets the line somewhere: a null run that sits under this capture's own hiss everywhere changed nothing, and naming it would tell the reader they were looking at a measurement they were not.

## 9.5 What a reading near a floor becomes

Each measurement turns the distance between a reading and its floor into a small number of states, and each states its own bar. The bars are not one rule shared between chapters; each is its own constant, quoted here through its own chapter's number, and the fact that several happen to be the same size is a coincidence this chapter does not lean on.

| measurement | the floor | the bar | under it, a reading becomes |
|---|---|---|---|
| Harmonic Distortion | the calibration stamp, adjusted for the sweep's level and its averaging; a plugin's operative floor | at the floor | a bound on that harmonic at that note; in the comparison between two records both readings are lifted to the floor they share before they are compared, so two readings both under it count as the same |
| Transfer Curve | none measured of its own: the note's presence is its one check (the phase removal's bar for which harmonics may vote, -46 dB relative to the note, is a fixed level, not a floor) | — | no memory verdict, no polarity, no pairing verdict |
| Compression | this capture's noise stamp, the largest of its companions | 6 dB | a mix within the bar, drawn dotted; the rig at or under the line (within 0.1 dB), drawn faint |
| Gain Map | the stamp through the median with the drive term, or the operative floor, and the analysis's residue | 6 dB | a square that is not clear; a peak tile quoted as a bound when no square with a neighbor clears |
| Chord IMD | the local floor beside each invented tone | 4 times it (12 dB) | absent, bounded by the capture floor; no floor at all under 3 empty slots |
| Waveform Matrix | the row's noise stamp | 6 dB | the tile wears the badge: most of the drawn wave is noise, never "below a floor" |

Two of the states deserve a sentence each. Compression's three floor classes — clear, a mix, the rig — are drawn as floor runs, solid, dotted and faint, so the trace itself says how far each stretch stands above the hiss (Section 4). Chord IMD's presence gate is decided once, at capture, and written into the record; the reading rules that come later can refuse a slot for other reasons, but they can never change whether it passed (Section 7). The Waveform Matrix's noise margin and badge is the one state that is not a bound at all, because the tile is a drawing of a real recording: the badge says what most of the drawing is, and the app's own sentence never says the tile is below a floor (Section 8). Compression's floor-bound zone is the same idea applied to a corner rather than a reading: a knee found on the bottom rung might really lie lower, so it is reported as "at or below".

### 9.6 Averaging lowers noise, not residue

Playing the Harmonic Distortion sweep several times and combining the results lowers the noise in the reading, and the floor follows it down: 4 sweeps, the app's default, lower the noise part of the floor by about 6 dB, eight by about 9 dB (Section 2, "The floor" and "Averaging"). But the lowering applies only where what the rig contributes is noise. At the faded ends of the sweep, where the residue is the measurement's own, a repeat carries the same residue again, and averaging buys nothing; the floor stays put there. That is the general rule, and it is why the third family of floors is separate from the other two: sweep averaging helps against whatever varies from one repeat to the next, and a floor the analysis makes for itself does not vary. The Gain Map's residue, the separation behind the presence check and a note's skirt in a spectrum are all of that kind. So is a plugin's operative floor, in the part of it that measures a chain not repeating itself: that is not noise a repeat could average away, and the app never lowers it.

## 9.7 Above every floor and still not the pedal

A floor is necessary and it is not sufficient. A reading can stand well clear of every floor the previous sections named and still not be something the circuit did. Three of the chapters carry a check for exactly that case.

The Harmonic Distortion sweep's harmonic shaping check asks whether a harmonic's readings across neighboring notes are orderly, as a circuit's would be, or scattered, as noise is — a coherence score over a run of neighboring readings, called shaped only when the orderliness stands at least 6 dB above the noise in it (Section 2, "Shaping"). A symmetric clipper's even harmonics are the standing example: they are exactly absent, so what the app reads there is the pedal's own hiss, often well above the rig's floor, and still not distortion. The wording matters, and this chapter is where it is easiest to get wrong: a reading that fails the check is noise that is not being deterministically shaped by the circuit. It is not "below the floor" and it is not "too quiet to measure" — both are false, since it stands above the floor. Chord IMD's coherence check asks a different question with the same shape: not whether a tone is above the noise beside it, but whether the played notes stayed still across the whole window, since if they did not, the notes' spill reaches across the record and each invented tone is read only where it stands clear of it (Section 7, "The coherence check"). And the Waveform Matrix's yardstick can hold something that is not hiss at all: a DC tail, the previous row's loudest step still draining through the pedal's output capacitor into this row's silence (Section 8, "What the silence can hold"). The stamp reads it as noise and judges the row's quiet tiles against a yardstick that is too high — a floor that is accurately measured and still wrong about what it measured.

