MUSIKINFORMATIK & MEDIENTECHNIK

MUSIKWISSENSCHAFTLICHES INSTITUT DER JOHANNES GUTENBERG-UNIVERSITÄT MAINZ

Clarence Barlow

On Musiquantics

Von der Musiquantenlehre translated

Report No. 51, Juni 2012

Herausgeber:
Bereich Musikinformatik, Musikwissenschaftliches Institut
Johannes Gutenberg-Universität Mainz
Postfach 3980
55099 Mainz
ISSN 0941-0309

Clarence Barlow

On Musiquantics

Part I: Texts

Translators 2002 from the original German *Von der Musiquantenlehre*:
Deborah Richards, Jay Schwartz
with the kind support of the Royal Conservatoire The Hague
Thereafter translator and editor Clarence Barlow

Preface to Part I

This work grew out of materials that I assembled for my class in computer music at Cologne Music University from the inception of the class in 1984 until its unfortunate termination in 2005. I named the course *Zur Musiquantik* (an artificial term, implying 'on music and quantity'), whence the English title of this book stems. This material was also in my syllabus at the Royal Conservatoire in The Hague under the same English name from 1990-2006, as it has been since then in my position at the University of California Santa Barbara. Driven by the fascination of the connections between music and mathematics, acoustics, phonetics and computer science, and further by my own preoccupation with the quantification of harmony and metre, over the years I gradually put together 32 chapters with numerous illustrations.

On Musiquantics has been consciously written in a very compact form, each chapter on two pages, more as a concentrated and comprehensive teaching accompaniment than as an autonomous textbook. It also fulfils a role as a reference book, especially if its content has already been assimilated. It is my hope that my former students, who applied so much patience to this material, some of them repeatedly and frequently attending the course, find in it everything they have learned from me, and that for those who have themselves gone into teaching, the book proves to be useful for their own educational work. I also hope that this book, completed after twenty-four years, helps my current and future students to learn the material more easily than it was possible for their predecessors.

I wish to thank all those friends and colleagues who came into contact over the years with the slowly growing book for their valuable suggestions and help. My publisher and dear friend, the now late Prof. Johannes Fritsch (1941-2010), waited with exemplary patience for twenty years for the first release in 2008 of the German version of *On Musiquantics*.

Finally I am most grateful to Prof. Frans de Ruiter, former director of the Royal Conservatoire The Hague, for his support in having the German version translated in 2002 into English (here in its British form) for the use of my students in The Hague. The translation formed a solid base for my continued work on the book, with innumerable corrections and additions and the complete rewriting of Chapters 16-18.

On Musiquantics Part I

Contents

Part II starts after Page 65 of Part I

Looking at a typical digital clock display, one sees that all digits, appearing in the same place, consist of various combinations of the same seven dashes. By individually switching these dashes on or off, the desired form is achieved. Is it possible to achieve other symbols under these conditions? If yes, how many, and what symbols are they?

The answer to these questions can be concluded from Γ01: one sees 128 various symbols, some of which are relatively familiar (viz. 73 shaded and wholly connected ones, reduceable to 28 basic mirrorable forms: 0 12678CGLPUcoru coru ⌐⌐P34AFHY⊢; in addition, all of them are serpentine, with the exception of the last 7).

The (im-)possibility of additional symbols using these seven dashes can be tested as follows:

The seven dashes are first brought into a fixed order of observation (e.g. |ı|¯|_|-|'|❙|ı|'|), so that they can be examined in turn and in the same manner with each combination to be tested. The digits 0 and 1 can be assigned to represent the conditions 'on' and 'off', respectively; the state of all dashes switched off can be written 0000000 – switching on the last dash yields the representation 0000001. If we list out all such combinations of 0 and 1 systematically (e.g. 0000000, 0000001, 0000010, 0000011, 0000100, 0000101, ... 1111001, 1111101, 1111111), we see that there are exactly 128 of them. If instead of the seven dashes only four were to be examined, our enumeration would appear as follows: 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111.

The arrangement of the digits 0 and 1 is called a *binary* number representation (from the Latin *bini* meaning 'in pairs'), because it is based solely on two digits, 0 and 1 – the more common number system employing the ten digits 0 to 9 is called a *decimal system* (from the Latin *decem* = 'ten'). The decimal number 5741 equals the summation 5000+700+40+1 or $5\times10^3 + 7\times10^2 + 4\times10^1 + 1\times10^0$ ($n^0 = 1$ for any number n); the binary number ijklmn correspondingly means $i\times2^5 + j\times2^4 + k\times2^3 + l\times2^2 + m\times2^1 + n\times2^0$ or $32i+16j+8k+4l+2m+n$, bearing in mind that the digits i to n have the value 0 or 1 only. 101101 (binary) is therefore equal to 32+0+8+4+0+1 = 45 (decimal).

Not only are the biological properties of the human hands with their ten fingers decisive for the global decimal system; in the numerals of various cultures of the world, the origins of these symbols can be recognised as the drawings of fingers and hands, as for example in the so-called 'Arabic' numerals common in the Western world.

1 shows one finger pointing upwards, 2 and 3 show two and three fingers pointing left. According to Victor Goldschmidt (1932), 5 depicts an open hand held downward and 0 a clenched fist representing ten – it was not until late in history that the role of the zero was adapted here (note also the dial of older telephones, where 0 triggers ten impulses). Furthermore 4 signifies, among other things, five minus one (‘ᔦ’), 6 depicts a hand with an upright index finger as 5+1 (‘ᖯ’), and 9 shows ten minus one (‘ᑫ’). Finally we see in 7 and 8 – similar to 2 and 3 – again two and three fingers, no doubt of the other added hand.

The decimal and binary number systems differ in their *bases*, 10 for the former and 2 for the latter. Comparable to the 10 decimal digits (0 to 9) there are only 2 binary digits (0 and 1). In two-digit numbers in a number system, each possible first digit can be succeeded by every possible second digit; hence there are $10^2=100$ two-digit decimal numbers and $2^2=4$ two-digit binary numbers. It can thus be seen that the quantity of all n-digit numbers amounts to B^n, where B is the base of the system. The largest n-digit B-based number amounts to B^n-1, e.g. the largest 7-digit binary number $1111111_b = (64+32+16+8+4+2+1)_d = 127_d = 2^7-1$. (The subscripts $_b$ and $_d$ mean ‘binary’ and ‘decimal’ respectively.)

The smallest non-negative number of every base is 0; the highest 7-digit binary number is 1111111_b or 127_d. There are $2^7=128$ distinct 7-digit binary numbers; this indicates that every number between 0_b and 1111111_b (i.e. between 0_d and 127_d) corresponds to an individual binary number and to one of the 128 depicted combinations of the seven dashes.

Bases other than 2 and 10 are not only possible, they are also commonly used (especially in data processing) – the octal system (Latin *octo* = ‘eight’) is based on 8, the more widespread hexadecimal system (*hex* from the Greek for ‘six’) on 16: the values 10_d to 15_d are hereby represented by the symbols A B C D E F, a convention even less creative than the proposal of the American Duodecimal Society in the 1940s that 10_d and 11_d be written as ‘X’ and ‘E’ (spoken *dek* and *el*). New digit symbols would have been more exciting, as e.g. ∩ ᖶ ᒣ ∂ ᒪ ᵹ (∩ ᖶ ᒣ ∂ ᒪ ᵹ in 7-dash representation; by the way, ∩ meant 10 in Ancient Egypt) as well as new non-decimal names of numbers: generally, the names of the numbers 11-19 even bear the word ‘ten’ explicitly, starting for instance with 11 in Italian, Romansh and Romanian, 13 in Germanic languages (the English ‘eleven’ and ‘twelve’ mean ‘one left’ and ‘two left’, respectively – after subtracting ‘ten’) and somewhat higher in other Romanic languages. Finally let’s mention the bases 20 of the Mayans and 60 of the ancient Babylonians.

T02 shows conversions of the numbers 0_d-255_d into binary, octal (with the subscript $_o$) and hexadecimal (subscript $_h$): digit breaks similar to $9_d \rightarrow 10_d$, $99_d \rightarrow 100_d$ are to be found in the binary system for instance at $11_b \rightarrow 100_b$, $1111_b \rightarrow 10000_b$, $11111_b \rightarrow 100000_b$ as well as in the other systems at $7_o \rightarrow 10_o$, $77_o \rightarrow 100_o$, $F_h \rightarrow 10_h$ and so forth. Each octal digit always corresponds to the same combination of three binary digits (e.g. $3_o=011_b$, $33_o=011\ 011_b$, $333_o=011\ 011\ 011_b$), and each hexadecimal digit corresponds to the same combination of four (e.g. $AD_h=1010\ 1101_b$; $DA_h=1101\ 1010_b$). Check the hexadecimal equivalents of the decimal numbers 2781, 57005 and 57007 using an appropriate pocket calculator.

If the variables x and y be interdependent such that for x-values, y-values also exist, a curve can be drawn in a two-dimensional *Cartesian Coordinate System*, invented by René Descartes (1596-1650) – see Γ03. This system specifies single points as number pairs (x,y), x indicating the horizontal distance and y the vertical distance from two axes at right angles to each other, the x-axis $y=0$ and the y-axis $x=0$, which intersect at the *origin* $(0,0)$. Γ03c shows that some x-values have no y-value, e.g. $x=\pm1.57$ in the curve with the *function* $y=\tan(x)$ (see the vertical dashed lines), or $x=0$ in $y=1/x$; the curves 'flee to infinity' and 'return from the other side'. Also for $y=\pm x^{½}$ $(=\pm\sqrt{x})$ in Γ03a and $y=\ln(x)$ in Γ03b there is no y for $x<0$: square roots ($\sqrt{}$) and logarithms ($\ln$) of negative numbers do not exist.

A knowledge of curves proves to be useful, for example, in the determination of an algebraic approximation in empirically derived or compositionally devised processes.

Observe Γ03a: in the square bounded by both axes as well as by $x=1$ and $y=1$ we see that $x^{½}>x>x^2$, a relationship reversed to the upper right outside the square. Γ03b shows $y=e^x$ approaching the x-axis to the left and $y=\ln(x)$ approaching the y-axis downwards; the curves are *asymptotic* (from the Greek *a* = 'not', *sún* = 'with' and *ptotós*, from *píptein* = 'fall', meaning 'not falling together'), the x- and the y-axis respectively serving here as asymptote. The constants e (=2.71828..) and π (=3.14159..) – see below – will be explained later.

Γ03c shows curves with axis-parallel asymptotes: e.g. in $y=1/x$, going inwards from left and right, the asymptote is the y-axis, while going outwards horizontally it is the x-axis; in $y=\tan(x)$ there are many parallel asymptotes $x=1.57n$ (more accurately $=n\pi/2$), where n is odd. In $y=\tanh(x)$ (short for $y=(e^x-e^{-x})/(e^x+e^{-x})$), the asymptotes are $y=-1$ to the left and $y=+1$ to the right. In the case of both the curved lines in Γ03a there are no straight lines which they approach: although at upper right all three lines visibly diverge, for all lines $y\to\infty$ (infinity) if $x\to\infty$. They are therefore not asymptotic. All asymptotes in Γ03b and c are vertical or horizontal; however, non-axis-parallel asymptotes are also possible.

In Γ03c the straight line $y=x$ touches the curves $y=\tanh(x)$ and $y=\tan(x)$ at the origin, all three moving at that point in the same direction; the straight line is *tangent* (from the Latin *tangere*, 'to touch') to both curves at this point. A tangent can touch a smooth curve at any point. In Γ03d the straight line (as a general equation $y=mx+c$) is tangent to the curve that it touches at point (x_p,y_p). The slope or *gradient* (from the Latin *gradus*, here 'degree') of the tangent is represented by m, the constant quotient a/b of the lengths of the two displayed sides a and b of an imaginary right-angled triangle of arbitrary size bounded by the tangent and the x-axis; the c in the equation corresponds to the y-value of the tangent at the y-axis. A horizontal line has the gradient 0; the more the line is raised anticlockwise, the higher the gradient will be, until in a vertical position it tends to ∞. The gradient of a line proceeding downwards to the right is negative.

Γ03e shows the region $0 \leq x \leq 1$, $0 \leq y \leq 1$ in 16 frames, each with three curves, the tangential behaviour of which is represented by the letters F, f, j and J: the curves of the F-row exhibit vertical tangents at (0,0), while those of the F-column show horizontal tangents at (1,1). Inversely, all (0,0)-tangents of the J-row are horizontal while those of the J-column at (1,1) are vertical. The tangents in Γ03e are therefore parallel to the axes. Small letters indicate non-tangents, whereby f and j show a bending that is similar to the form of the letter. Additionally, each frame shows the source equations of the curves, where the basic gradients g and Ϭ (=1/g; I pronounce it 'eej') at (0,0) and (1,1) have been arbitrarily set to g=2, 2½ and 3. For g=2, FF and JJ are quadrants of circles, Ff, fF, Jj and jJ are parabola segments (e.g. 'Ff' means row F, column f; ff-like curves are *convex*, jj-like curves are *concave*).

The equations in the nine frames to the lower left of this collection of curves in Γ03e are *cubic functions* of the form $y=a+bx+cx^2+dx^3$, as in e.g. $y=2x-3x^2+2x^3$ for g=2 in the curve fj; if $d=0$ (as for g=2 in fF and Jj), they are *quadratic functions* of the form $a+bx+cx^2$. Both are *polynomials* of the form $y=ax^0+bx^1+cx^2+dx^3+..$, where a, b, c, d.. are *coefficients* – the highest *power* (i.e. $^{0\ 1\ 2\ 3}$ ...) with a non-zero coefficient is the *degree of the polynomial*. These nine cubic equations have the collective basic form $y=m_0x+(3-2m_0-m_1)x^2+(m_0+m_1-2)x^3$, where m_0 and m_1 are the gradients at (0,0) and (1,1). Of the remaining seven equations, three – Fj, FJ and fJ – are interpolations between those of the block of nine. The other four were chosen arbitrarily.

Differential Calculus involves the degree of change of variables, as for example of the gradient of a curve at a point on the curve (i.e. at the tangent imagined there) as a measure of the rate of change in the y-value: the steeper the tangent, the more rapidly y increases in relation to x. The gradient of a curve is called the 1st *derivative* of y and is notated y' (or dy/dx); the 1st derivative of the gradient is the *curvature* of the curve as well as the 2nd derivative of y and is notated as y'' (or d^2y/dx^2). Hence, velocity (in respect to time) is the 1st derivative of distance, of which acceleration is the 2nd derivative and at the same time the 1st derivative of velocity. In the polynomial $y=a+bx+cx^2+dx^3+..$ the 1st derivative is given by $y'=b+2cx+3dx^2+...$ In general, for $y=mx^n$, $y'=mnx^{n-1}$ and thus $y''=mn(n-1)x^{n-2}$ and so forth. Γ03f shows the curve fj (g=2) and its first and second derivatives as functions of the gradient: the steeper fj is, the higher curve (i), and the more curved fj is, the steeper curve (i) and the higher or lower curve (ii) – a negative y-gradient leads to a lower y'. A convex curve has a negative curvature, a concave curve a positive. The inverse of the derivative is the *integral*, notated in *integral calculus* as $y=\int y'\,dx$.

One method of connecting several points with a smooth-looking curve is the *spline*, a chain of cubic functions with the property that at each of the points three pairs of equal values are given: those of both adjacent functions (y) as well as their gradients (y') and curvature (y'') – see Γ03g.

F04 shows a formula collection useful for the calculation of tempo acceleration by a constant factor in unit time, given the initial and final tempi, the total duration and the total number of beats.

It has been established for some time that there are two ways of perceiving pitch: one way allows us to recognise musical intervals and thus to enjoy tonal music, which predominates world-wide (see the examples in Γ05 – the 'neutral' intervals, halfway between major and minor, that of the fourth between perfect and augmented, have been in use in Western Music at the latest since the early 20th Century); the other way, much more frequently employed, has to do with the feeling of 'high' and 'low' or with 'bright' and 'dark' sounds, but not with intervallic evaluation. There are reasons to believe that this second method originates in the ear, and the first method in the brain. The second method is employed when we recognise the 'melody' or intonation of a language, in some languages essential to comprehension, as e.g. Chinese or Thai; only with repeated listening (e.g. in a looped recording) do we come to recognise the 'pitch melody' therein. The frequencies of language formants, too, the perception of which is fundamental in vowel recognition (an ability related to timbre in general) do not by a long way make us think of melody. Even in pitched music there are many situations in which an intervallic interpretation of the pitches present is undesirable, e.g. in slow, extended glissandi or in pointillism – in these cases it is primarily the second method of hearing which comes into play.

For many thousands of years, basic intervals like the octave, fifth, fourth (also written 8ve, 5th, 4th), etc., have been familiar to musicologists worldwide; the pitches of these intervals were known to possess various degrees of correlation – in Western Europe one spoke of '*consonance*' and '*dissonance*' (from the Latin *con* = 'together', *dis* = 'separate', *sonare* = 'sounding'). Characteristic of the development of pitch material in Europe during charted music history is the gradual insertion – as one says – of increasingly 'dissonant' intervals; this occurred through progressively complexer harmony (e.g. by new, daring interval combinations). The possible conclusion that the earliest-used pitch-set consisted solely of 'consonant' intervals, is obviously false. The use of smaller intervals, like seconds and thirds, as melodic building-blocks is definitely older than that of fourths, fifths, or octaves, known everywhere for their consonance, but intervallically more disjunct (imagine a melody exclusively made up of these intervals!). Furthermore, music theory differentiates beween two classes of interval size: *step* (the interval of a third and smaller) and *leap* (a third and larger): an internet search engine showed on 26.06.02 the following counts for seconds (77 as step:8 as leap), thirds (19:13), fourths (11:36), fifths (14:39), sixths (2:33), sevenths (0:10) as well as octaves (6:108). There are reasons for raising the threshold between these classes in lower pitch regions, as will be explained later.

It is possible that an interaction of both of the described methods of hearing – I would like to call them 'rational-intervallic' and 'pitch-spatial' – led to the making of scales, as explained in the following representation:

Phase 1 – Language intonation as a function of emotion is ritualised and formalised to a kind of sing-song. The range of the movement of pitches is relatively small (perhaps a few steps, up to a third) and is rational-intervallically undefined.

Phase 2 – Through the expansion of the range of the sing-song, more consonant positions within pitch-space are touched on (e.g. the fourth and the fifth), suggesting intervallic connections. This is where rational-intervallic hearing begins. Through the rationalisation of the pitches, scales are formed. Probably the fourth, fifth and octave are intervallically recognised and partially fixed, while the other intervals remain for a time only pitch-spatially relevant, a mixed system.

Phase 3 – Some of the clearer, tuned scale-degrees are felt to be exaggeratedly 'clean' and are then 'clouded', 'colourised', 'enriched' through slight detuning. Here pitch-spatial and rational-intervallic hearing are both employed: next to the rational-intervallically defined scale steps, pitch-spatially determined steps are perhaps also interpreted as approximated intervals, although they are now richer, more vibrant than if tuned purely. It might be of interest to know that the introduction of the first non-diatonic scale-degrees (e.g. B♭ or F♯ into the key of C) was named 'chromaticisation' (from the Greek *khroma* = 'colour'); thus the term *chromatic scale* for the final result of this process.

Phase 4 – The 'colourised' steps are rationalised anew: e.g. the minor third, which possibly resulted from a colouration of the major third, becomes established in its pure tuning (or *just intonation*, as it is generally called), in which it is familiar today.

Everything else is a cyclic, or more likely, a spiraling repetition of phases 3 and 4. In Chinese music, essentially clearly defined intervals like the octave are often detuned to such an extent that they sound vibrant but still function as octaves; this would be an example of phase 3. Through timbral colouring in Arabian music, the 'neutral third' – about half-way between a minor and major third – initially a rational-intervallically undefined pitch-spatial aberration of the other two, then gets drawn into the body of just intonation, an example of phase 4. In this context I would like to bring in theories of the origins of the moon: did the moon form near to and independently of, but simultaneously with the earth out of the same cloud of dust, or was it pulled out of the earth? In the same way, one could imagine the neutral third as an independent occurrence, or instead – following the reasoning given above – as a pitch-spatial detuning of the other thirds, subsequently rationalised.

This speculative explanation suggests incidents that probably – if at all – happened for the most part in human prehistory, or even in prehuman times, and certainly not in this order but rather interlaced. It is definitely conceivable that scale-steps functioning even today as leading tones tend more to pitch-spatial than to rational-intervallic tuning, and that they first gain temporary intervallic relevance through a change in role by modulation.

An *arithmetic series* is a series of numbers that is chacterised by a constant difference between two neighbouring numbers, e.g. 2, 5, 8, 11, 14,... Here the difference is always 3. In a *geometric series* the quotient of two neighbouring numbers is fixed, e.g. 2, 4, 8, 16, 32,... Here the quotient is always 2.

In the *overtone row* or *harmonic series* (see Γ06a) the frequencies of all pitches are whole number multiples of the fundamental frequency, usually given in *Hertz* ('Hz' for short), cycles per second, named after the physicist Heinrich Hertz (1857-1894), e.g. 100 Hz, 200 Hz, 300 Hz, 400 Hz, 500 Hz, and so forth, hence an arithmetic series; the names of the notes approximately corresponding to these frequencies are usually assigned according to the practice of adding an octave number to the note name: e.g. C_4 for the note commonly known as 'Middle-C' (close to 261 Hz; there is some dispute about where to start the numbering – opinions for the Middle-C octave-number vary from 3 to 5; in this book 4 is used, so that the lowest A on the piano is A_0). Strikingly, the overtone row's pitch intervals become progressively smaller ascending, although the difference in frequency remains constant at 100 Hz. The reason for this is that the quotient of two frequencies, not the difference, determines the size of the interval between the frequencies.

If we wanted to assemble a series of equally-distanced notes, we would have to multiply the frequency of each of the notes by a constant factor; in the case of the interval of the octave this factor would be 2, so that starting at the frequency 27.5 Hz (A_0), the notes following called 'A' are at frequencies 55, 110, 220, 440, 880, 1760... Hz: see Γ06b (the names of the notes are given here as well). Hence, this 'exponentially' increasing (geometric) frequency curve results in a seemingly evenly increasing pitch. If we calculate the so-called *natural logarithm* of these frequencies with the 'ln' key of a pocket calculator, we would get the following values: 4.0, 4.7, 5.4, 6.1 etc., numbers with a constant difference of 0.7: the term 'logarithm' is from the Greek *lógos* (= 'ratio') and *arithmós* (= 'number'). This arithmetic *logarithmic* number series offers a better representation of pitch perception than the linear series of the frequency values – this knowledge was impressively established by the scientists Ernst H. Weber (1795-1878) and Gustav T. Fechner (1801-1887) in the mid-19th century: the Weber-Fechner-Law maintains that a barely perceptible increase in sensory stimulation forms a fixed percentage of the initial stimulation and as a result that when physical stimulation increases geometrically, the biological perception of it behaves arithmetically. This law is valid in visual, auditive, tactile and practically all other sensory areas.

Γ03b shows three curves: $y=e^x$, $y=x$, $y=\ln(x)$. The curve of $y=\ln(x)$ looks like a bending towards the lower right of the two ends of the $y=x$ line, the logarithm of which it portrays. According to the laws of logarithms, $y=x$ is the logarithm of $y=e^x$; although the former is a straight line, it also results from a similar type of bending of the latter curve. One could say, the logarithmisation causes a ‘damping’ (‘convexisation’) of a given curve; in the reverse process – linearisation – the curves rise increasingly rapidly with increasing x (‘concavisation’). $y=e^x$ is the linear form of $y=x$, which is the former’s logarithmic form in this context; $y=x$ is the linear form of $\ln(x)$. It can thus be seen that the term ‘linear’ is not synonymous with ‘straight-lined’; here the term *antilogarithm* is sometimes used (the ‘antilog’ of x is e^x).

The origin of the logarithm (generally abbreviated as $\log$) could be illustrated as follows: given the equation $a=b^c$, c is the ‘logarithm of the argument a to the base b’, notated as $\log_b a$ or $\log_b(a)$. The decadic or base-10 logarithm of 2 (which is frequently and misleadingly abbreviated like the general logarithm as $\log$, but also as $\lg$, which is used here) is 0.30103, because $10^{0.30103}=2$. The expression $\lg(u)$ or $\log_{10}(u)$ could be understood as ‘10-to-which-power-is-(u)?’, by way of example. The natural logarithms (abbreviated $\ln$) are those to the base e, a constant with the value 2.7182818284590452353602874... , equal to $(1+{}^1/_n)^n$ where n is very large number (in the first calculations of logarithms, the base e proved to be the simplest to manage; $\ln(a)$ is also equal to the area bounded by $y=1/x$ and the x-axis between $x=1$ and $x=a$, a definition, the simplicity of which warrants the description ‘natural’). Equipped with this form of notation, the Weber-Fechner-Law can be written as follows: $E=k+c\log(R)$, where R represents the stimulus and E the perception (k and c are constants specific to this application).

Some equations:

$\log(mn) = \log(m)+\log(n)$

$\log(m/n) = \log(m)-\log(n)$

$\log(m^n) = n\log(m)$

if $n=\ln(m)$, then $m=e^n$

$\log(1) = 0$

$\log_a(a) = 1$

$\log_a(b) = 1/\log_b(a)$

$\log_a(m) = \log_a(b)\times\log_b(m)$

$\lg(m) = \lg(e)\times\ln(m) = 0.434294\ln(m)$

$a^{\log_a(m)} = m$

The natural logarithm can also be represented as an infinite power series:

$\ln(1+x) = x - x^2/2 + x^3/3 - x^4/4 + ...$

In many phenomena a geometrical parameter change leads to an arithmetic perception in the recipient – measured and perceived values mostly form different scales. A note of which the frequency as a measure of pitch is repeatedly multiplied by a constant factor (forming a geometric series), increases continually by a constant *pitch-interval* (arithmetically). A note of which the *sound intensity* as a measure of loudness is repeatedly multiplied by a constant factor (geometric), always increases by a constant *loudness-interval* (arithmetically). This relationship also appears outside of music: a star, of which the brightness is repeatedly multiplied by a constant factor (geometric), always brightens by a constant *brightness-interval* (arithmetically). Γ07 shows the arithmetic-geometric relationship in these three cases: in each case, the intervals are shown on the x-axis below and the corresponding factors of measurement (not true to scale) on the y-axis at left. As an additional illustration, divisions of a single interval are given at the top and the corresponding factors (here true to scale) on the right, serving as an optical enlargement of the small grey-filled rectangles in the lower left corner of each diagramme. Hence, the correspondence between the arithmetically increasing interval size and the resulting geometrically increasing factors can be seen in each diagramme, in Γ07a and b with the frequency factor 2 and sound intensity factor 10 per given interval – on the right of each we also have in a) the frequency factor $2^{1/5} = \sqrt[5]{2}$ per one-fifth of the basic interval and in b) the sound intensity factor $10^{1/10} = \sqrt[10]{10}$ per one-tenth of the basic interval); the unit of measurement of these basic intervals (octave and bel) will be explained later.

Concerning the brightness of stars: by definition, a star that is 100 times brighter than another star is five *magnitudes* brighter; this is the unit of the intervals shown in Γ07c. A brightness interval of 5 magnitudes corresponds therefore to a brightness ratio of 1:100, and similarly, a brightness interval of 10 magnitudes corresponds to a brightness ratio of 1:10000, because each single increase by a certain interval corresponds to the same geometric factor, in this case 100 times per magnitude. At the same time, a brightness interval of exactly 1 magnitude corresponds to a brightness ratio of $1{:}100^{1/5} = \sqrt[5]{100}$, or approximately 2.512 (see below).

If the factor F corresponds to one magnitude, an increase in a measured value by x such degrees or intervals (x can be a whole number or a fraction) indicates a multiplication of that value by F^x. As seen above, the division of the interval into x equal parts would lead to x consecutive geometric increases, each then to the x^{th} root of F: therefore, the factor corresponding to one single magnitude is $F=100^{1/5}$, because in this case $F^5=100$.

Generally, if an increase in the measured value by G intervallic units (e.g. pitch-interval, loudness-interval, or brightness-interval) causes a rise in the value by a factor of F, then an interval of g units would correspond to a value-quotient q_v of

$$q_v = v_2/v_1 = F^{g/G}$$

where the variable g is the interval in the given units, and v_1 and v_2 are the measured values of the elements forming the interval. Note that the units in which g and G are measured are identical, and that the constants F and G are mutually dependent! $F^{g/G}$ can also be written as $(\sqrt[G]{F})^g$, the g^{th} power of the G^{th} root of F. As a mnemonic help, notice also that F can stand for Factor and g as well as G for deGree.

An example from the field of light (the magnitude value decreases with increasing brightness; magnitude 1 is therefore 100 times brighter than magnitude 6, and negative magnitudes are even brighter):

F and G can be replaced by 100 and 5 respectively, from which the equation $q_b = 100^{g/5}$ results (the subscript $_b$ here means brightness). The brightness of the star Sirius is at magnitude −1.58, while that of the star Alpha Centauri is at magnitude 0.06. The brightness interval is therefore 1.64 magnitudes. Hence, Sirius is $100^{1.64/5} = 100^{0.328} = 4.529$ times brighter than Alpha Centauri.

Conversely, the size of the interval can be derived in the desired unit through the application of the measurements v_1 and v_2 or at least their quotient q_v:

$$g = G\,(\log(q_v)/\log(F))$$

Here, too, the variable g and the constant G have the same units; as above, the constants F and G are mutually dependent.

Another example from the field of light based on the corresponding equation $g = 5(\log(q_b)/\log(100))$:

The sun is 15,850,000,000,000) times brighter than a barely visible star of magnitude 6. Hence the sun is $5(\log(15{,}850{,}000{,}000{,}000)/\log(100)) = 33$ degrees brighter; its magnitude is −27.

At this point it might be interesting to note that whereas in most world languages 1,000,000,000,000 is called a ‘billion’, the English-speaking world is terminologically split – most call this a ‘trillion’. The two systems of enumeration are termed the ‘long scale’ (in which every new term above a million means a multiplication by a million, used in the languages of most non-English-speaking countries of the world and in the English spoken there, as well as by a number of people in the UK) and the ‘short scale’ (in which every new term above a million means a multiplication by a thousand, used in official English in the UK, Ireland, the USA, Australia, New Zealand, and in the languages of a few non-English-speaking countries, such as Brazil).

A sine wave (from the Latin *sinus* = 'fold') is a 'pure' tone, the sound of one single frequency; this is generally measured in Hertz, the number of oscillations per second. An arithmetically increasing series of Hertz frequencies does not lead to a perception of equidistant pitches; it is rather a geometric series that achieves this, as in e.g. the octave-series ...110 220 440 880... Hz. An octave-leap upwards, traditionally divisible into 12 *equal-tempered* semitones, or 1200 *cents* (from the Latin *centum* = 'hundred', abbreviated 'Ct'; 1 semitone = 100 Ct), corresponds to an exact doubling of the frequency; thus if the increase in frequency of 1200 cents causes a twofold increase of the same, then an interval of c cents corresponds to a frequency-quotient q_f of

$$q_f = f_2/f_1 = 2^{c/1200}$$

Here, c is the interval in cents, f_1 and f_2 are the frequencies in Hertz of the pitches making up the interval. Notice: the unit of c is identical with that of the power denominator 1200 ($q_f = 2^{h/12}$ holds for the interval h in semitones, $q_f = 2^{\Omega}$ for the interval Ω in octaves).

An example: Find the frequency of C_5, 300 Ct above A_4 (440 Hz).
The solution: $f_2 = f_1 \times 2^{c/1200} = 440\ \text{Hz} \times 2^{300/1200} = 440\ \text{Hz} \times 2^{1/4} = 440\ \text{Hz} \times 1.1892 = 523.25\ \text{Hz}$

Another example: Find the frequency of C_4, 900 Ct <u>below</u> A_4 (440 Hz).
Solution: $f_2 = f_1 \times 2^{c/1200} = 440\ \text{Hz} \times 2^{-900/1200} = 440\ \text{Hz} \times 2^{-3/4} = 440\ \text{Hz} \times 0.5946 = 261.63$ Hz.
It should be no surprise that this frequency amounts to half of that of C_5, because C_4 lies exactly one octave lower. See also T08a for a 12-octave list of frequencies with equal spacings of 100 Ct.

The reverse equation: $c = 1200(\log(q_f)/\log(2))$

An example: What is the size of the interval between 500 Hz and 600 Hz?
The answer:
$c = 1200(\ln(600/500)/\ln(2)) = 1200(0.1823/0.6932) = 1200 \times 0.2630 = 315.6$ Ct.
Through this it can be seen that the frequency ratio 5:6 corresponds as a rule to the interval 315.6 Ct (somewhat larger than the tempered minor third between A_4 and C_5).

Another example: How big is the interval between A_4 (440 Hz) and the frequency of the rotation of the earth (once in 23.9345 hours, i.e. 0.000011605 Hz)?
The answer:
$c = 1200(\log(440/0.000011605)/\log(2)) = 1200(\ln(37912204)/\ln(2)) = 1200(17.45/0.693)$
$= 1200 \times 25.18$ Ct or 25.18 octaves, i.e. 25 octaves and 2.11 semitones.
This means that the earth rotates with a frequency 11 Ct below the tone G_{-21}.

Concerning loudness: A barely audible tone of 1000 Hz exerts a *sound pressure* (a measure of the loudness) on the ear of approximately 20 *micropascal* (abbreviated 'µPa'), which equals 0.2 *nanobar* ('nbar'). If this pressure is repeatedly multiplied by a constant factor, the loudness appears to increase evenly. With a factor of e.g. 10, the corresponding loudness interval is given as 20 *decibels*.

A *Bel*, named after the inventor Alexander Graham Bell (1847-1922) corresponds by definition to ten times the *sound intensity* (another measure of loudness, in power e.g. Watt or ‘W’ for short, per area e.g. square metre or ‘m^2’ for short); hence, 2 Bels corresponds to 100 times the sound intensity. The sound intensity varies proportionally to the square of the sound pressure (see T08b and Γ09; the latter graphically displays this relationship) – a hundredfold sound intensity increase means a tenfold increase in sound pressure: 1 Bel = 10 decibels, 2 Bels = 20 decibels, corresponding to hundredfold intensity, tenfold pressure. In combining several (non-sinusoidal) sources of sound, their sound intensity is added; however, in some aspects (e.g. in sound wave amplitudes or electrical voltage) sound pressure provides a more suitable unit of measure. When adding sine tones, their phase plays an important role – e.g. two sine tones of opposite phase cancel each other out (see Chapter 29).

An increase in the sound intensity by 10 decibels (‘dB’ for short) results therefore in a tenfold increase in the sound intensity; hence, an interval of l dB corresponds to a sound intensity quotient q_i as follows:

$$q_i = i_2/i_1 = 10^{l/10},$$

where l is the difference in sound intensity in dB and i_1 and i_2 are the sound intensities of the sounds comprising the interval (e.g. in W/m^2).

As shown above, a 20 dB increase corresponds to a tenfold increase in the sound pressure; correspondingly, the following holds for a sound pressure-quotient q_p:

$$q_p = p_2/p_1 = 10^{l/20},$$

where l is the same as above, p_1 and p_2 represent sound pressure (e.g. in μPa).

An example: Find the sound intensity of a tone 120 dB above the level 1 pW/m^2
$i_2 = i_1.10^{l/10} = 1$ pW/m^2 $\times 10^{120/10} = 1$ pW/m^2 $\times 10^{12} = 10^{12}$ pW/m^2 = 1 W/m^2

Another example: Find the sound pressure of a tone 120 dB above the level 20 μPa.
$p_2 = p_1.10^{l/20} = 20$ μPa $\times 10^{120/20} = 20$μPa $\times 10^6 = 20$ Pascal (‘Pa’ for short).
As seen in T08b, the sound pressure 20 Pa corresponds to the sound intensity 1 W/m^2.

The converse equations are as follows: $l = 20(\log(q_p)/\log(10)) = 10(\log(q_i)/\log(10))$,
simplified through a base-10 logarithm to $l = 20 \lg(q_p) = 10 \lg(q_i)$.

An example: How many dB correspond to a doubling of the sound intensity?
$l = 10 \lg(2) = 10 \times 0.30103 = 3.0103$ dB.

Another example: How many dB correspond to a doubling of the sound pressure?
$l = 20 \lg(2) = 20 \times 0.30103 = 6.0206$ dB, meaning four times the sound intensity.

It follows from the above that sound pressure and intensity (analogous to frequency) are linear units, while decibels (analogous to cents) are logarithmic units.

If the frequency of a given pitch is doubled, we notice that the pitch rises by exactly one octave. This happens with any arbitrary initial frequency, so that we could say: the octave is a pitch interval corresponding to a frequency-ratio of exactly 1:2. In the case of the ratio 2:3 (e.g. 100 Hz:150 Hz, the 2nd and 3rd partials of an overtone series based on 50 Hz), we are dealing with a pure fifth; because 440 Hz is A_4, the frequency of E_5 is a fifth higher, 660 Hz. As a rule we can say that the most essential characteristic of a pitch interval is the ratio between its frequencies, representable as a pair of mutually prime whole numbers in the form P:Q.

Our musical experience tells us that a doubled fifth is a major ninth. Take the example C_4+fifth=G_4; G_4+fifth=D_5. If f Hz is the frequency of C_4, then the frequency of G_4 is $f\times 1.5=1.5f$ Hz. Since an increase by a fifth always causes a rise in frequency by 50%, the frequency of D_5 is $1.5f\times 1.5=2.25f$. The fact that halving this value ($2.25f/2=1.125f$) brings us down an octave to D_4, has to do with a falling octave halving the frequency. This process can also be described as follows:

	C_4	+fifth	+fifth	−octave	$= D_4$
or	f	$\times 3/2$	$\times 3/2$	$\times 1/2$	$= (9/8)f$.

This indicates that an <u>increase</u> in pitch by a given interval results from the <u>multiplication</u> of the frequency by the common fraction (quotient) corresponding to the frequency ratio of the interval. Conversely, a <u>decrease</u> by a certain interval corresponds to the <u>division</u> of the frequency by the corresponding fraction. A move from a pitch to another of the same frequency implies the unison: 1/1. Here then, are the fractions of the four intervals shown so far that extend to and include the octave; they are the unison (1/1), the major second (9/8), the fifth (3/2), and the octave (2/1).

Further scale degrees can be calculated similarly,
as e.g. the major sixth = major second plus fifth, hence the fraction = 9/8 × 3/2 = 27/16
or the perfect fourth = octave minus a fifth, hence the fraction = 2/1 × 2/3* = 4/3
or the major third = major sixth minus a fourth; fraction = 27/16 × 3/4* = 81/64
or the minor seventh = fourth plus fourth; fraction = 4/3 × 4/3 = 16/9

(*division by a fraction is the same as multiplication by the reciprocal of the fraction)

The above intervals are tuned exclusively by the addition or subtraction of octaves and fifths; this method of tuning is called *Pythagorean* after its propagator Pythagoras (ca.569-ca.475 BCE). The major third 81/64 found in this manner is also called the *ditone*, because it results from adding two 9/8-whole tones (9/8 × 9/8 = 81/64). Also, the 16/9 seventh is the same as an octave minus a 9/8 whole tone (2/1 × 8/9 = 16/9).

That other tunings are possible can be seen by way of example of the same 81/64 major third, which also can be represented by 64:81 (the ratio P:Q – whereby the arbitrary convention P<Q holds in this book – corresponds to the fraction Q/P if the interval increases, P/Q if it decreases). In the overtone series, where each partial is a multiple of the fundamental frequency, the interval between the 1st and 2nd partials (1:2) is, as to be expected, an octave, the interval between the 2nd and 3rd partials (2:3) is a fifth, that between the 3rd and 4th partials (3:4), a fourth (see Γ06a). Between the 4th and 5th partials, an interval can be found that is at 386 Ct smaller than the tempered major third (400 Ct); it sounds in context like an especially pure major third, and is called the *pure* or *natural third* because of its occurrence in the overtone series. Compare its size to that of its Pythagorean namesake:

with 5/4 the interval amounts to $1200 \times (\log(5/4) / \log(2)) = 386.31$ Ct,
with 81/64 it amounts to $1200 \times (\log(81/64) / \log(2)) = 407.82$ Ct.

The 21.51 Ct difference, about a fifth of a semitone, called the *syntonic comma*, corresponds to the interval $(81/64) \times (4/5) = 81/80$.

The harmonic series can be made audible on a tense oscillating string (for example on one of a piano or a stringed instrument) through touching nodes at special places; for the n^{th} partial, the distance between the corresponding node and any of the two ends of the string is $1/n^{th}$ of the length of the string. If e.g. the 5th partial of C_2 on the piano is sounded in this way, it can be ascertained that the note playable with the E_4-key actually lies only 14 Ct higher (assuming that the piano is in tune). This can also be calculated as follows: C_2 has a frequency of 65.4064 Hz (see T08a); thus its 5th partial has the fivefold frequency 327.032 Hz. However, the E_4 exactly 28 tempered semitones above C_2 is higher according to T08a: at 329.628 Hz. The sounding of both tones together results in a distinct ‘beating’ oscillation in loudness of about 2.6 Hz, corresponding to the difference in frequencies, explainable by acoustical principles – see Chapter 29.

Naturally, intervals can also be tuned through addition or subtraction of octaves, fifths and (pure) thirds, as e.g.:

the major sixth = pure third plus a fourth; the fraction $= 5/4 \times 4/3 = 5/3$ or
the minor seventh = two fifths minus a pure third; $3/2 \times 3/2 \times 4/5 = 9/5$.

Here, once again, are the Pythagorean derivations (measured from C – octave, fifth and third are indicated here by Ω, Q, T) of the tones

	D	E	F	G	A	B♭
	2Q−Ω	4Q−2Ω	Ω−Q	Q	3Q−Ω	2Ω−2Q
An alternative:		T			Ω+T−Q	2Q−T

A comparison of the equation for the conversion of pitch interval size (in cents) into frequency ratio with the equation for the conversion of loudness difference (in decibels) into intensity ratio shows interesting parallels, for example, if the frequency quotient of a tempered major third (400 Ct) is calculated:

$$q_f = 2^{400/1200} = 2^{1/3} = 1.25992105$$

It can thus be said: a frequency increase of 25.992% corresponds to an increase in pitch by a tempered major third.

Now calculate the intensity quotient of one decibel:

$$q_i = 10^{1/10} = 1.258925412$$

It can thus be said: a sound intensity increase of 25.893% corresponds to an increase in loudness by one decibel.

The similarity of both of these values can be used, for example, as follows: if one wishes to know how many decibels correspond to a doubling of the sound intensity, then one needs only to consider how many tempered major thirds are necessary for a doubling of the frequency, or in other words, how many major thirds fit into an octave – namely 3. Or if one wishes to know, for example, what intensity ratio corresponds to 10 dB, i.e. 1 Bel, one needs only to consider which frequency ratio is formed by 10 tempered major thirds (= 3 octaves plus a third): $2 \times 2 \times 2 \times 5/4 = 10$ as by the definition of the Bel (the natural third 5/4 is close to the tempered in size) – see Γ10a for an illustrated comparison on the basis of a keyboard of units of pitches and sound intensity (in this case intensity-ratios; for sound pressure, double the dB-value shown).

The fact that cents and decibels are logarithmic representations of measurement quotients (which are independent of measurement units), permits application in other areas, like, for example, in banking: were we to assume that a sum of money earns interest at an annual rate of 7.2%, we could rightly maintain that this sum will grow continually by $1200(\log(1.072) / \log(2)) = 120$ Ct or by $10\lg(1.072) = 0.3$ dB per year. In order to be doubled, the sum would need 1200 Ct/120 Ct or 3 dB/0.3 dB = 10 years. This exercise is surely amusing, but the common areas of application of cents and decibels should not hide the fact that they are logarithmically calculated units for measuring ratios.

In the same way that a number of metres describes the spatial distance between two points, measurements in cents and decibels are used to illustrate the distance between two pitches or loudnesses. Elevation is commonly given in metres above sea level; here, the elevation of sea level is defined as ‘zero metres’. In the field of temperature, the zero-point of the Celsius scale has been fixed at the melting point of ice. There are two dB-scales that are fixed in this way. One – called dB(SPL) from Sound Pressure Level – has its zero-point at 20 μPa (= 1 pW/m^2), which is, according to an earlier general assumption, the weakest possible audible sound pressure at 1000 Hz – this scale is used in acoustical measurements; T11 offers a series of commonplace dB(SPL)-examples. The other scale – called dBu – is utilised in recording studios: here, the zero-point is arrived at when the voltage of the circuit of flowing alternating current nearly saturates a magnetic tape – 0 dBu is standardised at 0.775 Volt, the common unit of voltage (‘V’ for short), named after the physicist Alessandro Volta (1745-1827). Starting at this maximum value of zero and falling, dBu-values are mostly negative; compare, for example, the calibration of a so-called ‘*Volume Unit* (VU-) *Meter’* in Γ10b. The conversion from voltage-ratio in dB is carried out according to the formula for sound pressure – voltage in Volts (V) behaves just like sound pressure (Pa) and amplitude (mm).

As can be gathered from T08b, a Pascal is by definition the pressure of 1 Newton of force (corresponding to a weight of about 102 grammes at sea level) applied to 1 square metre; accordingly, 20 μPa correspond to a weight dispersion of 2.04 milligrammes per square metre or about 20 grammes per hectare (thus 1 Pascal = approx. 1 tonne/hectare); the unit was named after the mathematician and philosopher Blaise Pascal (1623-1662). Another pressure unit is ‘bar’ (via ‘barometer’, from the Greek *báros* = ‘weight’, in this case of air): 1 bar = 100,000 Pascal or 1,000 Hectopascal (abbreviated ‘hPa’). The sound pressure of a sound wave is the converted measure (in RMS = ‘root mean square’ – see Chapter 12) of the deviation in air pressure caused by the wave; since this is normally described as 1 bar, one can imagine how a rapid change in air pressure of only ±0.1% (approximately 71 Pa RMS-sound pressure for a sine wave) can lead to a deafening 131 dB(SPL).

Γ10c shows air pressure fluctuations in the area of Cologne, Germany, during the years 2000 and 2001; these curves essentially represent a low-frequency sound wave, despite the calibration of the time-axis in months and the pressure-axis in Hectopascal. Here, the average RMS-loudness is 9.65 hPa or 153.67 dB(SPL)! In addition, the loudest frequency components are to be found in the single-digit Microhertz region, as visible in Γ10d. From this it can be gathered that if this period of time of 2 years were traversed in 2 seconds by means of a time-machine (like that portrayed by the author H.G.Wells), the resulting sound of this wave – now raised to the audible range – would with its 153.67 dB(SPL) cause serious damage to the unprotected ear. Back to Γ10d: observe the narrow sound band at 23 μHz (12 hour period), probably due to the tides, and another at 11.6 μHz (24-hour period), probably due to diurnal temperature fluctuations.

It is known that some animals can hear frequencies too high, others too low for the human ear to hear: the lower and upper limits of human hearing are commonly fixed at 20 Hz and 20 kHz respectively – this is a range of 1:1000 = approximately 10 octaves. However, it is not as if human hearing suddenly starts at 20 Hz and then stops just as suddenly ten octaves higher; audibility enters gradually at both ends of the range of hearing, and the degree of hearing also varies within this range. A sine wave with gliding frequency and constant sound intensity appears much louder, for example, at 1000 Hz than at 100 Hz or at 10000 Hz. If inversely the intensity of a sound is changed (for example through a loudness regulator), so that all frequencies make an equally loud impression, this results in fluctuations in the intensity level, as first published in 1933 by Harvey Fletcher (1884-1981) and Wilden A. Munson (1902-1982); these results were improved in 1956 by D.W. Robinson and R.S. Dadson and registered with the International Organisation for Standardisation (ISO 226), seen here as curves in Γ12a – one sees the dB(SPL) values of sine waves perceptually seeming equally loud at all frequencies, of which the intensity at 1000 Hz has been set at multiples of 10 dB(SPL). Additionally, algebraically calculated and upwards extrapolated curves (grey) have been inserted, which are a fair approximation below 130 dB(SPL) but are pure fantasy above.

Compare the data for six pitches (1½ octaves apart): 44, 125, 354, 1000, 2828 and 8000 Hz. In order to seem equally loud, the dB(SPL)-values have to move from e.g. 79 (44 Hz) to 63 (125 Hz), 56 (354 Hz), 60 (1 kHz), 54 (2.8 kHz) and 67 (8 kHz). These fluctuations are even more extreme with sounds at lower levels, e.g. at dB(SPL)-values 59 (44 Hz), 39 (125 Hz), 29 (354 Hz), 30 (1 kHz), 24 (2.8 kHz) and 39 (8 kHz). The audibility threshold, too, the loudness at which a sine tone is just no longer audible, shows the course 43, 20, 9, 3, −2½ (!) and 16 dB(SPL) at the six given fixed frequencies. The negative dB(SPL)-value at 2.8 kHz corresponds to a sound pressure of roughly 15 µPa, even lower than the dB(SPL)-zero-level of 20 µPa; the amplification in this frequency range is caused by a slight resonance of the outer auditory canal between the eardrum and the outer ear (see Chapter 30). The perceived loudness according to pitch and intensity is described by a level (*loudness level*) expressed in *Phon* (pronounced 'fon', from the Greek *phoné* = 'a sound', here abbreviated 'Ph'), i.e. the loudness level of e.g. 60 Ph corresponds to the intensity levels shown in the first example, varying along the so-called *Isophon* curve (from the Greek *ísos* = 'equal', and *phoné* as above). By definition, dB(SPL)- and Phon-values are identical at 1 kHz. Loudness levels can be read off the curves for every frequency (x-axis; in Hz) and every intensity level (y-axis; in dB(SPL)); e.g. a 500 Hz tone at 50 dB(SPL) seems 54 Ph loud, 4 Ph louder than at 1000 Hz. Interestingly, the audibility threshold is at 3 Ph (about 28 µPa at 1 kHz). In a way, isophons can be regarded as curves of 'relative deafness' – the higher they are, the worse one's auditory perception.

The expression 'loudness level' (unit: Ph) is analogous to 'intensity level' (unit: dB(SPL)). 'Level' – like 'interval' – always refers to a logarithmic scale.

A doubling of the intensity raises its level by approximately 3 dB, an observation derived from the definition of the decibel. At 1 kHz this would correspond to an increase in the loudness level by 3 Ph. It has been found that this increase does not make the tangible impression of a doubling of the subjective loudness: measurements show that 10 Ph are fundamentally necessary for this effect. Thus if the sound intensity of a 1 kHz tone is multiplied by 1024, the subjective loudness increases only 8 times (ι = intensity, ι = subjective loudness):

$$1024 \times \iota = 10 \times \iota\text{-doubling} = \iota\text{-level} + (10 \times 3\ \text{dB}) = \iota\text{-level} + 30\ \text{dB}$$
$$\rightarrow l\text{-level} + 30\ \text{Ph} = l\text{-level} + (3 \times 10\ \text{Ph}) = 3 \times l\text{-doubling} = 8 \times l\text{-impression}.$$

The linear behaviour of the subjective loudness deserves an equivalent linear unit, which exists in the form of the *Sone* (from the Latin *sonus* = 'a sound', here abbreviated 'Sn'). By definition, 1 Sn corresponds to 40 Ph. This gives rise to another equation: since the increase in loudness by 10 Ph causes a twofold rise in loudness, the loudness interval of $\varphi_2 - \varphi_1$ Ph corresponds to a loudness quotient q_l of

$$q_l = s_2/s_1 = 2^{(\varphi_2 - \varphi_1)/10}$$

where φ_1 and φ_2 are the loudness levels in Phon, s_1 and s_2 are the subjective loudnesses in Sone of the sounds making up the interval. By definition, 1 Sn = 40 Ph, therefore the formula can be simplified to

$$s = 2^{(\varphi-40)/10},$$

where φ is the loudness level in Phon and s the linear loudness in Sone.