## 9.8 Floors are measured, never computed

Every floor in this chapter comes from a capture: a calibration, a null run, this record's own silence or empty slots, or the analysis run on a known signal with nothing in the loop. None is worked out from the instrument's settings. A floor computed from what the rig is supposed to do would be a prediction about the rig, and stated beside a reading it would read as a property of the rig that had been checked. The app therefore also refuses to guess where it has nothing: a record with no stamp gets no floor claim, not an assumed one. A Compression record from before the stamp existed draws with no hiss line; a matrix record with no row stamp gets no badge either way; a Chord IMD capture with fewer than 3 empty slots reports no floor rather than one made from too little; and a Gain Map with no floor information at all states its loudest square as a raw maximum, which the Gain Map chapter's last section names as a limit rather than a feature.

## 9.9 Where it stops

The limits every floor shares, gathered from the six chapters' own lists; each chapter's list has the specifics.

- **A calibration's floor is one rig at one moment.** The stamp is a single reading at each pitch, taken at one loudness on one day; the median read, the drive term and the averaging term stretch it to the record's conditions, and none of them can make it a measurement of a different interface or a different day.
- **A null run joins only under the same calibration.** A loop measured under another calibration measured a different rig state, and a borrowed floor is worse than none; the null run's bound is also never read beyond the loudnesses it actually measured.

- **A plugin's floor is its own noise, not a converter's.** A plugin has no interface in the loop; what stands in for the rig is the noise the plugin itself makes, and the part of it that measures a chain not repeating itself is never averaged down.
- **A floor from a capture's own silence sees only this capture, and sees whatever the silence held.** The pedal's hiss is in it, which is its strength; a capacitor's slow drain from the previous row can be in it too, which is the Waveform Matrix chapter's first limit.
- **A floor cannot say whose distortion a reading is.** A tone that stands clear of this recording's noise was made by the pedal or by the interface; only a capture with nothing in the loop separates the two, which is what a null run is for and why Chord IMD, which has none, says so in its own limits.
- **Above the floor is not the same as the pedal's.** The shaping check, the coherence check and the DC tail are the three cases this chapter named; a reading above every floor is a candidate, and the chapter that owns it says which further checks it passed.

## 10 Glossary, in plain language

**alias image** A recording cannot hold tones above half its sample rate; higher ones fold back down and can land on a harmonic being measured.

**amplitude lattice** The list of loudnesses the grid is measured at: the quietest, the loudest, the standard one in the middle, and the gaps between them filled in so that a smaller grid's loudnesses are among a larger grid's.

**analysis lattice** The grid of frequency slots the two notes and all their invented tones land on, arranged so that no two invented tones share a slot or crowd each other.

**analysis window ($N$)** The long middle stretch of the recording that the analysis actually reads, cut so that neither the start nor the end of the notes is in it.

**analyzer residue** The small amount of distortion the analysis reports even when there is nothing at all in the loop — a property of the test, worked out afresh for each map, and one of the things a square must stand clear of before it may be called a measurement.

**audition model** A separate model of the pedal at each rung of the ladder that the listening panels play through; none of the map's numbers come from it.

**averaged cycle** Every cycle of the note laid on top of the previous ones and averaged, so that noise cancels and the shape stays — the figure is drawn from that one averaged cycle.

**bound** A reading given as "at most this much" or "at least this much" instead of a number, because what was measured cannot pin the value down any tighter — the app saying how far it could see rather than guessing.

**box normalization** Loop areas are stated as a fraction of the box the whole figure fits in, so a tiny signal and a loud one are on the same scale.

**calibration stamp** A note the app keeps with every measurement of how much distortion and noise the bare loop — cable, interface and nothing else — showed at each pitch when it was checked.

**capture floor** How deep this particular recording could see, stated under the table; it describes the recording and never changes a verdict.