The reverse formula: $\varphi = 10(4+(\log(s)/\log(2)))$

Examples:
How many Sones are 3 Phons?: $s = 2^{(3-40)/10} = {}^1/_{13}$ Sn (the audibility threshold)

How many Sones are 100 Phons?: $s = 2^{(100-40)/10} = 2^6 = 64$ Sn

How many Phons are 100 Sones?: $\varphi = 10(4+(\ln(100)/\ln(2))) = 10\,(4+(4.605/0.6931)]) =$ 106.4 Ph

It is generally assumed that the dynamic levels ***ppp***, ***pp***, ***p***, ***mf***, ***f***, ***ff*** and ***fff*** match the subjective loudnesses 1, 2, 4, 8, 16, 32 and 64 Sn (40, 50, 60, 70, 80, 90, 100 Ph). For the summation of subjective loudnesses, Sone values are added (with some reservations), e.g. two 60 dB(SPL)-noisebands at 44 and 1000 Hz, which give loudnesses of 30 and 60 Ph, the corresponding linear values of which (0.5 Sn and 4 Sn) add up to 4.5 Sn.

Γ12b shows a summary of all linear and logarithmic terms up to this point, attempting to visually capture the curved symmetry of the two basic functions $y=e^x$ and $y=\ln(x)$.

Observe Г13a: each horizontal row represents an open circle of fifths, each vertical column an endless series of natural thirds. The note C (in the central bold-face frame) is the orientating core of this network, although any other note could do as well. In this network one can identify Pythagorean intervals as well as those based on natural thirds, as in the Pythagorean C-Major scale F–C–G–D–A–E–B (boxes filled with light grey) or the one based on natural thirds, in oversimplification sometimes called *Aristoxenian* after one of its earliest propagators Aristoxenus (350-300 BCE): A–E–B/F–C–G–D (dark grey enclosed area). By the way, it is evident from the above that the Pythagorean system, based on octaves and fifths, sets the maximum involved prime number at 3, and the so-called Aristoxenian puts this value at the next prime number 5. These two systems are therefore also termed the *3-limit* and *5-limit* note-systems respectively.

At this point the justifiable question arises: which of these two alternative tunings is really 'C-Major'? It should be clear that the two tunings and therefore their frequencies differ: with C common to both, the notes A, E and B would be, using the 5-limit tuning, a syntonic comma (21.5 Ct) lower than in the 3-limit tuning. Furthermore, what does all this have to do with the widespread *equal temperament* of musical instruments (where all intervals are supposed to be multiples of one single basic interval – the 100 Ct-semitone)?

Take an example: we hear march music on the street, and we try to avoid stepping to the beat of the music while passing by. This is difficult, because, although the walking rhythm would really be the same as the music tempo for only small fractions of a second (no one could walk more precisely), our sense of timing compulsively 'bends' the perception of our steps to make everything comprehensible to the brain. Another example: we hear a small child singing, out of tune, a song unfamiliar to us but recognisably in the major scale: we achieve this recognition through the ability to adjust false notes while listening. Afterwards we would be able to play the corrected melody on the piano, probably with the approval of the child. Nevertheless, the temperament of even a freshly tuned piano seems to us to be quite out of tune immediately after the extensive enjoyment of mediæval music, because our sense of pitch has been sensitised and made more demanding.

Depending on our momentary musical sensitivity, we are therefore more or less able to bend pitches consciously or unconsciously to an imagined position where they make more musical sense. This faculty allows tempered-tuned music to appear in our imagination in various forms of tuning, 3-limit, 5-limit or any other so-called 'pure' tunings – i.e. representable through whole-number frequency ratios. In other words: the music's tuning is rationalised by the brain (assuming the composer has not consciously tried to withhold the music from this process – Schoenberg took pains to compose 'truly atonally') and each irrational frequency relationship is transformed subconsciously into a quantitatively nearby rational one. Harmony is the study of that which is intervallically <u>intended</u> or at least <u>understood</u>.

For purposes of further explanation, I draw upon three linguistic concepts: Phonetics, which describes the actual sounds, Semantics, which attempts (if need be through adjusted hearing) to make sense of the sounds, and Grammar, which through a set of rules can serve both the understanding of the sounds heard as well as their production for the purpose of comprehension. A pitch system, the grammar of which is effectively employed, can be grasped semantically, even if slight 'phonetic' deviations from the expected norm are present (similar to accent in language). An example: Bach also makes sense on instruments with modern tuning. In short: Grammar assists Semantics in comprehending Phonetics. Conversely, an effective semantic interpretation allows the postulation of a plausible grammar from something 'phonetically' perceived. A prerequisite for dictation is comprehension; the widespread distribution of tonal grammars contributes to the making of a tonal dictation easier than an atonal, a 12-tone dictation easier than a microtonal one.

There are numerous scales which can be described and tuned in terms of 3-limit, 5-limit or further considerations. Γ13b shows two more 5-limit pitch sets, of which one is the centuries-old classical European chromatic scale: the twelve tones make up a clean matrix of three rows of connected fifths by four columns of connected thirds (light grey-filled boxes). Based on this, the frequency ratios, reduced to an octave range, are in ascending order as follows:

1:1 15:16 8:9 5:6 4:5 3:4 32:45 2:3 5:8 3:5 5:9 8:15 1:2

Even the classical North Indian system of *srutis*, 22 intervals, described in music theory as a comprehensive pitch set for over two thousand years, can be found in this network of fifths and thirds (against a dark grey background). The regrettably widespread assumption that this set comprises 22 equal-tempered intervals (each thus $^1/_{22}$ of an octave = 54.5 Ct), is a myth. It is has also been convincingly proven that other intervals are made use of in North Indian practice besides these 22.

In Γ13a and b one can see many notes with the same name; the notes C, E, F, G, A and B appear three times, and the notes D and B♭ as many as four times. A comparison shows that every note is connected to its nearest namesake to the right by four upward fifths and a downward third (or vice versa), i.e. $3/2 \times 3/2 \times 3/2 \times 3/2 \times 4/5 = 81/80$, the syntonic comma of 21.5 Ct. Also, *enharmonic equivalences*, familiar to music theory, like the *minor diesis* 125:128 (Ω−3T: 41.1 Ct), the *diaschisma* 2025:2048 (3Ω−4Q−2T: 19.6 Ct) and the 524288:531441(!) *Pythagorean Comma* (12Q−7Ω – see the middle row from G♭ to F♯: 23.5 Ct) are in the network. The minor diesis could for instance appear in music where C♯ leads to D near D♭ leading to C – an example of this and of the diaschisma can be seen in Γ13c$_1$ & $_2$; in just-intoned music the notes C♯ and D♭, though enharmonically 'equivalent', can sound as in these examples 19.6 Ct or even 41.6 Ct apart.

Look at the triangle ABC in Γ14a: there is a right angle at apex C. The side AB opposite this angle is called the *hypotenuse* (from the Greek *hupó*=‘under’ and *teínein*=‘stretched’). With each of the other two angles (at the points A and B) as vantage points, there are two other sides besides the hypotenuse, the opposite side and the adjacent side – opposite the point A, for example, lies the opposite side BC, while AC ist the adjacent side.

The quotient opposite/hypotenuse is called the *sine* of the angle (Latin *sinus* = ‘fold’, abbreviated sin) at any apex, i.e. sin(A)=BC/AB, sin(B)=AC/AB (the angle at apex ‘X’ is also called ‘X’). The quotient adjacent/hypotenuse is the *cosine* of the angle (Latin *co-* = ‘complementing’, abbreviated cos), with cos(A) = AC/AB, cos(B) = BC/AB (the absolute values of sine and cosine stay between 0 and 1, because the opposite and adjacent sides can never be longer than the hypotenuse). The remaining quotient opposite/adjacent is the *tangent* (abbreviated tan): tan(A) = BC/AC, tan(B) = AC/BC. In fact the gradient at an arbitrary point (x,y) of the straight line $y=mx$ passing through the origin is the same as the quotient y/x and is thus also the tangent of the angle between the straight line and the x-axis – cf. Γ03d.

Triangle calculation or *trigonometry* (Greek *tri* = ‘three’, *gonos* = ‘angled’, *metron* = ‘measure’) shows that the sine, cosine and tangent of a given angle always remain constant. For example again in Γ14a: if the lengths of the triangle’s three sides AB, BC, CA have the ratio of e.g. 5:4:3, then the actual size of the triangle is not important for the angular content – if the ratio 5:4:3 remains constant, all of the proportions and therefore all three angles also remain constant. In this example, sin(A) = BC/AB = 4/5 = 0.8. From this, angle A can be calculated, the *arc sine* (abbreviated arcsin) of 0.8 – as can be shown by a pocket calculator, A = arcsin(0.8) = 53.13°. Also, B = arcsin(AC/AB) = arcsin(0.6) = 36.87°. It is not surprising that A+B = 90° (remembering that C = 90°), because the sum of all three angles of a plane triangle is always 180° (=A+B+C).

If a non-right angle and the length of one side of a right-angled triangle is known, one can determine all of the other properties of this triangle. Look at Γ14b – a circle encompasses the three right-angled triangles OMN, OPQ and ORS; the hypotenuses of all three represent the radius of the circle and are thus equal. If the angle ROS made by the apices R, O and S is 60°, we can show: RS = RO×sin(ROS) = $r$sin(60°) = 0.866r, where r is the circle’s radius. If POQ = 45°, then PQ = PO×sin(POQ) = $r$sin(45°) = 0.707r. If MON = 30°, then MN = MO×sin(MON) = $r$sin(30°) = 0.5r. Now sin(30°)<sin(45°)<sin(60°) because MN<PQ<RS.

If we draw a curve in a two-dimensional coordinate system, in which the x-axis displays a constantly increasing angle from left to right and the y-axis shows the corresponding value of $\sin(x)$, the result is the sine wave shown in Γ14c: to the right of the origin we see about 1¼ *period*s of the curve. The three values $\sin(30°)$, $\sin(45°)$, $\sin(60°)$ are drawn in as vertical lines to the right of the origin – they have the same mutual relationship as the lengths of the sides MN, PQ, and RS in Γ14b.

The correlation between circle and sine wave becomes even more evident by observing the cylindrical spiral in Γ14e from different perspectives: at top right (marked by 0°), its central axis first points straight at the viewer and is then horizontally turned by 10°, 30°, 60°, 75° and 90° (from top right to bottom right and then to the left). The first drawing (0°) shows a circle, while the last one (90°) clearly shows a sine wave.

In Γ14c the x-axis is not calibrated in degrees – at 30° (in parentheses) one sees $^{\pi}/_{6}$, at 60° $^{\pi}/_{3}$, at 90° $^{\pi}/_{2}$ and at 180° π. The quotient circumference/diameter of any circle equals the constant π, with a value of 3.14159265358979323846264 33.... The x-axis in Γ14c is calibrated in *radians* from 0 to 8 (Latin *radius* = 'measuring rod'), an angular unit found in mathematics, where 1 radian = $180/\pi$ degrees = 57.29577951°, the angle between two radii of a circle which cut off (subtend) a one-radius-long segment of the circle on its circumference. Γ14d compares this definition with an *equilateral triangle* with sides the length of the radius and with 60° angles. Since π is the quotient circumference/diameter, a complete run around the circle (360°) covers a 2π-fold radius-length; thus the movement of one single radius length along the circumference corresponds to an angle of $360°/2\pi$, i.e. 1 radian.

It is not necessary to always remember the origins of trigonometric functions in triangular measurements; the Cartesian representation is normally sufficient.

Here are some of the most important trigonometric rules:

$\sin(x) / \cos(x) = \tan(x)$

$\sin^2(x) + \cos^2(x) = 1$

$\sin(^{\pi}/_{2}-x) = \cos(x)$ (...these three equations are to be seen

$\sin(0) = 0; \sin(^{\pi}/_{2}) = 1$ in the mutually shifted positions of the

$\cos(0) = 1; \cos(^{\pi}/_{2}) = 0$ sine- and cosine-periods shown in Γ14c)

The rules can be easily proved by the triangle in Γ14a, e.g.

$\sin(A) / \cos(A) = (BC/AB) / (AC/AB) = BC/AC = \tan(A)$.

Since $AB^2 = BC^2 + AC^2$ (the law of Pythagoras), the following is also true:

$\sin^2(A) + \cos^2(A) = (BC/AB)^2 + (AC/AB)^2 = (BC^2+AC^2)/AB^2 = AB^2/AB^2 = 1$.

$\sin(90°-A) = \sin(B) = AC/AB = \cos(A)$.

Trigonometric functions can be expressed as series, e.g. (with x in radians):

$\sin(x) = x - x^3/3! + x^5/5! - x^7/7! + ...$, whereby $n!$ ('*factorial* n') $= n(n-1)(n-2)..\times 3\times 2\times 1$.

A sounding object is in a state of rapid vibration passed on to the air (or another medium); this vibration is transferred to the eardrum and arrives in this way in the auditory system. Observe Γ15a: the thick line at the bottom represents at left a solid, immovable object, the thin lines above at left represent immovable air molecules (air molecules really approach immovability only near to absolute zero, a temperature of $-273°$ C, but their usual movements have no relevance to hearing, because they are neither directional nor periodic). If the solid object starts to move back and forth, then the normal distance between it and the nearest air molecule, determined e.g. by the temperature, changes; the molecule 'attempts' to compensate for this by also moving back and forth. As long as the solid object vibrates, the air molecule must also vibrate; quietude is only reached after the solid object stops vibrating. But for the same reasons, the next molecule also starts to move, then the next etc. – a sound wave is formed that moves through the air with a speed of about 320 to 360 metres per second according to temperature and humidity; if the frequency of the vibrations shown in Γ15a is 1000 Hz, then the diagramme represents a time span (from left to right) of about 6 seconds as well as a distance (from bottom to top) of about 330 metres. This representation of the molecular density of air – only one molecule every ten metres – is naturally extremely sparse (there are about 27×10^{21} air molecules per litre under normal circumstances, corresponding to a one-dimensional density of 300 million molecules per metre, averaging a third of a nanometre in size and 50 yoctogrammes or 5×10^{-23}g in mass). The type of movement of the sound wave is called *longitudinal*, because it moves in the plane in which the molecules move; in waves on the surface of a body of water, for example, the particles move on the surface at right angles to the waves' diffusion – this type of movement is called *transversal*; light and other electromagnetic waves belong to this category.

A look at the grey strip in Γ15a placed between 90 and 100 metres shows a constantly varying distance between two neighbouring molecules – the variations are, as the vibrations themselves, sinusoidal. Since the air pressure is inversely proportional to the molecular distance, this too changes sinusoidally; this variation in air pressure leads to the phenomenon of sound pressure, which can be calculated using the formula $p=2\pi f a c d$, where p is the sound pressure, f is the frequency of the sine wave, a is the maximum molecular distance from the central position of rest (usually called 'amplitude' – see top of next page), c is the velocity of sound in the medium transmitting the sound, and d is the density of the medium. In this formula, the *particle velocity* (the velocity of the molecule going through the position of rest) is represented by $2\pi f a$, thus equal to $p/(cd)$. All units of measurement (Pa, Hz etc.) can be reduced to kilogrammes, metres and seconds, as can be seen at the bottom of T08b.

Concerning terminology: ‘amplitude’ is unfortunately very frequently used for ‘molecular displacement from rest’; the term only makes sense in reference to the general envelope of the maxima of a wave. To add to the confusion, ‘amplitude’ is also frequently used in the sense of sound pressure in general.

An example of the calculation of the particle velocity: at a sound speed of 333 metres per second and an air density of 1.3 grammes per liter, it can be seen that the sound pressure of a 1000 Hz sine tone of e.g. 60 dB(SPL) (= 0.02 Newtons/m^2 or kg×m^{-1}×s^{-2}) derives from a particle velocity of p/(cd) or 0.02 kg×m^{-1}×s^{-2}/(333 m.s^{-1}×1.3 kg×m^{-3}) = 0.000046 m/s or about $^1/_{20}$ mm (150000 times the molecular diameter) per second.

A sound wave is commonly represented by a two-dimensional curve of the changing air pressure or of the molecular distance from the mid-position against time, which under normal circumstances is proportional to the pressure. What happens when a moving object is subject to two sinusoidal vibrations of different speeds? These are simply added, as shown in Γ15b (bottom right): two equally strongly fluctuating sinusoidal vibrations with the frequency ratio of 2:3 work together as a curve, which like its sine components is also periodic, if more complicated. The stronger a sound wave spatially fluctuates, i.e. the higher its amplitude, the higher the resulting sound pressure will be; both of these values are proportional.

At top right in Γ15b, the two sine components are also to be seen as two parallel vertical lines – their frequency ratio 2:3 (a perfect fifth) allows them to be represented as the 2nd and 3rd partials of a *harmonic spectrum*, where the word ‘harmonic’ refers to the whole number ratios of overtones and the word ‘spectrum’ comes from the Latin *spectrum* = ‘apparition’ (after Newton’s spectral light experiments). The length of the vertical lines indicates the amplitude, i.e. the sound pressure of the notes – in this case equally strong.

Γ15b also illustrates the calculation of the RMS (*root mean square*), a method used to measure sound pressure, in this case that of the two sine components and their sum as shown in the diagramme. 37 sine values of regularly spaced x-values of the two components were selected (see the grey vertical lines), their squares added and the sum divided by 37. As a result, each of the curves (given a arbitrary maximal value of 100) yielded an average square of 5000, the square root of which – the RMS – is 70.7. The RMS of the additive curve (see the black vertical lines) is 100 and implies thereby an increase in the sound pressure of the sound wave shown here to a 100/70.71 = 1.4142 = $\sqrt{2}$-fold value; this corresponds to the related 2-fold increase of the sound intensity, which is to be expected: the simultaneous sounding of two sound sources results in general in the summation of their intensities. In addition, Γ15b shows at left a hypothetical ‘molecular snapshot’ of the perfect fifth-sound wave at right: a cloud of particles, the centre of which represents the sound source, shows wave maxima as dense rings, wave minima as sparsely populated ones.

One who uses a computer will notice a certain basic structuring of the machine (see Γ16a): next to the *computer* itself there is a *keyboard* for entering data, a *screen* for checking data, and – in order to store derived information – a *storage medium* for mechanical, magnetic or optical storage, and possibly a *printer* for documentation on paper. The storage medium itself has taken on numerous forms over the years, e.g. punched cards, punched tape, magnetic tape (from common cassettes to those on reels), diskettes in sizes 3-, 5½- or 8-inch, readable and writeable plastic discs using laser-beams or removable or fixed hard-disks (also but less frequently spelt '-disc').

All of the above-mentioned equipment can be summarised in the single term *hardware*. However, in order that the components function and communicate with each other, so-called *software* must be available, coded information that – as soon as it is entered – resides in the electronic circuitry (*chip*) reserved for it. Additional software can be purchased: these are complete *programs** that, for example, make comfortable word-processing or video-games possible, or – especially interesting for musicians – generate sound or draw musical notation. Commercial software is unfortunately most often meant for a specific brand of computer – the adjusting of a program to a computer that is foreign to it is cumbersome.

(*spelt thus in British English only for computer software; otherwise 'programme')

The user of programs deals with the computer at its most intrinsic level (see Γ16b). However, a user wanting to write a program has to go a step deeper, as it were, and to formulate the problem in a *programming language* (like e.g. C, Pascal, Lisp, Basic, Assembler, Perl, Java, etc.), then input the program into the computer through an adequate *editor* (from the Latin *ex+dare*= 'out-give'), in order then, through a *compiler* (from the Latin *com+pila*= 'together+pile up', or even '+plunder', in an earlier meaning) to translate it into *machine language* – the computer's 'own language'. Most often an editor is supplied as an integrated part of a compiler; since mistakes in programs are inevitable, this allows a comfortable frequent changing back and forth between editor and compiler. In the case of languages like Basic, each line of the program-text is usually translated and executed directly after it has been read by an *interpreter* (a 'simulataneous translator', from the Latin *interpretari* = 'explain' or 'translate'). The advantage of this is that one can program much more directly; the disadvantage: the interpreted program runs much slower and less economically than a completely compiled one.

Editor, compiler and interpreter are themselves programs that must be installed. In order that they run, they have to be compatible with the *operating system* installed in the computer, a program that works a step even deeper (e.g. CP/M, MS-DOS, RT-11, OSX, UNIX, Linux, Windows etc.). The operating system makes normal interaction with the computer possible – data can be copied from one medium to another or it can be printed; even more important, runnable programs can be started and – if necessary – interrupted, and if the user wants to input data into the running program, the operating system accepts this data and passes it on into the program. For some computers there have been of late more than one operating system available; as may be required, one can then switch from one to another (which can be cumbersome).

The computer can understand the operating system only if the manufacturer has written appropriate fundamental software (often called the BIOS or *basic input/output system*) for the *processor* (or CPU, for 'central processing unit'), the 'brain' of the computer (well-known processors are the Motorola 68000 series, the Intel 8086 series etc.). This is the lowest level of the sofware hierarchy.

With this we have come back to hardware: next to the processor, the virtual storage RAM (*random access memory*) is of great importance; the processor can arrive at any corner of this storage area at high speed to retrieve or store information there. By comparison, the reading of disk storage, even though still practically in the microsecond realm, takes quite a while longer (the *access time*); besides, in this case the computer often has to go through an entire bank of data (called a *file*) until it finds what it is looking for.

To summarise: if a computer is to be ready for use, the operating system must be installed in the RAM; only then can programs run. If a program is to be written and tested, the user writes the program text in the editor, then has the text translated into machine language by a compiler or an interpreter – in this form, the program can be started by the operating system. Most computer brands load the operating system automatically when switched on.

If we allegorically call the processor the brain of the system, the RAM would be the memory, the monitor facial expression, and the drive would be the briefcase. In addition, another kind of storage should be mentioned here, called the ROM (*read only memory*), a kind of brainstem, which contains previously prepared and entered data of fundamental importance to the programs and which thus can only be read, not written to; it can assume various forms, ranging from chips to compact discs.

Sound can be recorded by means of the magnetisation of iron oxide (tape) or through incisions in plastic (grammophone record), which varies according to the form of the sound wave – this is the so-called analogue method; alternatively, the sound wave can be represented as a series of numbers, which can then be later transferred back into electrical fluctuations and through this into sound waves – this is the digital solution. An analogue recording has certain disadvantages compared to a digital one – the magnetisation of a tape can wear off little by little with time or become imprinted as pre- or post-echo on neighbouring windings of the tape; more essential: one is forced to listen to the background noise that arises during recording or the crackling caused by dust in the grooves of a record. In contrast, in a digital recording, it is only numbers given by scanning the sound waves that are stored on numerous possible media; a possible depreciation of these numbers cannot normally occur – either they are erased, seriously falsified, or they remain the same. If some of these numbers are seriously falsified, the results are felt more as a complete loss rather than as a deterioration of the quality. The main source of disturbance in a digital recording is in the resolution of the scanning of the curve – if this is not high enough, the distortion of the sound curve is audible.

Γ17a shows at left a sine wave period in a scan with a resolution of 8×8, i.e. at eight equidistant positions on the x-axis a y-value is determined, rounded up or down to one of the eight numbers from 0 to 7. This approximation, seen here as a thick line, is obviously a great deal coarser then, for example, that made with an equally arbitrary resolution of 21×21, seen in Γ17a in the middle. At 50×50 (Γ17a at right) the curve looks even better. Not until a resolution of well over 500×500 is reached can the digitised wave acoustically acceptably resemble the original analogue wave. Commercially available equipment for converting waves back and forth between analogue and digital contain *analogue-digital* or *digital-analogue-converters* (‘AD’- or ‘DA’-converter for short) – they are currently and commonly capable of a scan-rate of 44100, sometimes 48000 Hz and a numeric value range from −32768 to +32767; e.g. in one period of a 20 Hz tone, this would correspond (at a rate of 44100 Hz) to a resolution of 2205×65536, at 20000 Hz about only 2×65536 – the latter frequency is so high, that overtones stemming from inaccuracies due to the lower resolution are inaudible. Each of the numbers scanned from a sound wave in this manner is commonly called a *sample* (in the sense of ‘specimen’), and the process is called *sampling*, although the word is often also unfortunately and misleadingly used for a short sound example actually containing many samples. The sound wave scan-rate mentioned above is called the *sample rate*.

The physicist Harry Nyquist (1889-1976) showed that for meaningful sampling there is a limit in the relationship between sound wave frequency and sample rate, viz. 1:2; this means that for a sound frequency ƒ, the sample rate should be above 2ƒ. For a sample rate S of 44100 Hz, the Nyquist limit Ŋ is S/2 = 22050 Hz. Γ17c shows three sine waves, the frequencies of which respectively lie below Ŋ (at Ŋ/6), at the limit and above the limit (at 13Ŋ/6); at a sample rate of 44100 Hz the three frequencies would be 3675, 22050 (=Ŋ) and 47775 Hz, shown by the smooth sinusoidal curves. The leftmost wave has been recorded correctly: the dots graphically representing the samples show an outline visibly true to the original. However, the samples in the middle wave give a false picture: zero-values, silence! The rightmost example at frequency 13Ŋ/6 shows a result which cannot be distinguished from that of the Ŋ/6-frequency wave: frequencies at the limit and higher are shown in the samples incorrectly as being below the limit. This phenomenon is called *aliasing* and can be remedied by previously filtering out all of the frequencies over the Nyquist limit, even for a high-enough sample rate.

Whether seen as a smooth analogue curve or as a digital number series, the recording of a sound wave necessitates a transfer of electrical current from one device to another. For the analogue transfer of a 1-kHz-tone at 0 dBu (studio norm) a curve varying between ±1.096 Volt ('V') with an RMS-Voltage of 0.775V (=$\sqrt{0.6}$V) is formed: alternating current of e.g. 1 Milliwatt at 600 Ohms resistance. If digitised, the sampled numbers are passed on in binary form, where the digit '0' means 0V and the digit '1' about +5V: direct current. In Γ17a (left) the eight sampled numeric values are 4, 6, 7, 5, 2, 0, 1 and 4, expressed binarily as 100, 110, 111, 101, 10, 0, 1 and 100. These numbers here never need more than three binary digits each, sufficient to present the series as 100 110 111 101 010 000 001 100, assuming that in converting them back, the DA-converter knows they are all three-digit binary numbers, this being then a '3-bit quantisation': the numbers are always input and output with a constant quantity of digits (in this case 3), irrespective of whether a zero is at the beginning of any of the numbers or not. A *bit* (from 'binary digit'), the smallest quantity of transferable information, contains the one-digit binary number 0 or 1; the data (here bits) is transferred with a speed called the *Baud rate*, named after the engineer and inventor Émile Baudot (1845-1903): 1 Baud = 1 bit per second in a bit transfer. The *bit resolution* is the number of bits determined for in- and output; the above-mentioned DA/AD-converters use in general a bit resolution of 16. If the eight 3-bit numbers above are transferred with an arbitrary velocity of 2.4 kilobaud, the transfer would take 10 milliseconds, just as long as one period of the frequency 100 Hz. Γ17b compares the analogue electrical transfer of this period (sine wave) with the digital transfer (square wave); in the latter, additionally necessary control bits are not shown. The digital transfer of the scan in Γ17a centre requires 21×5=105 bits (instead of 24 for Γ17a left), because the packaging of the numbers 16 to 21 contained here is impossible with less than 5 bits.

For a computer to serve its purpose it must be programmed; a series of commands must be entered therein, according to which the computer loads data from the hard disk or other medium into its virtual memory (RAM) to process it there and then pass it on to a place determined by the programmer. This data, basically binary numbers, is coded for better readability as regular decimal numbers and letters (together called *alphanumeric characters*) as well as punctuation marks. Because of the way the electronic components are constructed, the standard bit resolution for data transfer is 8; eight bits is called a byte (from 'binary term'), an information capacity allowing the representation of 2^8=256 different numbers (0_d to 255_d) or symbols. Higher numbers are processed by combining several bytes, e.g. the number 12345_d, in binary terms 11000000111001: here two bytes are sufficient, into which the bits are filled as 00110000 00111001 (or hexadecimal 3039_h: these four digits each represent four bits or a half-byte or a *nibble*). The terms bit, nibble and byte do manifest a certain undeniable sense of humour.

The coding of alphanumeric symbols and punctuation marks is usually effected according to the so-called *ASCII-code* ('American Standard Code for Information Interchange'): this represents all one-byte numbers, of which the first (i.e. left or higher) half-byte has one of the six values 2_d to 7_d (= 2_h to 7_h) – the first bit (to the very left, also called the *most significant*) is therefore always 0; this 7-bit-code can be seen in T18. The values 0_h to 1_h as the left half-byte correspond to symbols that one can usually produce on the computer keyboard by pressing the so-called Ctrl-, Control- or Command- keys, depending on the type of computer, simultaneously with one of the other keys (one of the 26 letters a-z plus six additional characters, adding up to 32 in all). Doing this resembles the inputting of upper-case letters by pressing the Shift-key together with a lower-case letter.

Special keys like e.g. Esc, Tab, Backspace, Return also produce these symbols independently – the first two terms are abbreviations for Escape (used frequently to end a program – ASCII No.27) and Tablature (frequently used for table-spacing – ASCII No.9 = Ctrl-I, the 9th letter of the alphabet, also written ^I), while Backspace is self-explanatory (ASCII No.8 = ^H) and Return or Enter (ASCII No.13 = ^M) historically refers to the carriage return lever which returned the typewriter platen (rubber cylinder) to its starting position.

The left half-byte values 8_h to F_h are reserved for additional characters, which can usually be generated only by programs or input by non-English keyboards. Only the programmer who consciously wants to work in a *machine-oriented* fashion, in order to take better advantage of special attributes of the device, would have to input commands as binary numbers; in order to avoid typographical errors through the numerous zeros and ones, the numbers – if (as in most cases) the basic software of the computer permits – may be typed in hexadecimally. A great additional help is a program or programming language called an *assembler*, which allows the input of hexadecimal commands in the form of simple understandable words. On the other hand, the programmer who wants to use the computer in a more comfortable way employs so-called *high-level languages* like Basic, Fortran, Pascal or C (to name just a few), of which the compilers translate the commands written in relatively readable form into binary numbers meaningful to the computer – i.e. into the type-dependent computer-specific *machine language*.

For example, a simple multiplication: two numbers, let us say 25 and 40, are multiplied and the result provided through a so-called *variable*, i.e. temporary storage that can always be reached under a name invented by the programmer. Let us call this variable `product`; in Basic and Fortran the multiplication statement is written `product=25*40`, in Pascal `product:=25*40;`, in C `product=25*40;` (notice the semicolon concluding instructions in Pascal and C). The result can be displayed in Basic and in Fortran by `print product`, in Pascal as `write(product);`, in C as (here simplified) `printf(product);`. The result is then displayed on the monitor: 1000, in Basic immediately, in Fortran, Pascal and C when the compiled program is running. Of course the variable `product` can be avoided by writing `print 25*40` in Basic and Fortran, or `write(25*40);` in Pascal, or `printf(25*40);` in C. Here, however, `product` can be used for other purposes, for example for a comparison – e.g. the result of the multiplication should be shown only if it exceeds 500. The corresponding instruction reads

in Basic: `if product>500 then print product`
in Fortran `if (product.gt.500) then print product`
`(.gt.` means 'greater than' or '>')
in Pascal: `if product>500 then write (product);`
in C: `if (product>500) printf(product);`
(meaning in all cases 'if `product` exceeds 500, then display `product`').

A program can contain a loop in which a command or a block of several commands can be repeatedly executed several times, e.g. in computing the first 10 numbers of the well-known *Fibonacci-series*, in which the first two numbers are both 1 and each number thereafter is the sum of the two previous numbers. This task can be easily solved in the four languages mentioned here (the result is 1, 1, 2, 3, 5, 8, 13, 21, 34, 55) – see L19: the assignments, typically written as `a = 1` (`a := 1` in Pascal), are shown here compactly without spaces. First of all, the value 1 is allocated to the variables `a` and `b`; thereafter the variable `c` repeatedly receives the value of `a+b`, is displayed, and the values of `b` and `c` are moved to `a` and `b`, respectively.

Introduced in 1972 by Dennis Ritchie (*1941) as a advancement of Ken Thompson's language *B*, *C* has become one of the most widespread languages. Its internal proximity to machine language makes highly efficient programming possible. Modularly structured, it permits the insertion of complete and autonomous blocks of instructions into the program text. Despite its role as point of departure for well-known languages like C++, C# and Java, C (thanks in part to the American National Standards Institute) has remained universally standardised through all the years.

A C program basically consists of a *main* part (called `main`) and, if needed, of *functions* written by the programmer. `L20` is a complete C-program for the conversion of number relationships (e.g. 2:3) into cents and decibels based on formulæ given in Chapter 6. It starts with preprocessor directives `#include <stdio.h>` and `#include <math.h>`. C is a very compactly defined language, the core of which does not even contain the instruction `printf`, described before: this is encoded in the includable pre-prepared *library* file `stdio.h`. The preprocessor is a part of the compiler, processing the program text before the compilation proper. `stdio.h` means 'Standard Input/Output', `math` ('mathematics') contains many common standard predefined functions such as logarithms; `.h` stands for 'header', because the `include` directives are at the head of the program.

The words `Conversion of number ratios into cents and decibels`, a commentary inserted as a memory aid, is ignored by the compiler because of the delimiters `/*` and `*/`. Note that in compilable code no non-English characters (such as ä, á, å etc.) are permitted; these are however admissible in commentaries.

`main()` follows. The parentheses, which occasionally contain parameters, show that this module (and others possibly present) are actually functions that can receive and return values (such as sines and logarithms). The value returned is frequently only a measure of success of the execution of the function, the *type* of which must also be declared, in `L20` with `int` for *integer* (whole-number), thus as `int main()`. The `return` to the operating system of a zero at the bottom of the program indicates that everything is in order and that the program will end in an proper fashion.

The contents of `main` are enclosed within curly brackets `{` and `}`; for better readability, they can be written – as in `L20` at the leftmost edge – one above the other. First, variables are declared, `p` and `q` as `int`, then `nLog2`, `nLog10`, `nLog_Quotient`, `ct` and `db` as floating-point values with fractional components following the decimal point (`float`). These variable-names are freely invented and serve to store the numbers `p` and `q` standing in a relationship `p:q`, the natural logarithms of 2, of 10 and of the quotient of the numbers `p` and `q` as well as the cent and decibel values calculated at the end.

At first the standard math function log(...) calculates the natural logarithms of 2 and 10 and stores them in variables nLog2 and nLog10, as in e.g. nLog2 = log(2); (computing time is saved by using variables instead of functions repeatedly). The next line contains a request for the whole numbers p (initialised at the start to 1) and q by printf("Enter P:Q etc. followed by the remark that an invalid entry will stop the program. These two printf functions, written separately for reasons of clarity, contain non-compilable strings of characters in quotes (".."); printf can also contain format strings (e.g. %d) – see further below. The \n at the end (standing for 'new line') causes a move to the next line before anything more is written.

Moving on: the while loop with the condition (p>0) repeats, for as long as p>0 is true, the block of statements attached just below it and set in a pair of curly brackets, right-indented in equal measure. while will run at least once due to p's prior initialisation to 1. From within the block, the program writes P:, gives p the temporary value 0 (why is explained below) and waits for a value to be entered for p through the keyboard by means of the read function scanf("%d", &p); – the format string %d (for 'decimal') indicates that a decimal whole number for p is expected; scanf usually needs & before a variable which is to be read, in this case p.

If p is entered invalidly, e.g. as x, the temporary value set earlier (zero) remains in force; thus the condition if (p > 0) is unfulfilled, causing the block following (on a third level down, set in its own curly bracket pair) to be skipped; in this case the program goes back to the while statement. But the condition (p > 0) is false here as well – the final instruction printf("Program done.\n"); is executed, the program returns a 0 and ends.

If however p is really >0, the 2nd-level instruction block will indeed be executed: the program writes Q:, waits for the keyboard entry of q through scanf, calculates the quotient logarithm log(1.0 * p / q) and stores this value in nLog_Quotient. The reason for the 1.0: if one whole number is divided in C by another, the result is also a whole number: 1/2 does not yield 0.5 but 0, leading to errors. The multiplication (higher priority than division) of p by the float value 1.0 yields a float, which, divided by q also gives a float. Worked out by the said formulæ, ct and db (1200 and 6.021 respectively, for 1:2) are displayed by two printfs: the first, printf("-----The ratio %d:%d corresponds to ",p,q); causes the line -----The ratio 1:2 corresponds to to be written, and the second, printf("%9.3f Ct or %6.3f dB\n Enter P:Q..\n",ct,db); causes the values of ct and db to be written in the format described here below, followed by an new invitation to Enter P:Q... The format string %d:%d in the first printf prescribes the display of p and q as whole numbers (e.g. 1:2). The string %9.3f indicates the writing of ct up to 9 places, with 3 places after the decimal point, thus: ▫1200.000 (▫ is here a space). In the same way, %6.3f causes db to be written up to 6 places, with 3 places after the decimal point, thus: ▫6.021. By the way, the f in printf stands for 'formatted'.

Besides `main` and given functions like `printf`, `scanf`, `log` and `while`, other functions can be defined by the user and called from within the program; they are usually placed before the functions out of which they are called.

L21 is a program that generates table T02. Let us begin with the main part. Following the declaration of `main` and the variable `counter`, the variable `digits[8]` of type `char` (short for 'character') is declared: it is often better to index several variables of the same type under one name as a so-called *array*. `digits[8]` comprises eight characters, beginning with the zero[th], `digits[0]` (in C one starts counting at 0; it is the number of elements which is declared). Type `char` occupies one byte; if one is to work with whole numbers ranging from 0 to 255, these need at most one byte and can be economically stored in a `char` variable, as opposed to the `int` type which usually occupies 4 bytes (32 bits) and can range in value from −2147483648 (= -2^{31}) to +2147483647 (= $+2^{31}-1$).

Variables of type `FILE *` (always in capitals) allow file access. The file name, here `"number_systems.txt"`, followed by `"r"` (for 'read') or `"w"` (for 'write') in the given function `fopen` are allocated to a file variable arbitrarily called `outfile`, thus: `FILE *outfile; outfile = fopen("number_systems.txt","w");`. The variable `outfile` now allows access to the file it represents (see `fprintf` below).

Apart from `while` there is another loop structure termed `for`. But whereas `while` only concerns itself with conditions, `for` initialises and updates a counter, in L21 so: `for (counter=0; counter<=final_number; counter++)`. The initial value of `counter` is 0. The loop condition: `counter` may not exceed `final_number`, a constant defined in line 3 of the program as 255. The expression `counter++` raises `counter` by 1, tantamount to `counter = counter + 1`, or 'add 1 to the value of `counter`, assigning the new value leftwards to `counter`, replacing the old value there'. Each repetition of the `for` instruction block raises `counter` by 1 until 255 is reached (and exceeded), when the loop ends due to the failure of the loop condition.

`fprintf` (the first `f` stands for 'file') then writes the value of `counter` in `outfile` according to the format `%3dd =` : the characters '`%3d`' prescribe a three-digit whole number, followed by '`d = `', as can be seen for instance on the second page of T02 at bottom left: '146_d = ' (font and size were set there manually, not by the program).

We have now arrived at the description of a user-defined function (`convert`). In each of the three calls to it 5-7 lines from the bottom of `main` there are three parameters in parentheses, a number (`2`, `8`, or `16`) and the variables `counter` and `digits`. The function itself starts at line 9: it does not return a value (no `return` at the end), which is why its type is `void`.

See the `int` variables `base` and `number` in parentheses at the top of the function, followed by the `char` variable `*code` (the `*` causes the final value of `code` to be sent back to `main` – see below). These three variables are directly linked to the parameters in the call: `base` successively takes the values 2, 8 and 16, `number` the value of `counter`. After *local* variables (valid only in `convert`) `i`, `power`, `buffer`, `remainder` and `divisor` are declared, `power` takes the value 1, `buffer` 255. While `power` is repeatedly multiplied by `base`, `buffer` is divided by `base` (terse like `++`, `a*=b` equals `a=a*b` and `a/=b` equals `a=a/b`): if `base` is 2, `power` successively takes, in 8 `while` repeats, the values 2, 4, 8, 16, 32, 64, 128 and 256, `buffer` in parallel 127, 63, 31, 15, 7, 3, 1 and 0 (note that 255/2 = `int` 127, not `float` 127.5): `(buffer>0)` is now false, `while` ends and `power` stays 256.

Now the value of `number` (= `counter` in the call) is stored in `remainder`, the value of `power/base` in `divisor` (256/2 = 128) and 0 in `i`. Another `while` loop runs, where:

1. `remainder` is divided by `divisor` and the result stored in `buffer`,
2. the remainder after dividing `remainder` by `divisor` is stored in `remainder` (`%` is a modulo operator, `%=` is treated like `*=`, `/=` etc.),
3. `divisor` is divided by `base` and the result stored in `divisor`.

The following value changes are implemented in 8 `while` runs – after the 8th run, `divisor` gets the value 0 whereby `(divisor>0)` is now false and `while` ends:

`divisor=` 128 →64 →32 →16 →8 →4 →2 →1 →0

If `remainder` (via `number` = `counter` in the call) is e.g. 186, in 8 runs it assumes the values 186%128 = 58%64= 58%32= 26%16 = 10%8 = 2%4 = 2%2 = 0;

and `buffer` 186/128=1, 58/64=0, 58/32=1, 26/16=1, 10/8=1, 2/4=0, 2/2=1, 0/1=0,

spelling 10111010, a binary number, which was calculated on `base` 2.

In the second line of the second `while` block in `convert` another user-defined `char` function is called: `hex_char`, defined in lines 4-8. It receives the `int` parameter `num` (value 0–15) and returns an ASCII symbol `num+48` (i.e. '`0`'–'`9`') `if num<10` else `num+55` (i.e. '`A`'–'`F`'). Thus `hex_char` delivers the hexadecimal representation of `num`, e.g. 8 becomes '8', 12 becomes '`C`'. The `if-else` structure is thus explained.

In `main` the binary, octal (`base` 8) and hexadecimal (`base` 16) numbers are stored as strings of characters: in the 8 runs of the 2nd `while` loop of `convert`, the index `i` of the 8 elements of `code` (cf. `digits`) takes on the successive values 1 to 8 (cf. `i++;`) and the characters sent by `hex_char` are stored in the `i`th element of `code`. After the loop, `code` additionally gets the finalising end-string '`\0`' and returns its content to `digits` in the `convert` calls. `fprintf` then writes `digits` in `outfile` (in format `%s` for 'string'), followed by '`b = `', '`o = `' and '`h = `' for `base` = 2, 8 and 16, respectively.

At the very end, the statement `fclose(outfile);` follows, the counterpart to `fopen` (the file is hereby 'sealed'), a statement of completion appears on the screen (with `printf`) and a zero is returned to the operating system, thereby ending the program.

In 1983 a conference of well-known synthesizer manufacturers decided to introduce a universal standard for the external digital control of their sound modules: *MIDI*, the *Musical Instruments Digital Interface*. This standard, requiring a relatively simple hardware connection, allows the communication between various music devices: the keyboard of a MIDI-equipped sound module can address the sound generator of another sound module; also, a computer can intervene and play the sound modules in real-time. On the outside, the MIDI connection usually has three five-pin standardised DIN sockets (from the German *Deutsches Institut für Normung* = the German Institute for Standardisation), labeled MIDI–IN, –OUT and –THRU – the first two are for receiving and sending MIDI-signals, and the last is a connection for passing signals through to other devices, allowing a compound of several appliances (see Γ22 for an example). The channel number coded in MIDI signals causes all following signals to be assigned to the sound generator momentarily set to this number; if, therefore, the signals of several channels go through all sound modules, each module would play only 'its own' part. The signals travel through a shielded two-core cable equipped with two five-pin standardised DIN plugs, this happening serially (i.e. as a bit series) with a transmission rate of 31250 bits, i.e. 3906 bytes per second.

If a sound is to be switched on, the *note-on* signal is sent to the generator, a one-byte hexadecimal number, of which the two nibbles are 9_h and channel number 0_h to F_h – thus the number 90_h indicates that a sound is to be played in the 'zeroth generator'. This first byte is called a *status byte*, the first nibble of which is between 8_h and F_h, depending on the type of command (i.e. the first, most significant bit is in this case always 1). The *pitch number* follows directly, the first of two *data bytes*, of which the first bit is always 0. All available notes, 128 in quantity, belong to the ten-and-a-half-octave chromatic scale from C_{-1} to G_9 – they are numbered from 0 to 127_d. The note C_4 corresponds to 60_d; in this way, the numbers of all notes with the same name C_n are divisible by 12. Before the note is played, its *force-of-attack* (called *velocity* by technocrats) must be given, which is also packed into one byte as a 7-bit-number between 0 and 127_d. The command 'play a C_4 of moderate loudness in channel zero' appears therefore (in decimal) as 144 60 64. Whereas programmers list the 16 channels from 0 to F_h, general sound-module users count these from 1 to 16.

As in the playing of a note, switching a note off or damping it is achieved by three bytes – the first is the *note-off* signal, $8m_h$ (m is the MIDI-channel number), the second is the *note number* and the third is the *damping speed* – however, very few sound modules react to this last information. The byte series $8F_h$ 60_d 127_d means 'quickly damp the C_4 in channel F_h'. Since each note needs at least six bytes to be turned on or off according to this definition, a maximum of 3906/6=651 notes can be transmitted per second.

A chord is possible through directly and sequentially switching its individual notes on – the 1½ milliseconds between a note-on and the next in a chord is short enough for the chord – if it is not too big – not be heard as an arpeggio. Notes can also be damped by the command series note-on, note-number, zero – in this case they are played 'without force' (this practice is of use in the economical so-called *running mode* – the status byte is entered at the start, to be repeatedly followed only by data byte pairs).

Two further important MIDI-commands are *instrument choice* (called *program change* by technocrats) – 2 bytes: Cm_h and i (m is the channel number, i is the instrument number from 0 to 127_d or more, depending on the sound module) and *pitch-wheel*, a feature that allows the parallel shifting of all pitches of a channel micro-intervallically, with 3 bytes: Em_h 0 and v (m is again the channel number and v is the shift as a number between 0 and 127_d, with 64_d as a normal condition; the range to top and bottom can be set from 0 to ±12 semitones by a special *systems command* specified by the manufacturer. At ±1 semitone the finest shift made possible through the second data byte is $^{100}/_{64}$ or under 1.6 Ct). Some sound modules also react to the first data byte, giving the pitch wheel a resolution of as much as 14-bits.

Finally, the versatile *control-change* command shall now be described: beginning with the status byte Bm_h, the first data byte indicates the type of application through the *control number*, like 7_d for the total volume (this allows a *crescendo/decrescendo* of a sustained sound) or 64_d for the simulation of a damper-pedal; apart from both of these (the values run from 0 to 127_d), some of the control numbers are standard, others are specified by the manufacturer – knowledge of their use can be attained in the sound module handbook. The second data byte determines the extent of the control: for control number 7_d, 0 means 'silent', 127_d 'full volume', for control number 64_d, control value 0 means 'no pedal', 127_d 'fully depressed'. T23 lists the eight main types of commands (those beginning with A_h, D_h and F_h are somewhat more special and are not discussed here).

All the signals described above are produced directly by one manually played sound module; they are discussed here concerning their production through a computer. L24a is a C program that plays the first bar of Schumann's *Happy Farmer* on a MIDI sound module; for instance, `play` in `main` sends values of pitch and force to the `play` function, where they are called `pitch` and `force`. The `play` and `damp` functions normally send three bytes via the function `send` to the MIDI socket (hexadecimal numbers are written in C with a `0x` prefix) – but the exact form of `send` depends on the type of computer; here, `send` only writes the values on the screen. The function `wait` checks the computer clock by way of `CLOCK_PER_SEC` and `clock()`, coded in `<time.h>`; it simply waits for the specified time duration. L24b is more compact: `score = fopen(FILENAME, "r");` opens the file `HAPPY_FARMER.TXT` (defined as `FILENAME` through the preprocessor directive `#define`) to read it. The loop `while` with the function `execute` runs as long as it returns the value 1 ('true') to `main`, i.e. as long as `input_amount` (see `fscanf`) has the value 3; if it is not, as when the input file end has been reached, function `execute` stops. The double equal sign '`==`' in `execute` is an equality comparison operator (cf. '`!=`', not equal to), different to the assignment operator '`=`' .

19 Harmony 4 – A Quantitative Approach to Harmonicity

Since ancient times it has been maintained that two notes of which the frequencies form a simple mutual ratio make up a ‘harmonic’ interval. But an ‘inharmonic’ interval, near in size to a harmonic one, falls, so to speak, into the pull of the stronger interval and through adjusted hearing seems to approximate the latter.

Can the simplicity of a ratio be expressed quantitatively? The ratios 1:2 (octave), 2:3 (fifth), 3:4 (fourth), 4:5 (major third) or 5:6 (minor third) appear ‘simpler’ than 8:9 (major tone), 9:10 (minor tone), and these in turn appear simpler than 15:16 (minor second), 32:45 (augmented fourth) or 45:64 (diminished fifth). N.B.: the more harmonic the interval, the smaller the numbers constituting the interval; yet, is the 3:5 major sixth more harmonic that the 4:5 major third?

The ratios missing in the list above, 6:7 and 7:8 as well as 10:11, 11:12, 12:13, 13:14 and 14:15, are not classical intervals – the numbers 8 and 9 were always preferred to the smaller 7, and 15 and 16 were preferred to 11, 13 and 14. It is striking, that while the historically preferred numbers are based on the prime numbers 2, 3 and 5, the others contain the higher prime factors 7, 11 and 13. In constructing a harmonic interval, both the smallness in size of the ratio-numbers as well as their divisibility is relevant. In measuring the *harmonicity* I tried to unite both these properties, which led in 1978 to the development of my so-called *Indigestibility Function* $\xi(N)$; see F25a. The power ^{2} is the *prime enmity factor*: raising it will cause the indigestibility of prime numbers to rise more steeply, and vice versa. It is a practical side-effect that $\xi(ab)=\xi(a)+\xi(b)$, as with logarithms.

T26a shows the indigestibilities of the numbers 1 to 100 – taking the first 16 in increasing order of their indigestibility gives rise to the series 1 2 4 3 8 6 16 12 9 5 10 15 7 14 11 13: the last four are the aforesaid ‘outcasts’. From the reciprocal of the sum of the indigestibilities of the mutually prime numbers P and Q, a function for the harmonicity of the interval can be constructed: the more indigestible P and Q are, the less harmonic the interval – see F25b.

In 1980 I asked each of twelve friends about what they thought was the order of difficulty in dividing a circle into 2-9 equal segments departing outwards from the centre of the circle. Their answers were on the average 2 4 (3 8) 6 (5 9) 7 (the numbers in parentheses are interchangable), evidently in vast agreement with the function of indigestibility. Unknown to me at the time, an interesting experiment had taken place in 1975 at Stanford University: test persons had been asked about their perception of the similarity of the ten digits 0 to 9 according to several criteria including ‘abstract quality’ (see Γ27a); the evaluated pairs of digits were then subjected to *multi-dimensional scaling*, a method which spatially places elements in an n-dimensional space such that their mutual distances correspond to their dissimilarities. A clear separation of the even numbers from the uneven was found as well as the separation of the prime numbers from the composite; in addition, the numbers got larger from left to right. In 1980 I discovered a structural link with indigestibility along the prime-composite number border.

T26c shows a listing of all just-intoned intervals within an octave with a harmonicity of 0.05 and above; the occasional minus sign indicates a polarity towards the upper tone of the interval, which then acts as root. Next to each intervallic ratio in the table appears its prime decomposition in powers of the prime factors; with the given *minimum harmonicity* (MH) of 0.05 the absolute power limit for the prime numbers 2, 3, 5 and 7 is the *maximum power series* (MPS) 7, 4, 2 and 1; prime numbers larger than these do not occur and have therefore the power 0. The intervals in this one octave under the given constraint number 37. With a range of three octaves and MH of 0.04, 240 intervals can be listed having an MPS of 10 6 3 2 1 1, graphically depicted in Γ27b. T26b shows the MPS and interval quantity at other MH-values with an arbitrary range of one octave.

According to the astronomer Daniel Kirkwood (1814-1895), the gaps in the asteroid belt between Mars and Jupiter were caused by 'commensurability', simple number ratios between periods of rotation of asteroids in the belt and that of Jupiter around the sun: due to resonances caused, material that was formerly present at positions of high commensurability in the belt was drawn towards Jupiter and destroyed at their thus elongated orbit's *perihelion* by the sun. Γ27c compares the density of the asteroid belt and the harmonicity of the aforesaid rotation intervals – a correlation is clearly apparent.

As to be expected, an increase in the MH reduces the interval density (the MH 0.1065 lets only a Mixolydian scale through!) and vice versa. There is a link between the MPS and the corresponding MH – see F25c; notice that an MPS also encompasses intervals of which the harmonicities are less than the given minimum value – they only guarantee that all intervals more harmonic than the minimum value will be included. With the MPS corresponding to a 0.03 MH (12 8 3 2 1 1 1) there are as many as 7533(!) different intervals in one octave, of which only 213 are as harmonic as 0.03 and above.