**carried state** Treating the rows of the grid as one continuous performance, the way a real pedal experiences them, rather than as separate fresh starts. A pedal measured through an interface is always in this state; Simulation Mode starts every row fresh.

**cell** One square of the map: one loudness, one note, and the distortion the pedal made there.

**center tile** The cell in the very middle of the grid, which is the standard loudness at the middle note of the range whatever size the grid is.

**cleanup reading** Where, going up the ladder, the pedal's distortion first crosses the line the chart draws: a loudness if it does; "never clean" if it is over the line even at the quietest rung; "always clean" if it never reaches it.

**clipping** What a distortion circuit does to a note that is too big for it: the peaks of the wave are flattened or rounded off. A wave whose shape has changed is, by the same token, a wave with new harmonics in it.

**coherence ($\Gamma$)** A score for how orderly a harmonic's measurement is across neighboring frequencies; real circuit behavior is orderly, noise is not.

**coherence check** A check that neither note drifted while the recording ran; if one did, the whole record is smeared and the app says so rather than quoting numbers.

**coherence detector** The slots either side of each note, kept with the record so that a note which drifted during the capture can be caught where its spill lands first; the chapter says how many.

**coherent sampling** Both notes are tuned so they fit the analysis window exactly, which is what lets each invented tone be read cleanly in its own slot.

**composed margin** How far a square's distortion stands above the biggest of the things that could fake it: the loop's noise, the plugin's noise, or the analysis's own residue.

**compression curve** One note played at many loudnesses in turn, from very quiet to very loud, to find where the pedal starts getting louder more slowly than you play.

**corroborated peak** A square that is clearly above the noise and whose neighboring note is too. Real distortion spills onto the next note; a noise blip happens to one square alone.

**coupling capacitor** The capacitor every pedal has on its output. It removes the wave's average level, so a clipper that clips the top and bottom at different heights shows equal heights after it — but the even harmonics that asymmetry makes are still there.

**cross-check point** The one cell of the grid that was measured at the same note and the same loudness as the transfer curve, so the two pictures can be compared.

**cycle snap** The measured stretch is trimmed to a whole number of cycles of the note, so the note fits it exactly and does not smear.

**dBc** A level relative to the note (or tone) that caused it, so that "how big is the distortion" does not depend on how loud the test was.

**dBFS** A level relative to the loudest a digital recording can be; 0 is the maximum and negative numbers are quieter.

**DC tail** The slow leftover a capacitor keeps after a loud lopsided note stops: a level that drains away over a fraction of a second rather than switching off, so the silence before the next row still holds some of it when that row's hiss is measured.

**decibel (dB)** A way of writing how much bigger or smaller one level is than another, on a scale where each step of 20 is a factor of ten in size: 0 means the same, $-20$ means a tenth as big, $-40$ a hundredth, $-60$ a thousandth.

**deconvolution** The step that undoes the sweep, turning the recording into a set of separate blips, one per harmonic.

**delivered fundamental** How loud the note itself actually arrived at the interface for one square of the map, with everything the loop added.

**derotation** Lining up repeated sweeps to the fraction of a sample before combining them, so that a tiny timing drift cannot smear the result.

**drive level** How hard the app drives the pedal for one row of the map: the level of the signal going into it, as your pick and your guitar's volume knob would drive it. The pedal's own knobs are never touched.

**drive term** The allowance for a square of the map being quieter than the loop check was: the same noise is a bigger share of a quieter note. Louder than the check, no allowance is given.

**drive window** The quietest and loudest rungs of the ladder. A real interface must not be driven into clipping, so its top rung is held back; a plugin has no converter to protect and is driven to the very top.

**edge mask** The first and last columns of the map, where the test tone was still fading in or already fading out; what they show is the test, not the pedal, so they are drawn faint and left out of the headline numbers.

**excited band** The range of notes the test tone actually played at full strength; the very first and very last moments are ramped and do not count.

**export grid** The notes the map is drawn at, spaced by equal musical intervals from the lowest the sweep plays to the highest.

**extraction window ($T_k$)** The slice of time around one harmonic's blip that is kept for measuring it; wide enough to hold the blip, narrow enough to exclude its neighbors.

**fine window** Extra, closer-spaced loudness steps over one stretch of the range, for when a feature is too narrow for the ordinary spacing to show.