The MPS of the accepted standard interval tuning of classical Occidental (and Indian) theorists seems to be 9 6 2 0 0..., while interval formation in ancient Greece at Pythagoras' time was more like 9 6 0 0 0.... Both series look like hesitant beginnings of the series 9 6 2 1 1 1 0... based on an MH of 0.04 for one octave. Larger prime numbers inhibit harmonicity; because of this, theorists have always handled them with considerable caution – in analysing intervals, they in earlier times preferred monstrous conglomerations of smaller prime numbers to the perhaps more elegant solution of smaller products of slightly larger primes (compare the 3-limit augmented 4^{th} 512:729 to its 5-limit counterpart 32:45).

Following the 3- and 5-limit systems, the next complex ones are the *7-limit*, *11-limit*, etc., whereby it is implausible to expect convincing music with an arbitrarily large prime limit without an appropriately developed, well-considered musical grammar for the system that results from it.

Setting the minimum harmonicity to 0.02, we encounter no fewer than 256 intervals in the semitone range of 550 to 650 cents (see Γ28a$_1$); the most harmonic among these are the ratios 5:7 and 32:45 and (barely weaker) their inversions 7:10 and 45:64, all of which are intervals that could be regarded as tritones. A question that then arises is, which of these or other nearby variants are understood by the sounding of a 600-cent interval, a question that cannot be answered without an exact inspection of the musical context in which this happens.

As an interval, 600 cents possesses no obvious harmonic meaning, just like the syllables 'damp light', which, depending on the context, could also mean 'damn plight'. An interval described in cents or semitones is simply a piece of information on distance. Only contextually adjusted listening gives the interval harmonic sense. In order to demonstrate this I recommend you play the notes C D F E (W.A.Mozart) and C♯ B♯ E D♯ (J.S.Bach) on the piano and compare the effect of the major 3rd C–E with that of the diminished 4th B♯–E (see Γ28b$_1$). The clear tension difference comes from a more or less unconscious rationalisation of the notes to a desirable imaginary optimal tuning in the listener.

If, starting at a C, the notes 3, 5 and 6 semitones above it are sounded in order, it is not difficult to hear these as E♭, F and G♭. If however the notes appear in the order 0, 3, 7 and 6 semitones above C, then the series C, E♭, G, F♯ is suggested, as shown in Γ28b$_2$. The four notes of the first series would possibly be harmonically understood as 1:1, 5:6, 3:4, 45:64, those of the second series as 1:1, 5:6, 2:3, 32:45, seen in Γ28b$_2$ as cross-relations of the four notes among themselves, relatively simple ratios excepting those from C to F♯ and to G♭. However, F♯ (calculated at 32:45, 590 Ct) has a sharper, higher-pitched effect than the mathematically higher-presumed G♭ (45:64, 610 Ct), even if both series are played on an equal-tempered piano – this is probably because the two notes function even 'backwards in time' more as leading-tones than as structural ones, the G♭ leading down to the F, the F♯ up to the G: they are heard ('melodically') as pitch-distances and are intoned, particularly on instruments like the violin, microtonally closer to the notes to which they are anchored (the 'leading-tone phenomenon'). In this mixed system, then, the tuning of the melodically heard F♯ and G♭ – Schenker might call these 'prolonging tones' – is less compellingly relevant than that of C, E♭, F and G, which are heard more harmonically. Nevertheless, the following concerns itself with the harmonic, the relevance of the melodic ramifications remaining unconsidered.

To achieve harmonic insight into a pitch set, as such irrational, i.e. of known interval size but unknown interval ratios, it must be *rationalised*, i.e. cent values have to be changed into ratios. For compositional purposes I took on this task in 1978.

The solution I found proceeds as follows: at first, all permissable *alternative tunings* of the notes are investigated; the choice depends on two main factors: *minimum harmonicity* and *tuning tolerance*. In harmonically complexer music, the minimum harmonicity should be set lower than in harmonically simpler music; lower sensibility results in a higher tuning tolerance. Starting from a selected minimum harmonicity, a maximum power series is calculated according to the formula in F25c to produce an interval list (like the one in Γ28a_1). It is of course possible to determine the maximum power series arbitrarily, like e.g. 0, 0, 0, 5, 3, 0,..., a tuning based only on the primes 7 and 11; however, a series like this is difficult to make use of with compositional conviction and can sometimes sound like an out-of-tune presentation of another tuning deriving from smaller prime numbers.

In applying the tuning tolerance, a Gaussian bell-curve, named after possibly the greatest of its earliest discoverers Carl Friedrich Gauss (1777-1855), is set at the position to be tuned: this curve, of which the width (variance) is proportional to the given tolerance, damps – increasingly upwards and downwards in pitch – the harmonicity values as in Γ28a_2. All intervals that are far from the bell's centre or in any case harmonically too weak to assert themselves are not eligible as candidates. The pitch distance above and below the centre where the Gaussian damping factor reaches the arbitrarily chosen value of 20 is called the *nominal tolerance*. Γ28c shows this damping process in the tuning of a major scale in cents; the nominal tolerance was set here to 50 Ct, half of the smallest distance between two neighbouring steps of the scale.

The next step is to determine the *quantity of alternative tunings* for each note. Then the sum of the *harmonic intensities* (= the absolute harmonicity value) of all intra-scalar intervals involving all alternative tunings is evaluated, whereby the alternative tunings for any one scale degree are excluded from mutual comparison: the tuning constellation chosen is the one with the highest harmonicity sum. If the number of the pitches to be tuned is e.g. 8 and the number of their tuning alternatives is 3, the number of tuning constellations is 3^8=6561; an 8-note-scale contains 28 intra-scalar intervals – the formula for this is $n(n-1)/2$, where n is the number of notes – therefore in this case a total of 6561×28=183708 harmonicity values must be added to find the optimal tuning, a task for a computer! It proved more efficient programming to add not the harmonicities but rather their reciprocals the inharmonicities (in each case the sum of two indigestibilities), whereby the smallest sum total is picked. The doubled* total number of intra-scalar intervals $n(n-1)$ divided by the minimum inharmonicitiy sum is what I call the *specific harmonicity* of the optimal tuning of the pitch set.

*using the doubled number excludes the factor 2, thus saving computing time

The tuning of one octave of a major scale finds the following to be the four most harmonic alternative tunings (AT) for each of the 8 pitches, using a minimum harmonicity (MH) of 0.03-0.05 and a nominal tolerance (NT) of 30-50 Ct (see T29a$_1$):

1/1	9/8	5/4	4/3	3/2	5/3	15/8	2/1
224/225	10/9	81/64	27/20	40/27	27/16	243/128	225/112
225/224	28/25	512/405	75/56	112/75	42/25	256/135	81/40
80/81	125/112	63/50	1701/1280	125/84	2048/1215	40/21	448/225

After 65536 checks of all 28 intra-scalar interval relationships of the 4 best candidates of the 8 scale degrees (4^8=65536), the computer chooses the tuning combination 1/1 9/8 5/4 4/3 3/2 5/3 15/8 2/1 (here all from the first set of ATs), a universally accepted solution known for a very long time – see also Γ13a. The specific harmonicity of this tuning is 0.2252.

In the harmonic minor scale, the following equally well-known tuning constellation was found to be optimal under the same conditions: 1/1 9/8 6/5 4/3 3/2 8/5 15/8 2/1. The specific harmonicity is 0.2032, as seen in T29a$_2$.

The pentatonic scale (C D E G A c, not shown in T29a), is tuned under the same conditions as above as 1/1 9/8 5/4 3/2 27/16 2/1 (the classical Pythagorean tuning; the Pythagorean major sixth 27/16 was taken from the second set of ATs because it agrees better with the major 2nd 9/8, major 3rd 5/4 and perfect 5th 3/2 – compare the respective numerators and denominators). The specific harmonicity is 0.2387.

The whole-tone scale is tuned at an MH of 0.03-0.05 and a NT of 40-50 Ct as 1/1 10/9 5/4 64/45 8/5 16/9 2/1 at a specific harmonicity of 0.1615 (seen in T29a$_3$).

An interesting case is the Bohlen-Pierce Scale (BP-Scale), in which the perfect 12th (1:3 or 1902 Ct, sometimes called the *tritave*; should the octave be renamed '*bitave*'?) is divided into 13 equal intervals. The cent values of the scale degrees are therefore 0 146 293 439 585 732 878 1024 1170 1317 1463 1609 1756 and 1902. Its rationalisation yields different tunings for various MH and NT values (for reasons of computing time economy only 2 ATs were examined – see T29a$_4$). MH 0.03-0.04 and NT 40-50 yielded 1/1 35/32 6/5 9/7 45/32 3/2 5/3 9/5 2/1 15/7 75/32 5/2 25/9 3/1 at a specific harmonicity of 0.12. The pitch deviations from the input cent values are 0 +9 +23 −4 +5 −30 +6 −6 +30 +2 +12 −23 +13 and 0 Ct. The constraints MH 0.05 and NT 10-30 Ct cause problems: a conflict between the relatively high MH and low NT results in fewer available tuning possibilities, so that the 1st and 2nd scale degrees as well as the 8th and 9th (at NT 10-20 Ct) are tuned identically. Only MH 0.04 downwards gives unambiguous rationalisations, with an average of higher pitch deviations. The BP just intonation given by other sources based on 3:5:7 chains is 1/1 27/25 25/21 9/7 7/5 75/49 5/3 9/5 49/25 15/7 7/3 63/25 25/9 3/1. The pitch deviations are in this latter case lower than in the rationalised version given above, which was based on intra-degree harmonicity considerations.

In T29b the ratios and harmonicities of the 78 interval relationships between the 13 steps of a twelve-tone equal-tempered octave are shown; this optimal tuning comes from the following conditions: MH 0.04, NT 30 Ct, AT 2 (a 3 here would lead to 1,594,323 rationalisations!). This method is also applied to the 14 steps of an octave divided into 13 equal intervals. The same MH (0.04), NT (30 Ct) and AT (2) result in a tuning network also shown in T29b.

Comparisons of the results of various MHs and NTs with these two equal tempered scales were also undertaken, graphically displayed in Γ30. The goal was a kind of landscape in which the relevance of a tuning solution is easily surveyed through its spread. For this purpose I developed a compact graphic method to represent the rationalisation and size of intervals called *ratioglyphs* (from the Latin *ratio* = ‘reckoning’ and the Greek *glúphein* = ‘to carve’), which is explained by the schematic at lower right in Γ30a and b (based on 2:3, 15:16 and 32:45 as well as on 16:21, 128:243 and 24:35): the upper half of each ratioglyph portrays the prime power along a ‘trunk’ to which the prime numbers are attached in rising order; the powers of these are given as lengths of horizontal ‘branches’ to the left for negative powers and to the right for positive ones. If the branch becomes too long it can twist upwards in order to save space. The lower half of the ratioglyph shows the interval size in cents, first in hundreds and then – as a deviation from the hundreds – in tens and units. In this way each tuned interval receives a precise, visually assimilable form through its ratioglyph.

In addition I developed a notation for note names, in which the intervallic origins of the given notes are made uniquely apparent: a number to the right shows the number of 5^{ths} (e.g. ^{3} or $_{3}$ for rising or falling), an accent shows the 3^{rds} (e.g. ` for rising, ´ for falling), and a question mark indicates the 7th (? for rising, ¿ for falling – see below). The note ¿f´^{1} is thus (Ω)+Q–T–S (or 2/1 × 3/2 × 4/5 × 4/7 = 48/35 or 547 Ct) above C (S means the *natural 7th* 4:7). In the case of multiple 3^{rds} or 7^{ths} the symbol is repeated, e.g. ``` or ??.

Returning to the ratioglyphs in Γ30: in both scales a special tuning solution stands out that is the most plausibly recommendable because of its wide dispersal: shown against a grey background in Γ30, this tuning holds in the 12-tone scale with an NT of 40 to 60, and in the 13-tone scale around NT=40 (a higher NT appears to be inappropriate in this case; after all, given enough NT all notes would be tuned to an octave or a fifth!). The multi-dimensional scalings (MDS) of both these optimal tunings show in each case an area in which the notes are spatially allocated according to harmonicity – the closer two notes are shown to each other, the more harmonic the interval between them is. These MDS-fields can be considered as ‘maps for harmonic modulation’ and can be applied to n-limit systems (notice e.g. the family grouping of the ?-notes on the one hand and the ¿-notes on the other in the MDS in Γ30b).

The rhythmic organisation of music generally shows a more or less strong dependancy on an internal hierarchical *metre*, a series of regular points in time, acoustically activatable in various ways. The *stratification* of a metre, the inner acoustical hierarchy of which is multiplicative, is shown in Γ31a by means of a $^{12}/_{16}$ bar, which exhibits a stratification of 2×2×3. If this stratification is examined in further depth with 32nd- (𝅘𝅥𝅰), 64th-notes etc., the geometric series given could proceed by constant bisection ..×2×2×2×2..; the position of the series' last divisor which is not 2 determines the order of the metre – according to this reasoning, $^{12}/_{16}$ is a metre of 3rd order, $^{6}/_{8}$ (=2×3) of 2nd, $^{3}/_{4}$ (=3(×2×2...)) of 1st and $^{4}/_{4}$ of 0th order.

Independent of the order of the metre, the individual pulses on every level of the stratification exhibit a variable metric relevance. Assuming the 0th (highest) level to relate to a full bar, a $^{12}/_{16}$ metre has on its 1st (second-highest) level two dotted quarter-note (♩.) beats, a metrically stronger followed by a metrically weaker. On lower levels there is a lack of sufficiently differentiated verbal descriptors – the evaluation of the pulse-strength of the four dotted 8th-notes (♪.) already on the 2nd level necessitates the introduction of numbers (here 'beat' means the same as the more general term 'pulse' only on the two highest levels – the 0th and the 1st).

The diverse acoustical activity of the pulses of a metre – in the simplest case, they are played or skipped – generates a rhythm that supports the metre to a higher or lower degree, i.e. the *metric field strength* can be consciously increased or decreased, with the rhythm varying between metric (with a clearly recognisable basic beat) and ametric (without a recognisable basic beat). This can happen by defining and employing pulse-strength in such a way, that in order to support the metre the stronger pulses appear more frequently than the weaker; in an ametric rhythm, pulses of all strengths would be equally frequent. Analogous to this, using a pitch scale, the *tonal field strength* or 'degree of tonality' can also be influenced by how often the notes of the scale are used, based on the harmonicity of the intervals between the notes and an arbitrary 'tonic'. Γ31d shows the relationship of field strength to probability and the resulting 'oftenness' (frequency of occurrence) of the elements – pulses of a metre, notes of a scale – as a straight line of variable gradient.

The ascertainment of sufficiently differentiated pulse-strength values remains to be gone into: this implied providing a unique evaluation of each pulse of a metre, so that no two pulses could be perceived as equally strong. I found this differentiation missing in diverse systems at the end of the 1970s and decided to search for my own appropriate method.

In 1978 I was able to develop a formula that calculates a pulse-strength evaluation (or *indispensability* of attack, as I call it) for every pulse at every level of a multiplicative metre of any order; from this formula, the values for the six 8th-notes of a $^3/_4$ bar are 5 0 3 1 4 2, those for a $^6/_8$ bar are 5 0 2 4 1 3. Both series of numbers show the strongest emphasis on the first 8th-note; the second strongest in $^6/_8$ is on the fourth, in $^3/_4$ on the fifth 8th-note. Here are the indispensabilities of the twelve 16th-notes (𝅘𝅥𝅯) of the third stratification level of the metres $^3/_4$, $^6/_8$ and $^{12}/_{16}$ (see also Γ31c):

$^3/_4$	(3×2×2) :	11	0	6	3	9	1	7	4	10	2	8	5
$^6/_8$	(2×3×2) :	11	0	6	2	8	4	10	1	7	3	9	5
$^{12}/_{16}$	(2×2×3) :	11	0	4	8	2	6	10	1	5	9	3	7

The values for the first and second levels are also contained herein: to make these evident, subtract the difference between the quantity of pulses at the level shown above and those at the desired level from the indispensability, keeping only non-negative numbers, e.g. for $^3/_4$ (the number of pulses on the 3rd level as shown above is 12):

1st Level (pulse quantity 3: subtract 12−3, i.e. 9): 2 – – – 0 – – – 1 – – –
2nd Level (pulse quantity 6: subtract 12−6, i.e. 6): 5 – 0 – 3 – 1 – 4 – 2 –

Notice that at all levels, the indispensability of the first pulse is always one less than the number of pulses, and that that of the second pulse is always zero. The indispensability values' relevance can be illustrated through a so-called dilution process: if the 'more dispensable' pulses at the initially attack-saturated x^{th} level of a metre are gradually removed, the impression of the metre remains intact, as e.g. on the penultimate levels of the $^3/_4$ and $^6/_8$ bars shown in Γ31b.

The indispensability formula is designed for the successive division of a metre by arbitrary prime number divisors. A prerequisite for each divisor is a *fundamental indispensability* series for the metric level containing the corresponding prime-quantity of pulses, e.g. for 2 the series 1 0, for 3 the series 2 0 1, for 5 the series 4 0 3 1 2 etc. In a metre with the stratification $p_1 \times p_2 \times p_3 \times \ldots \times p_n$, the indispensability $\psi(n)$ – note the lower-case ψ – of the n^{th} pulse is given by the formula in F32a. A valid prime-number fundamental indispensability series is given by the function $\Psi(x)$ – note upper-case Ψ – shown in F32b, where x corresponds to the part of the formula in parentheses immediately following Ψ_p in F32a.

The prime-number fundamental indispensability Ψ can be personally estimated or worked out by the formula in F32b; the basis here is the indispensability series for a metric level with one pulse less, in which the highest-level divisors are the largest (e.g. to find Ψ for a cycle of 23 pulses, the values are first calculated according to the ψ-formula for 22 pulses, whereby 22 is represented as 11×2; the values for 11 pulses are based on those for 10, understood as 5×2, etc.). The dropped pulse is reinstated between the last two pulses of the 'reduced' metre (e.g. $^4/_8$ standing in for $^5/_8$).

23 Metre 2 – The Measurement of Metric Coherence

In search of a method to determine the similarity between two metres of different stratifications and speeds, I was able to satifactorily employ the indispensability method. Compare for example a 2×2×3-metre in bar-tempo MM20 with a 3×5-metre in bar-tempo MM16. They share a common lowest-level pulse-tempo of MM240; also, 5 bars of the first metre are equal in duration to 4 bars of the second (containing in each case 60 lowest-level ‘elemental’ pulses in the time period between two successive coincidences of pulse 1). Thus one can lay, pulse for pulse, an indispensability series applied five times for 2×2×3 pulses alongside a series applied four times for 3×5 pulses, as shown in Γ33a. If the two metres have no common elemental pulses, their stratification must be continued further down through additional divisors until this is the case; for example, a 2×5-metre at a bar-tempo of MM50 (elemental pulse-tempo MM500) and a 3×2-metre at MM60 (elemental pulse-tempo MM360) are extended to 2×5×3×3×3×2 and 3×2×5×5, respectively, in order to reach a common elemental pulse-tempo of both metres at MM9000. Here too, the largest divisors are allotted to the highest levels (e.g. the extension of the 2×5-metre is ×3×3×2, not ×2×3×3).

The best method that I could find to determine the metrical similarity was to multiply the *relative indispensability* (nominal value divided by the maximum value, i.e. by the number of pulses minus one) of each elemental pulse of one of the metres by the relative indispensability of the concurrent pulse of the other metre, and then to establish the sum of all thus obtained products; coinciding stronger pulses have an advantageous effect on the sum total, and one has an even better effect if the products of the relative indispensability is squared before addition, similar to RMS evaluation. It turns out that the average product-square (let us call it ‘aps’) is larger for clearly related metres (for example for 2×3 and 3×2 in the same bar-tempo: 0.3245) than for unrelated metres (for example for 2×2×3 at MM20 and 3×5 at MM16: 0.1573). With this process it happens that for the least related metres the aps tends down towards $^1/_9$ (the aps of all pairs of numbers less than or equal to N is $N^4/9+N^3/3+13N^2/36+N/6+1$); thus 9aps−1 will tend towards zero. With half the negative reciprocal of the natural logarithm of (9aps−1)/3.5 (3.5 is the highest possible 9aps−1 between two 2×2 metres of identical bar-tempo) I found a scaling which to me is the clearest and most convincing; this coefficient is defined in the formula in F34.

T35a is a table of the calculated metrical ‘similarity’ of the four three-level metres upto third order with pulses not exceeding 12 in number (2×2×2, 2×2×3, 2×3×2, 3×2×2); the tempo ratios are determined here by combinations of the whole numbers 1 to 3. The first value in this table – 0.46382 as a metrical similarity between a 2×2×2-metre and itself – shows that the word ‘similarity’ causes a problem; here one should surely find an identity, more plausibly expressed by the value 1.0. The identity between 3×2×2 and itself is indicated (in the same table) however with 0.41454 – this way one can see that this measurement in general also takes the metric simplicity into consideration – 2×2×2 is simpler than 3×2×2.

For this reason I have decided to call this property that combines the similarity correlation between the two metres with the internal individual simplicity correlation of their pulses *metric coherence*. The actual 'similarity' can only be determined by a series of attempts, whereby one of the metres remains common to all comparisons: the coherence calculated by the comparison of the fixed metre with another metre divided by the coherence of the fixed metre with itself (*autocoherence*) gives the metric similarity; with two identical metres this is therefore 1, the value of the *identity*.

At the time that I developed the formula for metric indispensability, I also drew up a formula for the harmonicity of pitch intervals (see Chapters 19-21), which assigns to any frequency ratio P:Q a coefficient for its harmonicity. In harmonically 'strong' intervals like the octave 1:2 and the fifth 2:3 this value is larger than in 'weaker' intervals like the tritone 32:45. If one were to consider an audible pitch as an extremely rapid series of pulses, of which the tempo is the pitch's frequency (I call this the *frhyquency*, the 'rhythm-frequency'), the harmonicity would be have to be a kind of 'micrometric coherence'! The table in T35b shows the metric coherence of 32 different pairs of metres of the same bar tempo; for comparison, the harmonicities of the corresponding pitch-intervals are also included. The parallelism between them is even more visible in graph form – see Γ33b. The question then arises: 'Is harmony a special case of polymetre?'.

This prompted me to apply my method of pitch rationalisation to rhythm. Manually played from a music score, a rhythm was stored in milliseconds in the computer and converted to cent values of pitch as follows: an arbitrarily chosen time unit was allocated to an arbitrary cent value; according to the formula for converting frequency quotients into cents, each measured *delta-time* (the time difference between one event and the following) was converted into cents – in this way, a duration twice as long as the fundamental time unit was represented by −1200 Ct and a time half as long by +1200 Ct. The resulting pitch set was then rationalised according to the harmonicity method (with the parameters minimum harmonicity, nominal tolerance, and alternative tunings) and then converted back to rhythm. The results were completely satisfactory: tuplets (like triplets, quintuplets) as well as dotted rhythms were placed at the positions dictated by the score, although the played input was rhythmically not at all precise.

According to an analysis of the main entries in the Concise Oxford Dictionary (11th edition revised, 2004), the most common letter of the English language is E – it made up about 11% of all letters. A followed with 8½%, then R, I, O, T, N, S, L, C, U, D, P, M and H, each with a frequency of occurrence of 7.6 down to 3%. The remaining letters were less frequent, with X, Z, J and Q bringing up the rear, under 1%. Other languages exhibit other frequencies of occurrence. If a text were to be randomly generated abiding only by these Oxford Dictionary statistics, it could look like this:

...DHEOBNATECRCETGUSSIEVIRDEYLIAENDESRNSHAVODICAETCLTBERNETMAAKB...

This does not look much like English; the relative frequency of the letters is satisfactory, but their order is not. Therefore it is not isolated letters that should be counted, but rather their combinations, e.g. bigrammes, i.e. all pairs of letters like AA, AB, AC, ... BA, BC etc., whereby the ones most common in English are – after one authority – TH, HE, AN, IN, ER, RE, ES, ON, EA, TI, AT, ST, EN and ND, in decreasing order. Longer chains, trigrammes, lead to more faithful syntheses, of which the most frequent according to the same authority are THE, ING, AND, HER, ERE, ENT, THA, NTH, WAS, ETH and FOR. T36a shows the statistics of all individual letters as well as the most frequent bigrammes from the English words of this chapter; the number of appearances is given to the left of the '×'.

Chains of elements, like in this case letters, or of any kind of general symbols, were examined statistically as described above by the mathematician Andrei Andreyevich Markov (1856-1922) at the beginning of the 20th Century; in this context they are thus called *Markov-chains*. The longer the chains, i.e. the higher their order, the more they say about the general behaviour of the elements and the closer a re-synthesis is to the original source. T36b lists at the top an excerpt from a sentence about Markov ('At the start of the twentieth century, the Russian mathematician Andrei Andreyevitch Markov developed a method'). Under that, eight re-syntheses of the text are listed, corresponding to the Markov-orders 0 to 7; the gradual transition, from gibberish of order 0 (in which each letter is taken for itself), to something pronounceable of order 1 (based on bigrammes), to order 2 (trigrammes), is clear. Following this are simply linguistic improvements.

Markov orders are as such whole numbers, but real-number orders can also be realised as follows: in a real order $W+F$ (where W is a whole number and F is a fraction) one takes $100(1-F)\%$ of the evaluations in order W and $100F\%$ in order $W+1$, e.g. with the order 2.2 ($W=2$, $F=0.2$) 80% of the syntheses are of order 2 and 20% of order 3. T36b shows at the bottom four additional syntheses of orders between 1 and 2 – a gradual linguistic transition is also recognisable here.

Music elements, for example notes, can be analysed and re-synthesised according to the Markov method. Γ37a shows a fragment of a J.S.Bach Invention in its original form as well as a Markov re-synthesis of the order 0-7: a gradually decreasing 'chaos' is also perceptible here. From this phrase, 32 individual 16th-notes (staccato, smoothed out to legato in the notated example) have been listed here, whereby '–' shows the original rests:

--C-D-E-F-D-E-C-G---C'---BAB-C'---

The notes C–C' (C' is an octave above C) tally 2 2 2 1 1 1 2 2 plus 19 rests. Starting here, the synthesis of the zeroth order appears as follows:

D-BA-E-B-EBC'---EE-D-EEBF--B--F--

In this case, the same eight notes tally 0 2 6 2 0 1 5 1 plus 15 rests; the deviation in frequency of occurrence from the original would be gradually minimised in a prolonged re-synthesis. Raising the order increases the resemblance to the original; by the 6th order the original in this example is restored.

Stochastics (from the Greek *stochastikós* = 'skilful aim' in the sense of 'cleverly guessed') is the study of the probability of random events. In 1957, the composer Iannis Xenakis (1922-2001) introduced the term *stochastic music* for statistical sound calculations.

The probability values employable in a composition can be gathered not only from already existing music; the point of departure is more frequently rooted in music-theoretical or compositional considerations. Γ37b depicts a case in which a series of 9 conditions of probability are translated into musical notes. The leftmost graph ('1') shows a full-length line at the MIDI-note 76 (E_5, see the x-axis), corresponding to a 100% probability for this pitch, which is confirmed in bar 1 of the musical notation below. Graph 9 shows the same thing for the MIDI-note 66 ($F\sharp_4$): see bar 9. Between these, the graphs show varying probabilities for the notes from E_5 to $F\sharp_4$: a bell-shaped curve appears at the right and shifts to the left, where it disappears. The musical notation below the graphs have been generated according to the probabilities shown here (adding up in each and every stage to 100%); the higher the probability, the more frequent the corresponding pitch will be in general. However, the individual measures would have to be much longer to keep well to the actual prescribed probability of the notes; in any case, the musical example shows – according to the graphs – a gradually descending series of pitches from E_5 to $F\sharp_4$.

Various methods for generating stochastic music can be employed using a computer; the simplest method is to divide the range from 0 to 1 into as many zones as there are elements to be chosen from, where the width of a zone is proportional to the corresponding probability (e.g. for 50% there would be a zone of 0.23 to 0.73 – see the pentatonic scale example in Γ37c); after this, one throws a series of randomly generated numbers between 0 and 1 into the area divided as described and then notes the zones into which they have fallen.

If a sound wave is generated by putting several simultaneous sine waves together, it is formed by the numerical addition of these sine waves, as can be seen in Γ38a: in this case, two curves are added in the 1:2 frequency ratio of an octave, evident from the lengths of their respective periods. The amplitudes of the two curves also act here as in the ratio 1:2 – the slower curve swings twice as far out in the y-dimension as the faster curve, shown graphically below as the spectrum of the resultant bold curve, a curve that is itself periodic because of the periodicity of its components. The spectrum plots the amplitude or sound pressure of the partial frequency on the y-axis; spectra can also show sound intensity or dB-levels.

Γ38b shows a somewhat richer spectrum – in this case, there are five partials with the mutual frequency ratios 1:2:3:4:5 and amplitude ratios $^1/_1$:$^1/_2$:$^1/_3$:$^1/_4$:$^1/_5$. Γ38c continues with 25 partials; here, too, the n^{th} partial has the frequency nF and the amplitude A/n, where F and A are the frequency and the amplitude, respectively, of the first partial or fundamental. The conspicuous wiggles in the summation curve have become smaller but greater in number in comparison to Γ38a and b, a process that is continued to form the curves in Γ38c and e (called a *sawtooth wave* because of its shape).

Fourier Analysis – a trigonometric method named after its inventor, the mathematician Jean-Baptiste Joseph Fourier (1768-1830) – untangles the individual sine-components of a periodic curve like the one above: one period of the curve serves as the input; as output one gets among other things the amplitudes of the components. The spectrum in Γ38e shows that the sawtooth curve drawn there corresponds to the sum of the first 500 partials with amplitude A/n, and that the *square wave* in Γ38f is the sum only of odd-numbered partials but also with amplitude A/n; the *triangular wave* in Γ38g also consists only of odd partials – however, the amplitudes in this case are squared ($^1/_1$, $^1/_9$, $^1/_{25}$ etc. of the fundamental amplitude), and: the phases of the partial numbers 3, 7, 11 etc are reversed! Γ38h shows spectra and waves of the amplitude function $^1/_\xi$ where ξ corresponds to the indigestibility function that I use for measuring harmonicity (see Chapter 19) – the 'more digestible' of the partials are in this case louder than the 'less digestible'.

In this way one can calculate the *harmonic* spectra (containing whole-number frequency ratios) of all periodic curves. Among the various forms of Fourier Analysis, the *Discrete Fourier Transform* (DFT, sometimes referred to by its faster version FFT, *Fast Fourier Transform*) is very common; the analysis proceeds in time-frames that correspond to a hypothetical period length ($N=\mathfrak{S}/f$, where N is the number of samples per time-frame, $\mathfrak{S}$ is the sampling rate and f is the spectral fundamental frequency). Sound waves of all types can be analysed based on an arbitrary fundamental, the spectrum of which is calculated from one time-frame to the next. The frequency of this fundamental determines the density of the information in the frequency region being investigated: if the fundamental is only just below this region, the partials within the region could be too widely separated. It is important to know that the amplitudes in the DFT-extracted spectrum are valid in their relation to the partials' absolute pitch, more or less independently of the fundamental frequency.

The bold curve in Γ38d has a sawtooth spectrum as in Γ38c, although this is visually not so evident – in contrast to all of the preceding examples, the partials in this case are of unequal phases, i.e. the sine components start their individual periods at different angles, although the acoustical result sounds the same as the sawtooth sound with partials of equal initial phase (see again Γ38c); the spectrum is also derivable through a DFT here. In listening to sounds, the human ear acts similarly; individual partials, particularly lower ones, can be heard individually – independent of phase – when listened to carefully.

In this way the note G_3 (392 Hz) played on a bassoon was analysed and illustrated as in Γ38i-l; one sees – connected by splines – the relative amplitudes in % and the coresponding loudness level in dBu against the pitch in semitones and partials. Above the 12^{th} partial (4704 Hz), the level drops below −30 dBu or 3.2% amplitude and is no longer shown.

Fourier Synthesis, also called *additive synthesis*, is the reverse method, described here at the beginning for constructing sound waves; it is not suitable for generating noise. If for example, a synthetic piano sound is desired, spectra of successive periods of the sound wave of a specific note on the piano are determined through Fourier Analysis and stored digitally. Since the sound (especially directly after the initial striking of the key) is not static in time, several periods up to a length of about 0.3-0.4 seconds (about 100 periods for C_4) are analysed; in this case it is not necessary to store the spectra of each and every period – they can instead be additively fused into time-frames of e.g. 10 ms. Following this simulated synthesised attack, the rest of the sound and its decay can be realised by the looped repetition of the last period(s), gradually faded out in loudness. In order to achieve natural sounding timbral transitions, this process is repeated for a sufficient number of fundamental pitches, spaced at best up to not more than a major third apart. With this spacing, 22 fundamentals (7 octaves) × 30 time-frames (300 ms) × (say) 50 partials would result in 33000 numbers, which would require, for example, for 16 bits each, a storage capacity of about 64 kilobytes. From this material one can calculate for composed pitches and loudnesses sound waves of longer duration and then convert these from digital to analogue; however, this (oversimplified) ‘piano sound’ lacks typical noise components mainly produced by the striking of the hammers.

Γ38m shows a hand-drawn wave period, the Fourier Analysis of which is shown up to the 99^{th} partial in T39. This analysis forms the base for a re-synthesis, with 6, 16 and 75 partials as shown in Γ38n-p; one cannot fail to notice how, with an increasing number of partials, the original form is gradually re-constructed.

In the early years of electronic music, introduced around 1952 significantly by the composers Herbert Eimert (1887-1972) and Karlheinz Stockhausen (1928-2007), analogue sound production was achieved mainly with simple sound generators (which usually offered only simple wave forms like sine, sawtooth and square); in addition to this there were noise-generators and diverse sound-transforming modulators (e.g. the ring-modulator). From 1957, Max Mathews (1926-2011) enabled a larger diversity of calculable tones and noise to be produced by digital means, which was, however, a very cumbersome alternative due to the relatively low speed of the computers at the time as well as their small storage capacity. Today – thanks not least to the introduction of MIDI – the array of available devices includes a large range of synthesizers and samplers.

The technique of sampling – which has advanced to be the most important method of sound production today – is rather simple: for a satisfactory reproduction of e.g. a well-known type of sound like that of the piano, the original sound wave of notes every three or four semitones apart over the complete required range, recorded through a microphone, is AD-converted and stored as a series of numbers (samples). Waves of intervening fundamental pitches are simulated through interpolation. In order to use the storage capacity for long sustained sounds sparingly, only the beginning of the sound (up to about 300-400 ms), important for its recognition, needs to be rendered; the remainder can be formed from a few typical periods at the end of this initial portion, which are then repeated in a loop, the amplitude of which is shaped by an appropriate envelope. At a spacing of a major third and a sampling rate of 44100 per second, 22 waves (7 octaves) × 0.3 seconds (300 ms) × 44100 samples would result in 291060 numbers, which would require, for example, for 16 bits each, a storage capacity of about 4.4 megabytes.

If new, hitherto unknown types of sounds were desired before 1980, one had to turn to general technologies like calculated, DA-synthesised sound waves, which was a very time-consuming matter then. Or one employed Fourier Synthesis with several sine generators sounding simultaneously in real-time, which was complicated because of the quantity of sound producing devices necessary for the partials of a sound. Or one had to rely on analogue synthesisers, which had the ability to produce few and relatively specific sounds depending on the given electronic circuitry. However, methods like *Frequency Modulation* (FM) or *Phase Distortion* (PD), developed in the 1980s, proved to be relatively uncomplicated; they allowed a large spectral variety to be obtained by the clever use of relatively few sound generators.

The FM-method published in 1973 by John Chowning (*1934) has been employed in a different form for quite some time in broadcasting: a sine wave frequency, fluctuating slightly around a mean value over time, can be analysed as a frequency-time curve. If these fluctuations are relatively slow compared to the periods of the wave of mean frequency (the *carrier*), the decoding can be carried out more precisely: FM-radio works this way, in which a high carrier frequency (88-108 MHz) is *modulated* by a much more slowly vibrating sound wave (20-20000 Hz). If the carrier wave and the modulation are close together in frequency, they are more difficult to distinguish from each other – through their mutual influence, spectral side-bands originate which enrich the sound timbre. In this way, Chowning was able, given only two frequencies, to produce several more. The basic formula for a curve subjected to FM has the form $e=A\sin(\alpha t+I\sin\beta t)$, where e is the deviation of the wave from its position of rest, t is time, A is the maximum deviation (amplitude), and α and β are the periods of the carrier and the modulation waves, respectively. I is the *modulation index*, which influences the resultant spectrum. Γ40b shows a carrier wave of 100 Hz modulated by 10 Hz with a modulation index of 0.2. The resulting wave (in Γ40a still unmodulated), 'swollen' towards the middle and 'squeezed' at the edges – better visible with higher indices (reaching 15 in Γ40h) – shows by Fourier Analysis (Γ40b-e) symmetrically outward-moving side-bands (the partials are displayed as multiples of 10 Hz); from Γ40f on, the bands, their lower ends shifted into the negative frequency area, reappear with opposite phase in the positive region, thereby damping other frequencies present there. The amplitude development of the side-bands can also be calculated by *Bessel Functions*, named after the mathematician Friedrich Wilhelm Bessel (1784-1846); the explanation of these functions is beyond the scope of this book. With modulators equal to and exceeding the carrier frequencies, the forms shown in Γ40i-l are generated at indices 1 and 10, respectively. FM can be cascaded: the modulation can itself again be modulated, and so forth – the first known FM-synthesizer, the Yamaha DX-7, employed six tone-generators linked to each other in 32 different ways.

Phase Distortion (PD) was developed by several researchers approximately at the same time as the origins of FM sound synthesis – expressed in highly simplified terms, a wave is sent through a table containing substitute values assigned to every possible input y-value. Γ40m shows five different results: the original curve shown at top left (here a sine wave) is subjected to the one-to-one correlations shown in the grey-filled boxes to the right (showing the 'old' y-value on the x-axis against the 'new' value on the y-axis): in the three rightmost examples, shown in the middle of the diagramme, the output is clearly much more complicated and richer in partials than the input-signal – see the spectra at the bottom. A more basic form of PD-synthesis is the application of a variable sample rate to the sound wave (in this case, the meaning of the term phase distortion is more understandable): here, too, one can achieve complex sounds with very simple means.

Γ38i-l show a number of parallel vertical lines indicating on the x-axis the frequencies of the partials of a bassoon spectrum and the corresponding amplitudes and loudness levels on the y-axis. All spectra containing partials can be illustrated like this, both harmonic (in which the partials make up whole-number frequency ratios and therefore form a harmonic series) as well as inharmonic ones (where the frequencies do not form an harmonic series, e.g. as in many types of bells). An acoustical phenomenon of this sort comprising a number of partials is called a complex tone: its spectral representation consists solely of parallel vertical lines; if the spectrum is harmonic, the sound wave is periodic.

Compare this with the spectrum shown in Γ41a of a forcefully struck tam-tam. Instead of parallel lines ones sees a curvy envelope; the timbre of this instrument, a noise, contains no perceivable sine-components. What kind of frequencies are contained in this type of spectrum?

It is generally assumed that a noise-band bounded by two frequencies contains 'all the frequencies in between' and that a point on the spectral envelope indicates the amplitude of the frequency at this point on the x-axis. According to this scenario, even the narrowest band of noise would have to consist of an infinity of sine waves (because for each pair of neighbouring frequencies there would have to be another one between them), which either add up to a sound wave of infinitely large amplitude or would each have to have an infinitely low loudness level (zero).

This portrayal is not useful. Here it will be shown that noise-spectra have stochastic properties, i.e. that they work according the principles of probability and chance. In Γ41b, 882 arbitrarily generated random numbers are seen in graphic form (98 dots in each of 9 vertical boxes with a light-grey background): from left to right whole numbers between zero and a certain maximum were produced by a random generator. If this series is sent as samples through a DA-converter, it is heard as *white noise*, in which all (audible) frequencies are physically equally loud (at a sampling-rate of 44100 Hz, these 9 boxes would take 20 ms to traverse). A Fourier analysis at the top of Γ41b shows a well-spread spectrum: nine DFTs are shown as black curves, each an analysis of a ninth of the random numbers (i.e. 98 each). The spectra, though different, all range from 0-20 kHz (each DFT box additionally shows in the background the other eight spectra for comparison in grey).

Noises also lend themselves to Fourier analysis: in Γ41b the amplitudes of a fictitious fundamental and its likewise fictitious partials were calculated (50 Hz and multiples thereof). For noise, one can interpolate between these partials – the lower the 'fundamental', the greater the density of the 'partials' in mid-range.

Observe the 98 random samples in the leftmost, slightly darker grey-filled box in Γ41b – stretched horizontally and compressed vertically, they can be seen again in the longish grey box just below: each successive pair of samples can be imagined as end points of a sine wave segment of maximum amplitude (−32768 to +32767 with 16 bits) – it was possible to calculate the variable frequency of this wave of invariable loudness from the sample rate (44100 Hz was assumed here) and sample-pair values – as shown in the graph below labelled 'SIS' (for 'sustained interval sequence') – allowing the white noise to be interpreted as a 'fleeting sine tone' with a random momentary frequency: the pitch histogramme at the bottom (marked 'Hgm' – the x-axis is calibrated in cents) shows the SIS-intervals as especially present in the zone roughly a tritone above and below 44100 Hz (0 on the x-axis), in seeming opposition to the DFTs above, to which they however are not necessarily related.

These considerations form the base of a sound wave system I developed in 2001 called ISIS ('Intra-Samplar Interpolation of Sinusoids'): in it, every pair of consecutive samples in any sound wave can be imagined as being connected by a sine segment of definite frequency and maximum amplitude. The formula for this frequency is

$$f = \mathfrak{R}(\arcsin(s_2)-\arcsin(s_1))/2\pi,$$

where f is the ISIS-frequency, $\mathfrak{R}$ the given sampling rate and s_1 and s_2 are two consecutive samples within a ±1 range. It was found meaningful to place f – if needed by adding or subtracting $\mathfrak{R}$ – within the frequency range $\mathfrak{R}(1\pm\frac{1}{2})$, i.e. sampling rate ± Nyquist limit ($\mathfrak{R}\pm\mathfrak{N}$). This version of ISIS involves *contigual phasing*, i.e. two neighbouring sine segments meet – reminiscent of splines – always with the same phase.

Γ41c shows an example of *ISIS Analysis* with 12 arbitrary samples (cf. Γ03g) – from each of the 11 successive sample-pairs one frequency was extracted by the above formula (e.g. 43195, 42685, 49460 Hz, shown with their connecting phases at the top of the diagramme – very high frequencies!); they can however be physically transposed or otherwise used as pitches, and then if need be, converted (back) into a sound wave by the following formula:

$$s_2 = \sin(\arcsin(s_1)+(2\pi f/\mathfrak{R})),$$

where f, $\mathfrak{R}$, s_1 and s_2 have the same meaning as before.

For Γ41d, the reverse method – *ISIS Synthesis* – was invoked: the examples shown here are 1) a repeating A_4, 2) an A_4–A_5 tremolo, 3) a random series of notes between A_4 and A_5 as well as 4) a stochastic notes series with probability maxima at A_4 and A_5 – histogrammes for the frequencies employed are shown above ('Hgm'). In synthesis I did not keep to the $\mathfrak{R}\pm\mathfrak{N}$ range. A DFT of the synthesised curves seen at the bottom right of each of the four diagrammes based on $73^1/_3$ Hz shows in 1) 440 Hz, 2) 660 Hz, to which the octave 440+880 Hz has 'fused'!, 3) 660 Hz with softer accompanying side-band frequencies and 4) a rich spectrum.

A noise can therefore quite plausibly be regarded as a sine wave of fleeting pitch, of which the frequency-histogramme has a special correlation, as yet unknown, to the spectrum.

Pitch is perceived in two main ways. In one of them, the frequencies establish audible intervallic links, from which the world of harmony derives (rational-intervallic hearing). More frequently, however, the sense of hearing is used to distinguish between high and low, bright and dark, to non-intervallically recognise language formants and intonation (pitch-spatial hearing). A whispered s.. sounds higher than a whispered sh without one needing to recognise the interval between these two noise-bands, nor to even notice it.

An experiment: two relatively narrow noise-bands with mid-frequencies 300 and 2000 Hz are presented to test-persons asked to place a third noise band half-way between the other two. Intervallic considerations suggest a mid-frequency of 775 Hz at the geometrical mean; however, the mid-frequency chosen will lie in fact at around 920 Hz, about a minor third higher. This can be explained by the fact that when rational-intervallic hearing is switched off, another – subjective – system of pitch-hearing comes into play which deviates from the physical cent-scale: see Γ42a.

A second experiment: the frequency of a sine wave is stochastically 'blurred' to a noise-band; if the sound pressure remains constant, the loudness does not change, even when the bandwidth of the noise continues to expand around a fixed mid-frequency – this should not be surprising, because with constant central pitch and constant sound pressure one would expect the loudness to remain constant. Suddenly, however, at a certain bandwidth the loudness begins to distinctly increase with further expansion – this *critical bandwidth* can be relatively clearly determined and possesses a specific size for every mid-frequency: see Γ42b.

Γ42c shows the course of the critical bandwidth in the region of 50 Hz to 18 kHz; given the mid-frequency of a band on the x-axis, its corresponding width can be read on the y-axis. If as in Γ42d, 100 Hz is the lower limit of a band, the upper limit of this band is around 200 Hz (follow the y-axis); the band starting at 200 Hz ends at 300 Hz, that starting at 300 Hz ends at 400 Hz. The band limits appear therefore to form an overtone series – 100, 200, 300, 400 Hz...; but instead of the expected 500 Hz, the next frequency limit is 510 Hz, followed by 630, 770, 920, 1080, 1270, 1480, 1720, 2000, 2320, 2700, 3150, 3700, 4400, 5300, 6400, 7700, 9500, 12000 and 15500 Hz – these values can be seen in Γ42d marked off on the y-axis by the numbers running diagonally from 1 to 24 marked 'bk' in the middle of the diagramme. Taken in cents, the bandwidths decrease rapidly at first, settling down to a bandwidth of 260 Ct around 2000 Hz, after which they expand again to just over 400 Ct, as can be seen again in Γ42c.

This series of band-limit frequencies represents in fact the above-mentioned subjective pitch scale; the unit for this is Bark (here abbreviated 'Bk') named after the acoustician Heinrich Barkhausen (1881-1956), who introduced the unit Phon. Eberhard Zwicker (1924-1990), who worked with the critical bandwidth from very early in its history, calls it a scale of 'toneness' (German *Tonheit*), analogous no doubt to the subjective 'loudness'-scale (German *Lautheit*); I prefer the term 'tone-height' – 'tone' is not an adjective!

Returning to the first experiment outlined in this chapter – the band-limits (the horizontal lines in Γ42d) show 920 Hz at the 8 Bk mark, subjectively half-way between 3 Bk (300 Hz) and 13 Bk (2000 Hz).

T43 tabulates in detail the relationship Bark→Hertz; it was calculated according to these formulæ by Ernst Terhardt (*1934) proposed in 1979 and Hartmut Traunmüller (*1944) proposed in 1990 as an algebraic approximation of the empirically derived values – the transition from the one formula to the other is at 219.5 Hz (= 2.16 Bk):

If $b \leqslant 2.16$:	$f = (4000/3)\tan(b/13.3)$	(Terhardt)
If $b \geqslant 2.16$:	$f = 1960(b+0.53)/(26.28-b)$	(Traunmüller)

If $b \geqslant 20.1$, Traunmüller improves his formula by the following operation:

$$b \leftarrow (b+4.422)/1.22.$$

The reverse formulæ:

If $f<=219.5$:	$b = 13.3\arctan(3f/4000)$	(Terhardt)
If $f>=219.5$:	$b = ((26.81f)/(1960+f))-0.53$	(Traunmüller)
Thereafter, if $b \geqslant 20.1$:	$b \leftarrow b+0.22(b-20.1)$	(Traunmüller correction)

(f is the frequency in Hz, b is the tone-height in Bk).

Γ42e shows the results of these formulæ and the results of four other approximations in graphic form – the grey vertical lines show an overtone series on 100 Hz, the dots showing the empirical values.

The critical bandwidth and the Bark scale appear in a series of other phenomena: e.g. the smallest discernable interval between two alternating sine waves is about 0.02 Bk in the whole audible range: about 35 Ct at 100 Hz, 6 Ct at 1000 Hz, 8 Ct at 10000 Hz. The quite coarse interval perception at low frequencies explains the well-known practice of tuning a double bass or timpani using higher natural harmonics on the open strings of the bass or near the brighter sounding rim of the timpani.

The scientists Reinier Plomp, Willem Levelt and others have also shown that the typical size of the interval between neighbouring notes in a chord in classical music has a definite correlation to the Bark scale: lower pitched-intervals, measured in semitones, are commonly larger than higher intervals. Furthermore, voice-leading in a variety of styles is generally characterized by more leaps in lower registers than in higher registers, in which steps are more frequent.

Legendary research by Plomp and Levelt additionally shows that the most dissonant interval between two simultaneous sine tones can be given in Bark (0.25), independent of the frequency. This interval is about 440 Ct at 100 Hz, 70 Ct at 1000 Hz and 100 Ct at 10000 Hz: see Chapter 30.

Composers who wish to achieve pitched music in which rational-intervallic hearing is irrelevant (e.g. extended glissandi or pointillistic textures) are here recommended the use of the Bark scale – the spread of the pitches then seems much more even.

If two sine waves sound simultaneously, their curves add up to a composite form like those shown grey-filled in Γ44: here are two frequencies in an octave ratio with a) equal initial phase, b) with a phase-shift of a eighth-period of the higher frequency. The total physical loudness in dB(SPL) depends on the RMS (the 'root-mean-square' or square root of the average square) of the curve. In this context, the phases that are actually irrelevant for the sound as perceived do play a role, as can be seen in the tiny RMS-difference between the 'a' and 'b' parts of the diagramme (1.00000 vs. 1.00013). A Fourier analysis of the two composite curves yields the original frequency, amplitude and phases as can be seen in the DFT-illustration to the right, shown as the 8^{th} and 16^{th} partials of an arbitrarily fixed fundamental, one period of which encompasses the whole of the sound wave shown; the phase is given by the slope of the spectral lines. In extreme cases, the addition of two sine waves with identical frequencies and amplitudes but with opposite phases gives a total loudness of zero, silence!: the curves cancel each other out. In this case, it is of course impossible for Fourier Analysis to reconstruct the original curves.

Γ44c shows two sine curves of opposite initial phase with a frequency ratio of 8:9 – the sum is a curve the loudness of which fluctuates: the frequency of the variations is the difference between the sine frequencies, in this case 9−8=1, i.e. an eighth of the lower component and a ninth of the higher one – one period of this is shown in the diagramme. This clearly illustrates the well-known phenomenon of 'beating' of frequencies close to each other. Again, Fourier analysis was able to plausibly reconstruct the situation (the opposite phase is mirrored in the direction of the spectral lines).

Γ44d shows two sine curves in the irrational and therefore non-periodic ratio of the Golden Section ($1{:}^{(1+\sqrt{5})}/_2$ or 1:1.618033989, close to 8:13); an examination of the RMS, period by period of the lower frequency, results in irregular variations. These can be interpreted as an irregularly changing loudness; the actual frequency and the sluggishness of the auditory perception of these frequencies determines the appropriate size of the time-window and thus the constancy of the supposed loudness. In fact, even a single sine tone could, with a large time-window encompassing the entire wave, be shown to have a steady loudness, and with a small enough window (approaching sample speed) as having a rapidly fluctuating loudness.