**floor class** Whether a distortion reading is clearly the pedal, a mix of pedal and hiss, or just the hiss.

**floor runs** The stretches of the distortion trace drawn solid, dotted or faint according to how far above the hiss they stand.

**floor source** A label saying where the hiss line on the chart came from, so the reader knows which measurement they are looking at.

**floor span** How far below the standard loudness the grid's quietest column sits: a fixed distance, so that on any rig whose output has been measured with a meter the quiet tiles land at the same small signal at the pedal.

**floor-bound zone** The bottom rung of the ladder: a corner found there might really lie lower still, so it is reported as "at or below".

**frame fit** Lining the two measurements up in time by one shift that makes every harmonic agree at once; how well they then agree says whether the two records really describe the same device at the same setting.

**fundamental ($f_0$)** The note being played. Distortion shows up as extra tones at whole-number multiples of it.

**fundamental ellipse** The smooth oval a plain tone filter draws by shifting the note in time: real, and not memory, so it is measured separately and subtracted.

**fundamental presence** A check that the note itself is still there in the output; if a device removes the note, ratios to it mean nothing and the app says so.

**gain map** The same test tone played at several loudnesses in turn, from quiet to hot, so that the distortion can be drawn as a map — how hard the pedal is driven up one side, which note along the other.

**harmonic order ($k$)** Which multiple of the note an added tone is: twice the note is the second harmonic, three times is the third, and so on.

**harmonic pulse** After the analysis untangles the recording, each harmonic shows up as its own separate blip, spaced apart in time so it can be measured alone.

**harmonic response ($H_k(f)$)** For each multiple of the note, how much of it the device adds, across the whole range of notes.

**harmonic role** Which of five jobs an invented tone does to the chord you played: thickens it, adds weight below it, adds a color that is in tune with the chord, clashes, or — all together — how much there is.

**harmonic shaping** A check that a measured harmonic is really the circuit's doing and not just the device's hiss showing up where a harmonic would be.

**hinge fit** Drawing the loudness curve as two straight lines joined at a corner, and sliding the corner along until the two lines fit the points best.

**impulse response ($h[n]$)** The device's fingerprint in time: a sharp blip for the note itself and earlier, smaller blips for each harmonic.

**incremental slope** How much louder the output gets for each step louder at the input: one for one is clean, nothing for one is a wall.

**input-normalized** Measured relative to the strength of the test tone, so that the numbers compare across drive levels and rigs.

**instantaneous frequency ($f(t)$)** The pitch the test tone is at, at a given moment.

**intermodulation product ($mf_1 + nf_2$)** An extra tone the pedal makes from the two notes together — a sum, a difference, or a combination of multiples of the two — and the recipe that names which combination it is.

**inverse filter ($\tilde{X}^{-1}(f)$)** The exact mathematical undo of the test tone, written down from its formula rather than measured, so it keeps working above the highest note the tone reached.

**just ratio** The whole-number version of the interval you asked for — three to two for a fifth — which decides where each invented tone falls musically.

**knee** The loudness at which the pedal starts getting louder more slowly than you play — given as a number when the test could see it, and as "at or below" when the pedal was already compressing at the quietest rung.

**lattice snap** The small, disclosed adjustment of the two notes — a few cents — that lets the analysis read every invented tone cleanly.

**lattice spacing ($g$)** How many slots apart any two invented tones must be; the probe is chosen so this is more than a dozen.

**leakage map** A test with a perfectly clean signal, whose only distortion is what the analysis itself invents — measured so that it can be allowed for when checking the real cases.

**level lattice** The ladder of loudnesses the map is measured at, from the quietest rung to the loudest in equal steps of decibels — each rung the same ratio louder than the one below, not the same amount.

**local floor** The noise measured right next to each invented tone, from slots that hold nothing else; a tone counts as present when it stands well above that.

**measurement floor** How quiet a measurement can go before it is measuring the recording equipment instead of the pedal; a number at or under it is stated as a bound — at most this much — never as a value.

**median read** Reading the loop's noise note by taking the middle value of a few neighboring pitches at once, so one unlucky reading cannot make the loop look quieter than it is.

**memory verdict** The app's three-way call on the loop that is left once the tone filter's oval is taken away: small enough to be treated as no memory, large enough to be called memory, or in between — where the app hedges, because a tone filter ahead of the clipper draws the same kind of loop as mild memory and one note cannot tell them apart.