The subjectively perceived total loudness can be established as follows:

If the curve is aperiodic, so that the initial phases of the sine components are not significant, then the total physical or subjective loudness can be obtained by the respective addition of the individual sound intensities in W/m^2 or of the subjective loudnesses in Sones. Take e.g. a sawtooth tone with a fundamental frequency of 100 Hz, as shown in Γ44e. The phases of the partials are considered irrelevant here. By definition, the amplitudes of a sawtooth spectrum are proportional to the inverse of the partial number: these are A/n, where A is the amplitude of the fundamental. In this way, the intensities are correspondingly i/n^2, where i (proportional to A^2) is the intensity of the fundamental. Therefore the total intensity, with a practically infinite number of partials, converges to $(1+{}^1/_4+{}^1/_9+{}^1/_{16}\cdots)\times i = 1.6449i$. The individual intensities can now be determined using i/n^2 as 59% (=1/1.6449), 14.8%, 6.5%, 3.6% etc., of the absolute total intensity. If the total level is e.g. 90 dB(SPL), then according to T08 the total intensity is 1 mW/m^2 and thus that of the fundamental is 0.59 mW/m^2 and that of the following partials 0.148, 0.065, 0.036 etc. mW/m^2. To find the total subjective loudness, the spectrum, as a function of the amplitude in terms of the frequency of the partials, must be re-scaled into a physiologically relevant spectrum, viz. in Sones in terms of Hertz – or Bark, as in Γ44f. This happens primarily by using the Fletcher-Munson isophonic curves (Γ12a) and Hz→Bk conversion (Γ42e): if the frequency (Hz) and the loudness (as intensity or dB(SPL)-level) are known, then the loudness of each partial can be read in Phons and therefore also in Sones. Now the total loudness would be found through the summation of the Sone-values, were it not for the phenomenon of masking.

It has been experimentally shown that if two sine tones sound simultaneously, the lower tone has a damping effect on the loudness of the upper tone, especially if they have a subjective distance to each other of less than one Bark. In other words: if the two tones are close enough to each other and their loudness difference is large enough to the advantage of the lower tone, it could even happen that the higher tone is completely masked and not heard at all. Γ44g shows the sawtooth spectrum as a phenomenon of so-called *loudness density* in Sones per Bark as a function of the subjective pitch in Bark. The loudness of each tone is represented – instead of by vertical spectral lines – by a sloping curve relatively steeply ascending on the left and very slowly descending on the right: for a single sine tone, it is the area contained between this curve and the x-axis that corresponds to the loudness of the tone in Sones; the area unit is the product of the x-unit (Bark) and the y-unit (Sones/Bark), therefore Sones. In a spectrum of the loudness density (as in Γ44g), the maximum value of all curves together forms a common envelope curve, that contains between itself and the x-axis an area giving the total loudness. These overlapping single areas clarify the phenomenon of masking: each commonly shared loudness area only counts once: only the total area of all mutually masked partials together corresponds to the actual subjective loudness in Sones.

A sound wave entering the ear sets a number of things in motion. After passing through the *outer ear*, it reaches the *ear drum*, which transfers the motion to three tiny moveable bones or *ossicles* (*hammer*, *anvil* and *stirrup*) in the middle ear – see Γ45a. The last of these three is connected to the small membrane that covers the *oval window* to the *cochlea* in the inner ear, which in humans is a cone-shaped fluid-filled tube about 35 mm long and from 9 down to 3 mm in diameter coiled two-and-half times – see Γ45a$_1$ centre, for demonstration purposes also 'unrolled' to the right. Γ45a$_2$ shows the unrolled cochlea turned 90° around its central axis, then turned by 90° in the horizontal plane, showing it 'from the front' in a series of cross-sections in Γ45a$_3$. This tube is divided into three chambers – see the enlargement of a cross-section in Γ45a$_4$ – by two membranes, one of which (visible in all these representations of the cochlea), the *basilar membrane*, is a kind of 'hearing keyboard' with about 3500 hairs along it (see the Organ of Corti in the further enlargement in Γ45a$_5$). These hairs, if bent, electrically convey diverse pitch-impressions to the brain. A division of the basilar membrane into 24 equally long segments of 1.5 mm each (150 hairs) corresponds to the Bark scale.

The hairs are bent in the following manner: if an arriving sound wave shakes the oval window, the basilar membrane becomes wavy, similar to shaking out a towel – within milliseconds, a wave runs from the oval window to the tip of the cochlea, whereby the amplitude of the wave increases and then decreases. Because of the varying thickness of the membrane, the position of maximum amplitude depends on the frequency of the input signal: towards the oval window the membrane is thinner and harder and reacts more strongly to higher frequencies, vice versa at the tip (see again Γ45a$_2$). At the maximum, the hairs are bent by the independent movements of the tectorial membrane and the Organ of Corti (see again Γ45a$_5$) and transmit this condition via the auditory nerves to the brain. The smallest perceptible interval between two alternating sine tones is 0.02 Bk; this must therefore stimulate the basilar membrane at least 150 × 0.02 = 3 hairs apart for the two tones to be distinguishable.

The auditory nerves between the basilar membrane and the brain contain neurons, cells that in complete silence 'fire' irregularly up to 150 times a second, sending electrical impulses of a few microvolts and about 1 ms in duration to the brain. If the hairs of the basilar membrane are bent in the rhythm of sound wave periods, the neurons coordinate themselves temporally – the more sluggish of them do not fire at every sound period, but the total impression made is periodic. In this way, the brain receives double information: 1. spatially – through the auditory nerves connected to the point of stimulus on the basilar membrane, a rough pitch-analysis in Bark is created for pitch-spatial, non-intervallic hearing; 2. temporal – the rhythm of the electrical impulses of the neurons allows frequencies and therefore pitch to be rational-intervallically perceived.

In 1965, Reinier Plomp and Willem Levelt found that two simultaneously sounding sine tones seem the most dissonant (= 'the least pleasant') at a mutual distance of about 0.25 Bk; Γ45b shows graphs of the degree of dissonance as a function of the pitch distance in Bark. To be seen are Plomp and Levelt's original measurements

('P&L') in five octave-wide regions from 125 to 2000 Hz in light grey, the average of these in middle grey and a final stylised curve in bold light grey. The most dissonant interval, equal to 0.25 Bk, varies from 440 Ct at 100 Hz through 70 Ct at 1000 Hz up to 100 Ct at 10000 Hz. Also seen in the diagramme are algebraic approximations of the P&L curve, one by William Sethares (*1955) and one by Richard Parncutt (*1957), whose elegant solution is the best I know: $d=(4eb\times e^{-4b})^2$, where d is the dissonance and b is the pitch distance in Bark (e is the constant 2.71828...).

It could come as a surprise that the tritone, the major 7th (about 1 and 2 Bk respectively at 300 Hz) and other intervals known for their dissonance cause no dips in this curve. Here it is important to differentiate between the terms 'consonance'/'dissonance' and 'harmonicity'/'inharmonicity'): harmonicity ('intervallic clarity') stems from the numerical simplicity of the ratio between the two frequencies of an interval, whereby timbre is of hardly any significance; string music, for instance, can be re-instrumentated for winds and/or transposed to extreme registers without losing its harmonic meaning. By contrast, consonance refers to the 'smoothness' (dissonance to the 'roughness') of a sound. In the lowest piano octave, a perfect fourth – due to the larger interval size of the critical bandwidth there – can be shown to be more dissonant than a tritone, a whole-tone more dissonant than a semitone. The psychological phenomenon harmonicity originates in the brain's time-related perception of neuron firing: the physiological phenomenon consonance originates in distances on the ear's basilar membrane (0.25 Bk corresponds to about $^3/_8$ mm or 38 hairs.)

A pair of simultaneous sine tones a tritone or a major 7th apart seem in fact no more dissonant than a fifth or an octave: it is the friction of the smaller intervals between the complex tones' partials that makes the intervals dissonant. Γ45c shows the consonance behaviour of the tritone B_3–F_4 (250-354 Hz) based on the two notes' spectra; for each, six partials are shown including all frictional points – the total dissonance is 2.8 as also seen in Γ45d, which shows the dissonance sum as a continuous graph over two octaves (the curve for the lower octave is a re-creation of and therefore practically identical to the one published by Plomp and Levelt in 1965) – see for comparison the grey harmonicity curve at the bottom. I find the partials' subjective loudnesses also significant: if the dissonance values are multiplied by the loudnesses (in Sones) before they are added, the softer partials carry less weight as seen in Γ45e.

Γ45f shows an attempt to determine the dissonances of the twelve chromatic notes of an octave according to this method involving the subjective loudness. Sawtooth spectra were picked as timbre with 6, 12, and 24 partials. Three pitch registers were examined, the ascending octaves starting at 27.5 Hz, 220 Hz and 440 Hz respectively. It was found that for the octaves from 220 and 440 Hz, the classical standard dissonances minor 2nd, tritone and major 7th are, as expected, evidently more dissonant than the major 2nd, the perfect 4th and the minor 7th. However, the case is not so clear in the 27.5 Hz octave: major 2nd, minor 7th and perfect 4th (tiny dips here notwithstanding) are comparable in dissonance with the minor 2nd, tritone and major 7th respectively – try this out on a piano. In this low range, a semitone seems a bit like a mistuned unison, a tritone like a mistuned 5th, and a major 7th like a mistuned 8ve. Γ45f also shows (as grey-filled areas) the results of the Plomp and Levelt calculation method for six equally loud partials in the three pitch registers.

In the body of timbres there is a special subset which – in spite of the simplicity, small number and strict application of the laws governing it – is through its highly exhaustive employment of permitted possibilities richly varied and yet finely differentiated – the sounds of human language. Phonetics is the study of the laws of these timbres.

Physiologically considered, language sounds originate primarily through the spectral filtering of tones (e.g. vocal cord vibrations) and/or noises (e.g. the hiss of the flow of air) through the adjustable size of the mouth, nose and throat and thus by the variable resonance of these cavities. If air flows through the open mouth, *vowels* are formed (like a, e, i); if this cavity is narrowed through the tongue or lips, *fricatives* (e.g. s, z, v) are formed; if the mouth is closed by the tongue or lips with a simultaneously open nasal cavity *nasals* (e.g. m, n, ng) are formed, and if the nasal cavity is also closed and the internal air pressure is raised, *plosives* (e.g. p, b, d) are formed when the mouth is re-opened. Pressing the tip of the tongue onto the roof of the mouth with space on the sides of the tongue generates *laterals* (e.g. l as in loud).

In British English there are about 24 *consonants* (including so-called *semivowels* or *approximants*) as well as about 12 clearly differentiated *vowels* (the latter e.g. in keep this play there and the bard puns your own book soon or heat hit hate hair hat hurt heart hut haught hoe hood who'd or beat bit bait bet bat bird bard but bought boat book boot, along with at least 5 *diphthongs* (from the Greek *dís+phthóggos* = 'twice+voice', e.g. my boy sure fears gout – the vowels in e.g. play and own are also arguably diphthongs). These three dozen basic sounds are represented by 26 letters in different, in some cases ambiguous combinations – frequently the correct pronunciation can only be derived from the context: for example read as in read that book! sounds different in I've read it. Over 60 consonants as well as more than 20 vowels, occurring in all world languages, are recognised by the International Phonetic Association (IPA); in the International Phonetic Alphabet developed by the IPA, the twelve words given in the first example above as an illustration of the vowels could be written thus (oversimplified!): [kip ðɪs ple(i) ðɛ ænd ðə bɑd pʌnz jɔ o(u)n bʊk sun]. T46 shows most of the sounds acknowledged by the IPA in tabular form, subdivided according to biological-articulational aspects (see below).

However, back to the English vowels: The first four demonstrate long/short character ([kip/ðɪs], [ple/ðɛ]), the last four ([jɔ/on], [bʊk/sun]) short/long. If one pronounces the twelve vowels in succession – [iɪeɛæəɑʌɔoʊu] – one will notice that the tongue, at first high up in the front of the mouth, is lowered to the [ɑ], the tongue's highest point moving somewhat to the rear; thereafter it is raised again while being shifted even further back in the mouth. This movement causes the oral cavity in front of the tongue, as well as the rest of the space behind it, to change in size, having a direct effect on the sound production: the height of the tongue is responsible for one decisive variable, the forward-backward placement in the mouth of its highest point for another.

Γ47a shows a cross-section of the oral, nasal and pharyngeal cavities. If the lips are rounded, a fourth cavity between lips and teeth is created, which further influences the vowel formation: if an [i] (as in feel) is spoken with rounded lips (*labial*), it becomes an [y] (German ü, as in fühl), an illabial [e] becomes a labial [ø] (German ö). Inversely, a normal labial [u] becomes an [ɯ] with the lips unrounded, a sound that occurs for instance in languages as diverse as Portuguese, Thai and Korean, an [ɔ] with unrounded lips becomes an [ʌ], as in but. The [œ] can be found in the French [bœf] (bœuf). The [æ] occurs e.g. in [kæt] (cat). The dark British a in guard is given by [ɑ]; the [ɒ] represents the similar American English o in God. Γ47b shows tongue placement in the vowels [ieɛaɑɔou] – the phonetician Daniel Jones (1881-1967) termed these eight the *cardinal vowels*.

The tongue plays a similarly important role in producing consonants, which are classified according to three principles – where ('tongue position' in T46a – the Latin names for the different parts of the oral and pharyngeal cavities shown in Γ47a serve as a reference; 'retroflex' and 'glottal stop' require special explanation – see below), how (the 'way' the sound is produced, plosive, nasal, fricative etc.) and whether it is with vibrating vocal cords (voiced) or not (unvoiced). As to 'retroflex': the tongue is in this position if the tip is curled back and up toward the palate – the well-known 'Indian accent' owes its special timbre to this position; some retroflex sounds also occur in Norwegian and Swedish. The 'glottal stop' can be demonstrated by the difference in pronunciation of the word 'bottle', as it is spoken in standard British English (['bɔtl]) and in London dialects (['bɔʔl]); the ['] as phonetic symbol indicates a stressed following syllable (e.g. [fo'nɛtiks] for *phonetics*). Γ47c shows the position of the tongue in generating the unvoiced fricatives [fθsʃçχ] as in five three seven ships, the German ich (articulated in the front of the mouth, similar to a constricted h in huge!) and ach (articulated in the throat). Γ47d shows the tongue and lips in the production of the nasals [mnɲŋ] as in demure, tenor, (French) seigneur and singer.

T46a lists the vowels and T46b the consonants in the IPA phonetic alphabet according to their physiological place of origin.

The sounds of human speech, known as *phonemes*, are produced mainly by pressing air through the vocal tract in the throat and mouth – consonants produced in this way are termed *pulmonic* (from the Latin *pulmo* = 'lung'); moreover, other sound effects are possible such as e.g. clicking the tongue. The variable diameter of the vocal tract leads to the formation in it of single interconnected spaces, e.g. between the lips and the teeth, the teeth and the arch of the tongue, the space behind the arch of the tongue etc. The different sizes of these spaces amplify corresponding frequencies in the noise of the air stream and also where required in the sound of the vocal cords or the like.

Phonemes are invariably produced by shaping the mouth in a certain way, resulting each time in the same frequency conditions; thus each sound can be classified according to its typical spectrum. The frequency bands amplified by the spaces in the vocal tract and prominent within the spectrum are called *formants*. Γ48a_1 shows the spectrum of the vowel [i] based on the fundamentals C_3 (131 Hz) and $E♭_4$ (311 Hz). The overtone series of each spectrum has been marked with dots, above and below, respectively; in both spectra, the frequency regions at 250 Hz and 2500 Hz, common formants of [i], are louder than the surrounding frequencies and are marked with F1 and F2. Due to the independence of the vocal cords from the vocal tract, these absolute regions remain intact, even with other fundamentals. Γ48a_2 shows two spectra of [a] also based on C_3 and $E♭_4$ – the two formants are here F1=750 and F2=1500 Hz. Γ48b shows amplitude envelopes for the first two formants of the eight vowels [ieɛæ] and [uoɔɑ].

It has been established that the first two formants are sufficient for the recognition of all vowels, even though a number of higher ones can be measured. The first formant, formed by the posterior pharyngeal resonance chamber behind the arch of the tongue, ranges from ca. 250 Hz with a highly raised arch (as with [i] and [u]) causing an increase in the volume of the posterior chamber, up to 800 Hz when the tongue is laid flatter (as with [a]), with the anterior oral chamber and the now constricted posterior chamber similar in volume. This first formant depends therefore mainly on the height of the arch of the tongue, a vertical parameter. The smaller the posterior chamber volume, the higher the 1st formant and vice versa. The second formant, formed by the anterior chamber between the arch of the tongue and the teeth, ranges from 800 Hz with the tongue arched at the back causing a larger anterior volume (as with [u]), up to 2500 Hz and higher with the arch in the front causing a smaller anterior volume (as with [i]). Thus this second formant results from the arch of the tongue moving forwards and backwards on the horizontal axis running through the vocal tract.

This two-dimensionality can be used to represent all vowels on one Cartesian plane, as can be seen in Γ48c: the two formants are assigned to the x- and y-coordinates. Measurements taken from six sources are shown here, respectively grouped for each vowel in a grey field bordered by the sources. It is remarkable that the fields, despite the diversity of the sources, practically never overlap. Every field is marked near at the centre by the symbol for the vowel. Notice, too, the four slanting straight lines in grey connecting points in or very near the fields [ieɛæ], [yøœɑ], [ɯɣʌɶ] and [uoɔɒ] – the sixteen vowels are located on a grid of tritones (C_4, $F\sharp_4$, C_5 and $F\sharp_5$) for F1 and whole tones (F_5, G_5, A_5, B_5, $C\sharp_6$ etc.) for F2 – see also Γ48e. Sound syntheses of the sixteen vowels based on these formant positions have proved to be convincing.

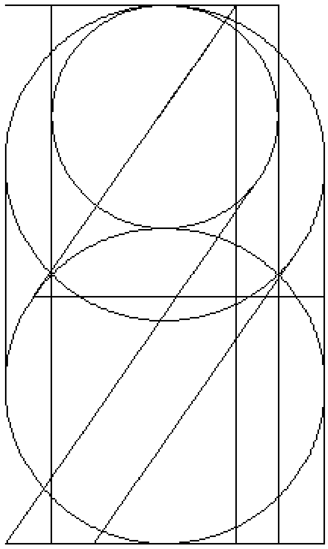
Γ48d shows, based on their positions on the grid in Γ48c, the formants of the vowel cycle [ieɛæaɑɔouɯyi]. This trapezoid form displays the outer limits of human-biological vowel formation; all humanly produceable vowels can be located within the trapezium. The F1-mirrored formant pairs shown (with invented symbols) at left are electrotechnically realisable and sound intriguing, but transcend the possibilities of the human mouth.

Vowels (from the Latin *vox* = ‘voice’, among other things) can sound alone; on the other hand the ‘consonants’ (from the Latin *con+sonare* = ‘together-sounding’ since they usually accompany vowels) form a diffuse group, from nasals (e.g. [nɳŋɴɱm]), laterals ([lɭʎʟ]) and noisy fricatives ([fvθðszʃʒçʝ] etc.), all of which like vowels can actually also sound alone, to truly *con*-sonant plosive sounds ([pbtdkg] etc.). While no special alphabetic or phonetic symbols have been assigned by the IPA to the nasals in their unvoiced form, the distinction voiced-unvoiced is strongly made in all other cases. The unvoiced [ɬ], e.g. in Welsh (written in that language as ll, a double-L), is both a lateral and a fricative – try to get a Welsh person to say ‘Llanelli’, the name of a town in Southern Wales. Γ48f schematically shows the spectra of six fricatives, [xfθʃçs] as Gaussian curves. Here it is not formants but more or less broad noise bands which characterise the spectra, for which reason they have been calibrated in Bark and Phon. Whereas the [f] and [θ] spectra are very similar (and their sounds are often inadvertently confused), the [xfθʃçs] sequence clearly reflects a generally rising frequency. This sequence is shown in Γ48g in the form of a sonagramme (from the Latin *sonus* = ‘a sound’ and the Greek *grámma* = ‘something written’); the x-axis represents time, the y-axis (here subjective) pitch and loudness is shown by means of grey-scale values. The use of sonagrammes for the visual demonstration of the spectral analysis of speech, graphically displaying musically relevant parameters such as pitch, time and loudness, is very widespread.

Part II: Figures

Preface to Part II

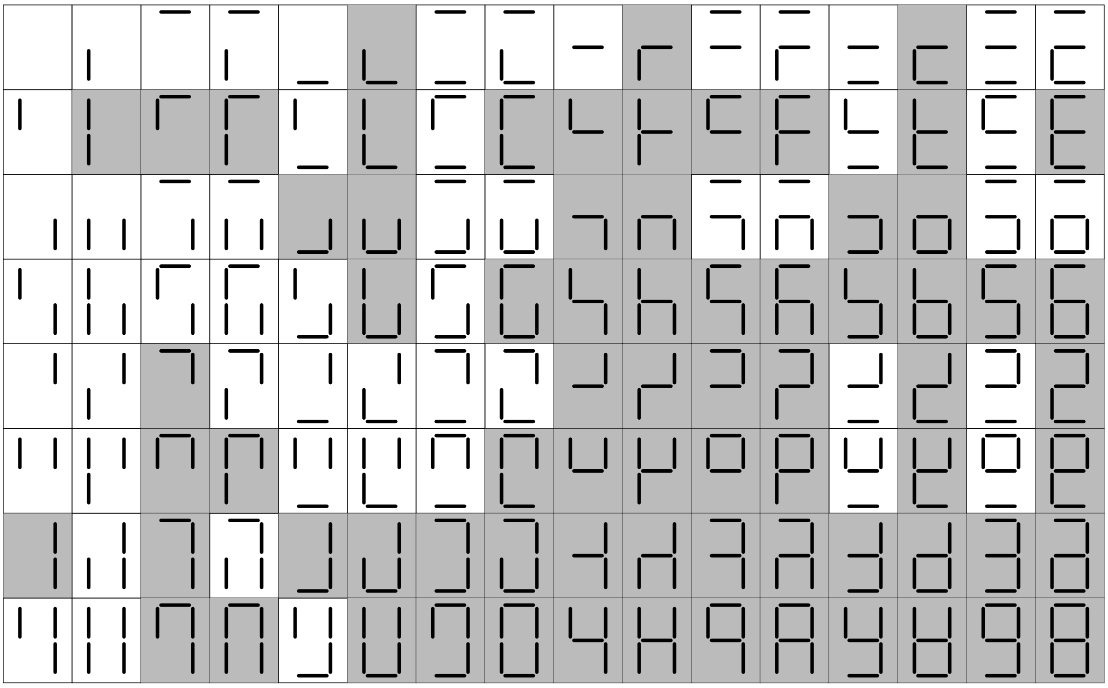
All formulæ and graphs were generated in Postscript format (thanks to a suggestion by Björn Erlach) by computer programs written by myself. I also designed some True Type fonts for this book, such as ‘Musiquantik’, containing – among others – the sixteen characters 0 1 2 3 4 5 6 7 8 9 A B C D E F, seen as hexadecimal digits on pages 2ff: these were all extracted from the lattice on the right.