**memoryless lattice** The two positions a harmonic can sit in, relative to the note, in a pedal with no memory and no filtering: exactly in step with it, or exactly reversed. How far each measured harmonic sits from the nearer of the two is what the filters added, and the removal takes that out.

**methods manifest** The list of every setting the app uses to make a measurement, written out by the app's own code so that it cannot be typed wrongly. The documents read their numbers from it.

**noise bound** The largest distortion reading the hiss alone could produce at each loudness; anything under it is the rig, not the pedal.

**noise margin and badge** A note on a quiet tile saying that most of what it shows is hiss rather than the pedal shaping the note.

**noise stamp** How much hiss the loop and the pedal make on their own, measured in the silence just before the note starts and kept with the record.

**nonlinear loop area** The loop that is left once the tone filter's oval is taken away: the part only a real memory effect (or a filter ahead of the clipper, see the phase removal) can draw.

**note anchor** The standard test note, always one of the grid's notes when the grid's range includes it, so the matrix and the transfer curve can be checked against each other at the same note and loudness.

**note lattice** The list of notes the grid is measured at, chosen so that a smaller grid's notes are always among a larger grid's.

**null-run reference** A recording of the loop with nothing in it, used to say how much distortion the rig itself adds at each loudness, scaled to how loud this pedal's note actually was.

**one-bin estimator** The way the app measures how loud the note (or one of its overtones) is inside the measured stretch: it asks for one exact pitch and ignores everything else.

**operative floor** For a plugin, the noise the plugin itself makes, measured before the run, which stands in for the loop noise a hardware rig would have.

**orientation resolution** A check that the picture is the right way up: a plugin measurement can come out upside down, and the pedal's own measured polarity is used to turn it back.

**pairing residual** A single number for how well the two records agree once lined up; too large, and the app declines to apply the correction and says so.

**parity test** A check, run automatically on every change, that a second, independently written program gets the same answer as the app — and that both get the right answer on devices whose behavior is known exactly.

**peak tile** The map's headline number, which says one of three things: the most distortion the pedal clearly made; or only that it made no more than some small amount the rig could not see under; or that no percentage can be quoted at all.

**per-step noise gate** Rungs where the note is lost in the hiss are left out and counted, so the hiss cannot pretend to be a flat, compressed pedal.

**phantom capture** A made-up recording whose distortion is known to the digit, used to check that the analysis reads back exactly what was put in.

**phase removal** Using the pedal's own measured harmonic timing to undo what its tone filters did to the picture, so that what is left open is memory and not filtering.

**polarity** Whether the pedal turns the wave upside down. Stated only when the measurement is sure; otherwise nothing is claimed.

**presence gate** The fixed bar, a set multiple of the noise measured beside it, that an invented tone must clear to count as measured; the chapter quotes the multiple.

**probe grid** The list of loudnesses the test plays, from quietest to loudest with each rung the same ratio louder than the one below, and an optional cluster of closer steps where the user wants a closer look.

**product order ($|m| + |n|$)** How many multiples of the two notes go into an invented tone; the app lists every combination up to seven.

**product reading** What the app says about each invented tone: measured, nothing found down to a stated level, or could not be read — and why.

**rate constant ($L$)** How quickly the test tone climbs; a larger value means a slower, longer sweep.

**read scatter** How far a single measured level might be off simply because of noise, stated alongside the level.

**reduced tone ratio ($a : b$)** The two notes' frequencies as a simplest fraction; a fraction with big enough numbers keeps every invented tone on its own slot.

**refinement (nesting)** Adding points to a grid by splitting its biggest gap, so a bigger grid always contains a smaller one exactly and the two can be compared.

**residual alignment error** A leftover timing error between the test signal and the recording of a fraction of a sample or more; it opens the picture into a loop that is wider for higher notes, and is a property of the rig, not the pedal.

**resolution band** How precisely the corner is known: the range of corner positions that fit the points about as well as the best one.

**row noise stamp** The level of whatever the loop and the pedal put out on their own in the silence just before each row's note starts — usually hiss — kept with the row so its quiet tiles can be judged against it.

**sample array (encoding)** The exact way each tile's numbers are written into the record, labeled so that a program reading them later never has to guess.

**settle region and measurement window** Within each rung, first a short wait for the pedal to settle at the new loudness, then the stretch that is actually measured.

**sideband pair statistic** The two notes' neighboring slots compared with each other, a shape that hints at what kind of drift occurred without claiming to know.

**static curve** The single line drawn through the middle of a loop, taken as the device's clipping shape: the upward and downward passes averaged together.

**stepped tone** The test note itself: one pitch, held steady, climbing the loudness ladder one rung at a time, with a short smooth ramp between rungs.

**summed presence reading** The check that the note itself came through, done from the one distortion number each rung stores; if it fails on most rungs, no knee, cleanup level or percentage is quoted.

**sweep averaging** Playing the sweep several times in a row and combining the results so that random noise partly cancels while the device's behavior adds up.

**synchronization condition** A small adjustment to the sweep's length so that the arithmetic that separates the harmonics works out exactly.

**synchronized swept sine** ($x(t)$) A test tone that glides smoothly from low to high pitch, timed so that the distortion each note produces can be picked apart afterwards.

**tanh** The hyperbolic tangent, the textbook soft clipper: a smooth S-shaped curve that passes small signals through unchanged and bends ever more gently toward a ceiling it never quite reaches. It clips the top and bottom of the wave alike, so it adds only odd harmonics, and they fall away quickly. The worked example in these documents is this curve with the input doubled first.

**tile** One small picture in the grid: the input wave and the output wave over the same couple of cycles, at one note and one loudness.

**total harmonic distortion** (THD) The size of everything the device added at multiples of the note, compared with the note itself, as a single number.

**transfer curve** A picture of what comes out against what went in, drawn while one steady low note plays: a pedal with no memory draws one line, and a loop means the output depends on where the signal has been as well.

**truncated column** A column near the top of the map where some of a note's harmonics were too high to be measured reliably, so its distortion figure is known to be short and is kept out of the headline numbers.

**two-tone measurement** The app plays two notes at once — a power chord's fifth by default — and measures every extra tone the pedal invents from the pair.

**unity guard** A check that the quiet end of the curve really is clean before a corner is believed; if the pedal was already squashing at the quietest step, the corner is reported as "at or below the quietest step".

**unsigned lobe area** How much area the loop encloses, counted lobe by lobe regardless of which way each lobe is traced — a figure-eight counts twice, not zero.

**validity bound** The range of notes over which a given harmonic can be measured reliably; beyond it the recording itself gets in the way.

**value alignment (app level)** When two grids are compared, cells are paired by the note and loudness they actually recorded, not by where they sit in the grid, and the view says how many pairs it found.

**voltage anchor** The standard loudness, fixed as a voltage at the pedal rather than as a number on the computer, so that any rig whose output has been measured with a meter delivers the same signal at the pedal, whatever its gain; on a rig that has not been metered, and on a plugin, a fixed number on the computer stands in for it.

**waveform matrix** The scope view: the actual wave that went into the pedal and the actual wave that came out, recorded at several notes and several loudnesses and shown side by side. Nothing is modeled or cleaned up.

**zero-crossing start** Every tile starts where the input wave crosses zero on the way up, so all the tiles line up and the pedal's own delay shows as a shift of the output.

## Changes since the previous version

What changed in this companion since the copy published before it, newest first.

### 1.1 (43) — 30 September 2026

Against the first public copy, published on 23 September 2026.

- **A new chapter: "Floors: how deep a reading can see".** It sits after the Waveform Matrix chapter and explains what a floor is, where floors come from, why the largest one decides, what a reading near a floor becomes, why averaging lowers noise but not the analysis's own residue, and why a reading above every floor can still be something other than the pedal. Every earlier chapter's floor paragraph points to it.
- **Chord IMD.** When the played notes moved during a capture, the record no longer writes off every invented tone. It works out how much of the notes' spill reaches each one, reads the tones that stand clear of it, and marks only the others unread.
- **Harmonic Distortion.** The very last point of each sweep, where the sweep fades out, is left out of every reading, and the simulated pedals the checks use are run the way the app's Simulation Mode runs them.
- **Transfer Curve.** One more drive note, in the top octave.
- **Waveform Matrix.** Two nearly identical interface settings could give the largest grid two different sets of quiet levels; they now give the same one, and grids recorded before keep theirs.
- **Every chapter.** The chapters now say that everything in this companion describes a pedal that has settled. The sections on how you can check that the app does what it says are broken into short named paragraphs, several passages are reworded, and spelling is American throughout.