On Musiquantics Part II

Contents

Г01 – 128 combinations of 7 dashes



T02 – Conversion of number systems

$0_d = 00000000_b = 000_o = 00_h$
$1_d = 00000001_b = 001_o = 01_h$
$2_d = 00000010_b = 002_o = 02_h$
$3_d = 00000011_b = 003_o = 03_h$
$4_d = 00000100_b = 004_o = 04_h$
$5_d = 00000101_b = 005_o = 05_h$
$6_d = 00000110_b = 006_o = 06_h$
$7_d = 00000111_b = 007_o = 07_h$
$8_d = 00001000_b = 010_o = 08_h$
$9_d = 00001001_b = 011_o = 09_h$
$10_d = 00001010_b = 012_o = 0A_h$
$11_d = 00001011_b = 013_o = 0B_h$
$12_d = 00001100_b = 014_o = 0C_h$
$13_d = 00001101_b = 015_o = 0D_h$
$14_d = 00001110_b = 016_o = 0E_h$
$15_d = 00001111_b = 017_o = 0F_h$
$16_d = 00010000_b = 020_o = 10_h$
$17_d = 00010001_b = 021_o = 11_h$
$18_d = 00010010_b = 022_o = 12_h$
$19_d = 00010011_b = 023_o = 13_h$
$20_d = 00010100_b = 024_o = 14_h$
$21_d = 00010101_b = 025_o = 15_h$
$22_d = 00010110_b = 026_o = 16_h$
$23_d = 00010111_b = 027_o = 17_h$
$24_d = 00011000_b = 030_o = 18_h$
$25_d = 00011001_b = 031_o = 19_h$
$26_d = 00011010_b = 032_o = 1A_h$
$27_d = 00011011_b = 033_o = 1B_h$
$28_d = 00011100_b = 034_o = 1C_h$
$29_d = 00011101_b = 035_o = 1D_h$
30_d = 00011110b = 036o = $1E_h$

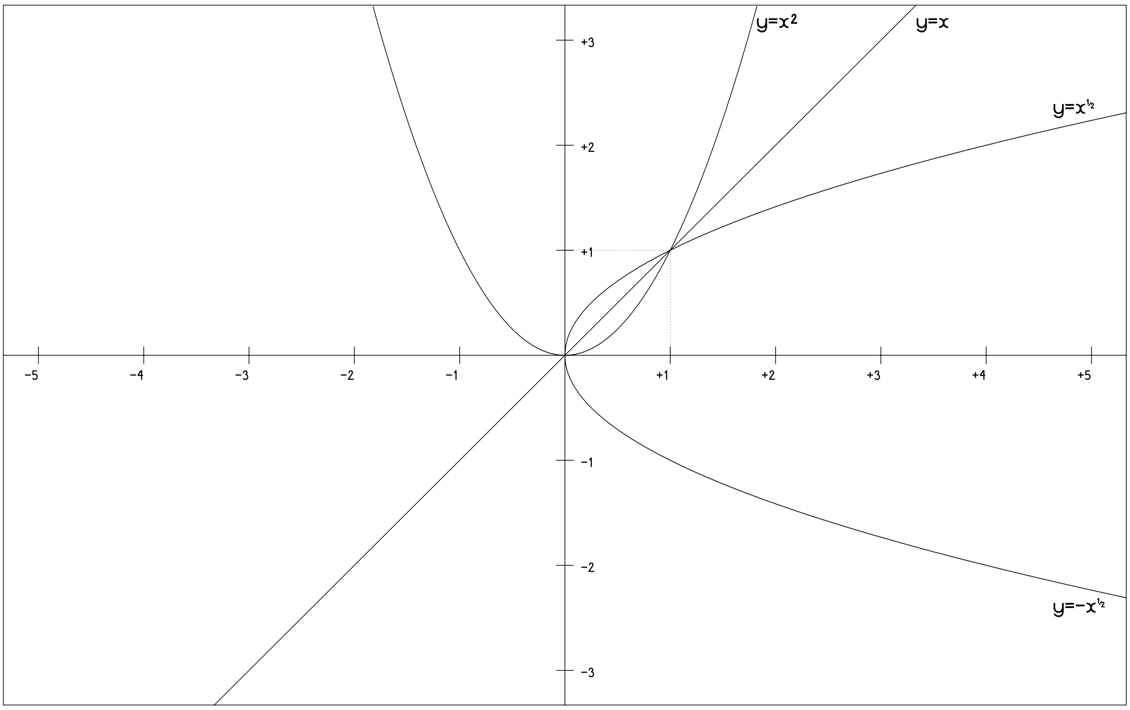
$31_d = 00011111_b = 037_o = 1F_h$
$32_d = 00100000_b = 040_o = 20_h$
$33_d = 00100001_b = 041_o = 21_h$
$34_d = 00100010_b = 042_o = 22_h$
$35_d = 00100011_b = 043_o = 23_h$
$36_d = 00100100_b = 044_o = 24_h$
$37_d = 00100101_b = 045_o = 25_h$
$38_d = 00100110_b = 046_o = 26_h$
$39_d = 00100111_b = 047_o = 27_h$
$40_d = 00101000_b = 050_o = 28_h$
$41_d = 00101001_b = 051_o = 29_h$
$42_d = 00101010_b = 052_o = 2A_h$
$43_d = 00101011_b = 053_o = 2B_h$
$44_d = 00101100_b = 054_o = 2C_h$
$45_d = 00101101_b = 055_o = 2D_h$
$46_d = 00101110_b = 056_o = 2E_h$
$47_d = 00101111_b = 057_o = 2F_h$
$48_d = 00110000_b = 060_o = 30_h$
$49_d = 00110001_b = 061_o = 31_h$
$50_d = 00110010_b = 062_o = 32_h$
$51_d = 00110011_b = 063_o = 33_h$
$52_d = 00110100_b = 064_o = 34_h$
$53_d = 00110101_b = 065_o = 35_h$
$54_d = 00110110_b = 066_o = 36_h$
$55_d = 00110111_b = 067_o = 37_h$
$56_d = 00111000_b = 070_o = 38_h$
$57_d = 00111001_b = 071_o = 39_h$
$58_d = 00111010_b = 072_o = 3A_h$
$59_d = 00111011_b = 073_o = 3B_h$
$60_d = 00111100_b = 074_o = 3C_h$
$61_d = 00111101_b = 075_o = 3D_h$

$62_d = 00111110_b = 076_o = 3E_h$
$63_d = 00111111_b = 077_o = 3F_h$
$64_d = 01000000_b = 100_o = 40_h$
$65_d = 01000001_b = 101_o = 41_h$
$66_d = 01000010_b = 102_o = 42_h$
$67_d = 01000011_b = 103_o = 43_h$
$68_d = 01000100_b = 104_o = 44_h$
$69_d = 01000101_b = 105_o = 45_h$
$70_d = 01000110_b = 106_o = 46_h$
$71_d = 01000111_b = 107_o = 47_h$
$72_d = 01001000_b = 110_o = 48_h$
$73_d = 01001001_b = 111_o = 49_h$
$74_d = 01001010_b = 112_o = 4A_h$
$75_d = 01001011_b = 113_o = 4B_h$
$76_d = 01001100_b = 114_o = 4C_h$
$77_d = 01001101_b = 115_o = 4D_h$
$78_d = 01001110_b = 116_o = 4E_h$
$79_d = 01001111_b = 117_o = 4F_h$
$80_d = 01010000_b = 120_o = 50_h$
$81_d = 01010001_b = 121_o = 51_h$
$82_d = 01010010_b = 122_o = 52_h$
$83_d = 01010011_b = 123_o = 53_h$
$84_d = 01010100_b = 124_o = 54_h$
$85_d = 01010101_b = 125_o = 55_h$
$86_d = 01010110_b = 126_o = 56_h$
$87_d = 01010111_b = 127_o = 57_h$
$88_d = 01011000_b = 130_o = 58_h$
$89_d = 01011001_b = 131_o = 59_h$
$90_d = 01011010_b = 132_o = 5A_h$
$91_d = 01011011_b = 133_o = 5B_h$
$92_d = 01011100_b = 134_o = 5C_h$

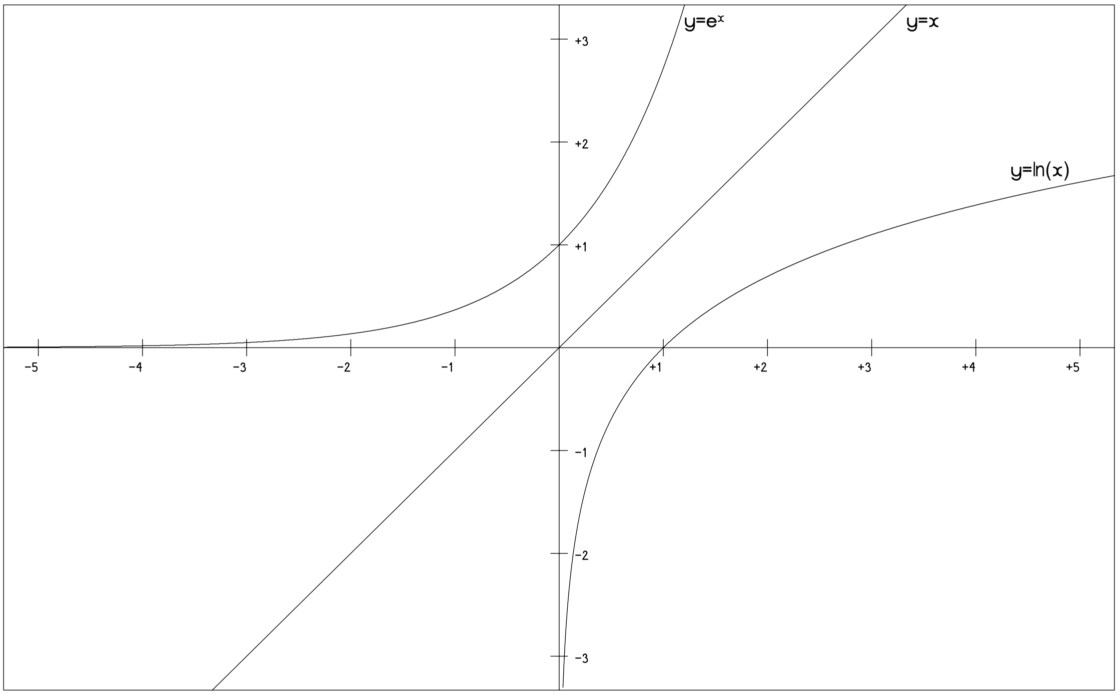
$93_d = 01011101_b = 135_o = 5D_h$
$94_d = 01011110_b = 136_o = 5E_h$
$95_d = 01011111_b = 137_o = 5F_h$
$96_d = 01100000_b = 140_o = 60_h$
$97_d = 01100001_b = 141_o = 61_h$
$98_d = 01100010_b = 142_o = 62_h$
$99_d = 01100011_b = 143_o = 63_h$
$100_d = 01100100_b = 144_o = 64_h$
$101_d = 01100101_b = 145_o = 65_h$
$102_d = 01100110_b = 146_o = 66_h$
$103_d = 01100111_b = 147_o = 67_h$
$104_d = 01101000_b = 150_o = 68_h$
$105_d = 01101001_b = 151_o = 69_h$
$106_d = 01101010_b = 152_o = 6A_h$
$107_d = 01101011_b = 153_o = 6B_h$
$108_d = 01101100_b = 154_o = 6C_h$
$109_d = 01101101_b = 155_o = 6D_h$
$110_d = 01101110_b = 156_o = 6E_h$
$111_d = 01101111_b = 157_o = 6F_h$
$112_d = 01110000_b = 160_o = 70_h$
$113_d = 01110001_b = 161_o = 71_h$
$114_d = 01110010_b = 162_o = 72_h$
$115_d = 01110011_b = 163_o = 73_h$
$116_d = 01110100_b = 164_o = 74_h$
$117_d = 01110101_b = 165_o = 75_h$
$118_d = 01110110_b = 166_o = 76_h$
$119_d = 01110111_b = 167_o = 77_h$
$120_d = 01111000_b = 170_o = 78_h$
$121_d = 01111001_b = 171_o = 79_h$
$122_d = 01111010_b = 172_o = 7A_h$
$123_d = 01111011_b = 173_o = 7B_h$
$124_d = 01111100_b = 174_o = 7C_h$
$125_d = 01111101_b = 175_o = 7D_h$
$126_d = 01111110_b = 176_o = 7E_h$
$127_d = 01111111_b = 177_o = 7F_h$
$128_d = 10000000_b = 200_o = 80_h$
$129_d = 10000001_b = 201_o = 81_h$
$130_d = 10000010_b = 202_o = 82_h$
$131_d = 10000011_b = 203_o = 83_h$
$132_d = 10000100_b = 204_o = 84_h$
$133_d = 10000101_b = 205_o = 85_h$
$134_d = 10000110_b = 206_o = 86_h$
$135_d = 10000111_b = 207_o = 87_h$
$136_d = 10001000_b = 210_o = 88_h$
$137_d = 10001001_b = 211_o = 89_h$
$138_d = 10001010_b = 212_o = 8A_h$
$139_d = 10001011_b = 213_o = 8B_h$
$140_d = 10001100_b = 214_o = 8C_h$
$141_d = 10001101_b = 215_o = 8D_h$
$142_d = 10001110_b = 216_o = 8E_h$
$143_d = 10001111_b = 217_o = 8F_h$
$144_d = 10010000_b = 220_o = 90_h$
$145_d = 10010001_b = 221_o = 91_h$
$146_d = 10010010_b = 222_o = 92_h$
$147_d = 10010011_b = 223_o = 93_h$
$148_d = 10010100_b = 224_o = 94_h$
$149_d = 10010101_b = 225_o = 95_h$
$150_d = 10010110_b = 226_o = 96_h$
$151_d = 10010111_b = 227_o = 97_h$
$152_d = 10011000_b = 230_o = 98_h$
$153_d = 10011001_b = 231_o = 99_h$
$154_d = 10011010_b = 232_o = 9A_h$
$155_d = 10011011_b = 233_o = 9B_h$
$156_d = 10011100_b = 234_o = 9C_h$
$157_d = 10011101_b = 235_o = 9D_h$
$158_d = 10011110_b = 236_o = 9E_h$
$159_d = 10011111_b = 237_o = 9F_h$
$160_d = 10100000_b = 240_o = A0_h$
$161_d = 10100001_b = 241_o = A1_h$
$162_d = 10100010_b = 242_o = A2_h$
$163_d = 10100011_b = 243_o = A3_h$
$164_d = 10100100_b = 244_o = A4_h$
$165_d = 10100101_b = 245_o = A5_h$
$166_d = 10100110_b = 246_o = A6_h$
$167_d = 10100111_b = 247_o = A7_h$
$168_d = 10101000_b = 250_o = A8_h$
$169_d = 10101001_b = 251_o = A9_h$
$170_d = 10101010_b = 252_o = AA_h$
$171_d = 10101011_b = 253_o = AB_h$
$172_d = 10101100_b = 254_o = AC_h$
$173_d = 10101101_b = 255_o = AD_h$
$174_d = 10101110_b = 256_o = AE_h$
$175_d = 10101111_b = 257_o = AF_h$
$176_d = 10110000_b = 260_o = B0_h$
$177_d = 10110001_b = 261_o = B1_h$
$178_d = 10110010_b = 262_o = B2_h$
$179_d = 10110011_b = 263_o = B3_h$
$180_d = 10110100_b = 264_o = B4_h$
$181_d = 10110101_b = 265_o = B5_h$
$182_d = 10110110_b = 266_o = B6_h$
$183_d = 10110111_b = 267_o = B7_h$
$184_d = 10111000_b = 270_o = B8_h$
$185_d = 10111001_b = 271_o = B9_h$
$186_d = 10111010_b = 272_o = BA_h$
$187_d = 10111011_b = 273_o = BB_h$
$188_d = 10111100_b = 274_o = BC_h$
$189_d = 10111101_b = 275_o = BD_h$
$190_d = 10111110_b = 276_o = BE_h$
$191_d = 10111111_b = 277_o = BF_h$
$192_d = 11000000_b = 300_o = C0_h$
$193_d = 11000001_b = 301_o = C1_h$
$194_d = 11000010_b = 302_o = C2_h$
$195_d = 11000011_b = 303_o = C3_h$
$196_d = 11000100_b = 304_o = C4_h$
$197_d = 11000101_b = 305_o = C5_h$
$198_d = 11000110_b = 306_o = C6_h$
$199_d = 11000111_b = 307_o = C7_h$
$200_d = 11001000_b = 310_o = C8_h$
$201_d = 11001001_b = 311_o = C9_h$
$202_d = 11001010_b = 312_o = CA_h$
$203_d = 11001011_b = 313_o = CB_h$
$204_d = 11001100_b = 314_o = CC_h$
$205_d = 11001101_b = 315_o = CD_h$
$206_d = 11001110_b = 316_o = CE_h$
$207_d = 11001111_b = 317_o = CF_h$
$208_d = 11010000_b = 320_o = D0_h$
$209_d = 11010001_b = 321_o = D1_h$
$210_d = 11010010_b = 322_o = D2_h$
$211_d = 11010011_b = 323_o = D3_h$
$212_d = 11010100_b = 324_o = D4_h$
$213_d = 11010101_b = 325_o = D5_h$
$214_d = 11010110_b = 326_o = D6_h$
$215_d = 11010111_b = 327_o = D7_h$
$216_d = 11011000_b = 330_o = D8_h$
$217_d = 11011001_b = 331_o = D9_h$
$218_d = 11011010_b = 332_o = DA_h$
$219_d = 11011011_b = 333_o = DB_h$
$220_d = 11011100_b = 334_o = DC_h$
$221_d = 11011101_b = 335_o = DD_h$
$222_d = 11011110_b = 336_o = DE_h$
$223_d = 11011111_b = 337_o = DF_h$
$224_d = 11100000_b = 340_o = E0_h$
$225_d = 11100001_b = 341_o = E1_h$
$226_d = 11100010_b = 342_o = E2_h$
$227_d = 11100011_b = 343_o = E3_h$
$228_d = 11100100_b = 344_o = E4_h$
$229_d = 11100101_b = 345_o = E5_h$
$230_d = 11100110_b = 346_o = E6_h$
$231_d = 11100111_b = 347_o = E7_h$
$232_d = 11101000_b = 350_o = E8_h$
$233_d = 11101001_b = 351_o = E9_h$
$234_d = 11101010_b = 352_o = EA_h$
$235_d = 11101011_b = 353_o = EB_h$
$236_d = 11101100_b = 354_o = EC_h$
$237_d = 11101101_b = 355_o = ED_h$
$238_d = 11101110_b = 356_o = EE_h$
$239_d = 11101111_b = 357_o = EF_h$
$240_d = 11110000_b = 360_o = F0_h$
$241_d = 11110001_b = 361_o = F1_h$
$242_d = 11110010_b = 362_o = F2_h$
$243_d = 11110011_b = 363_o = F3_h$
$244_d = 11110100_b = 364_o = F4_h$
$245_d = 11110101_b = 365_o = F5_h$
$246_d = 11110110_b = 366_o = F6_h$
$247_d = 11110111_b = 367_o = F7_h$
$248_d = 11111000_b = 370_o = F8_h$
$249_d = 11111001_b = 371_o = F9_h$
$250_d = 11111010_b = 372_o = FA_h$
$251_d = 11111011_b = 373_o = FB_h$
$252_d = 11111100_b = 374_o = FC_h$
$253_d = 11111101_b = 375_o = FD_h$
$254_d = 11111110_b = 376_o = FE_h$
$255_d = 11111111_b = 377_o = FF_h$

Γ03 – Two-dimensional curves

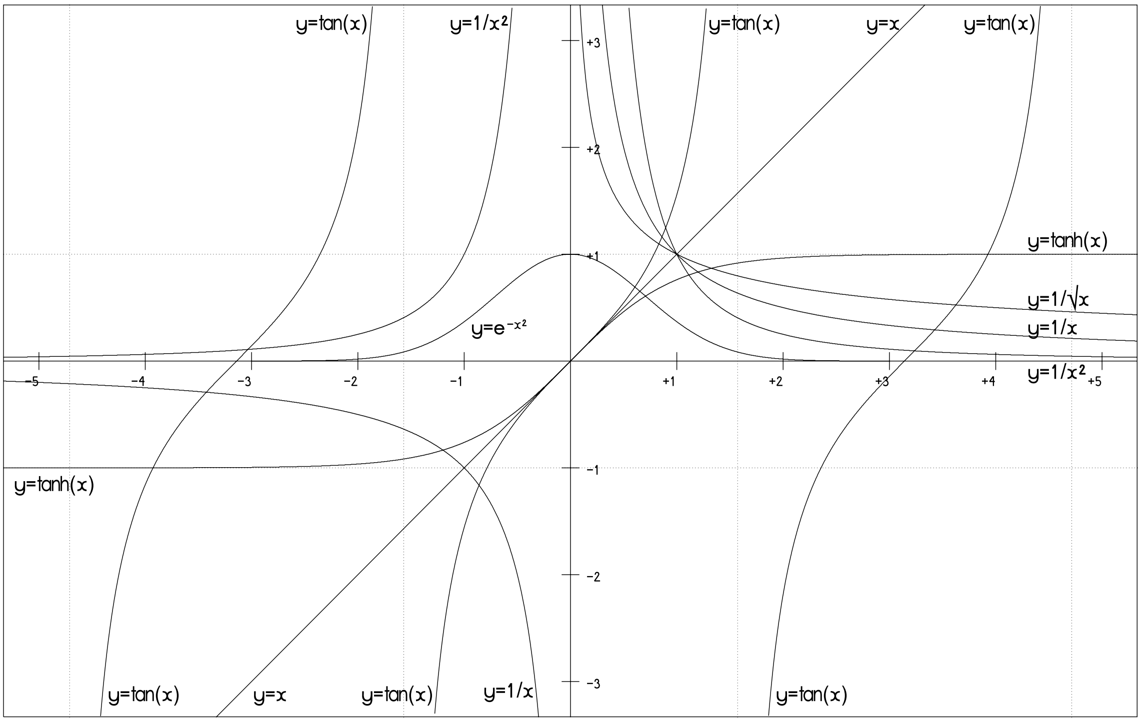
a) Parabolas with y=x straight line



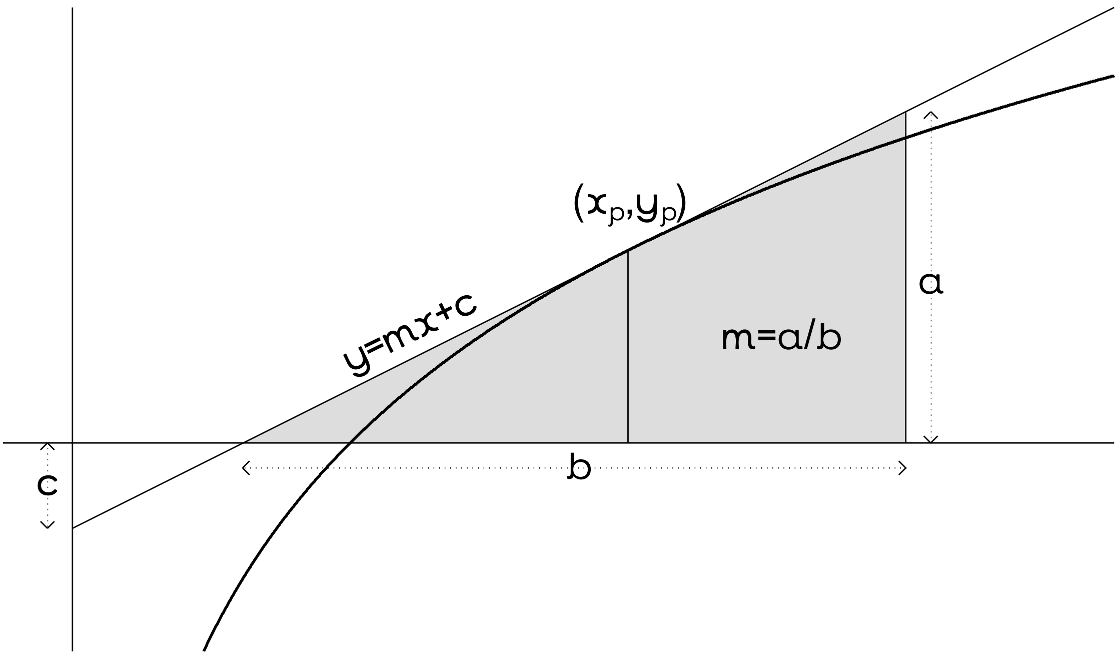
b) Exponential and logarithmic curves with y=x straight line



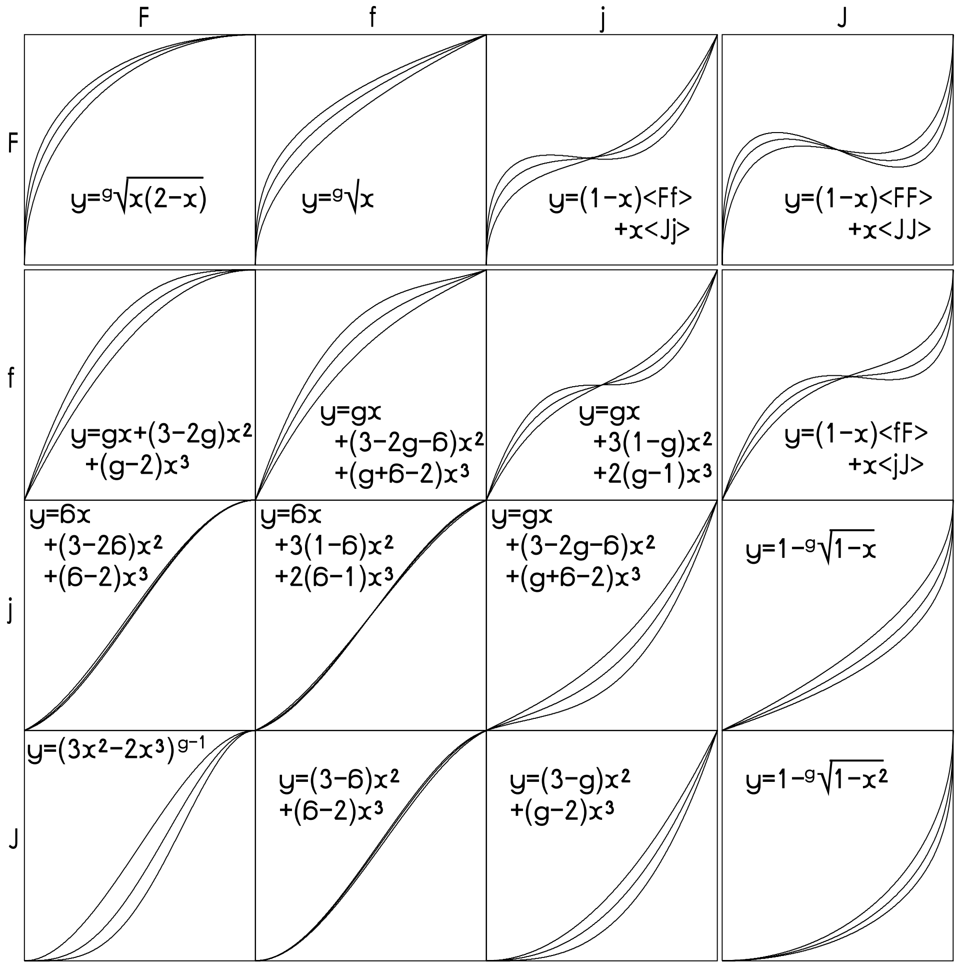
c) Vertically and horizontally asymptotic curves with $y=x$ straight line



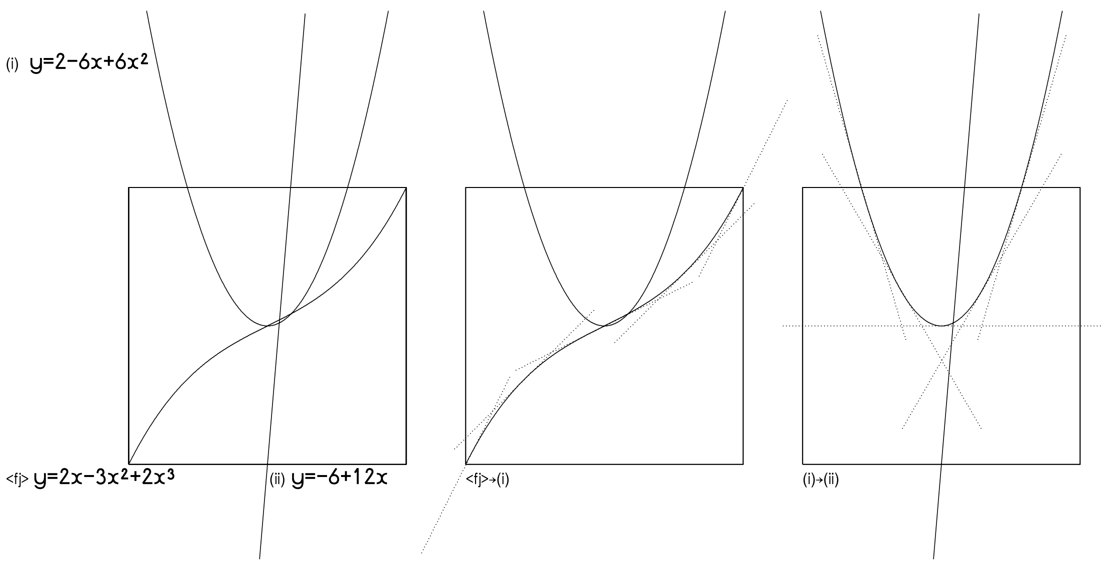
d) Example of a tangent $y=mx+c$ of gradient m at a point (x,y) on a curve (bold)



e) 16 non-tangential as well as uni- and bilaterally axis-parallel tangential curves ($0 \leq x \leq 1$, $0 \leq y \leq 1$) – in each case, the (positive) basic gradient $g=2$, $2\frac{1}{2}$ and 3; $\delta = 1/g$

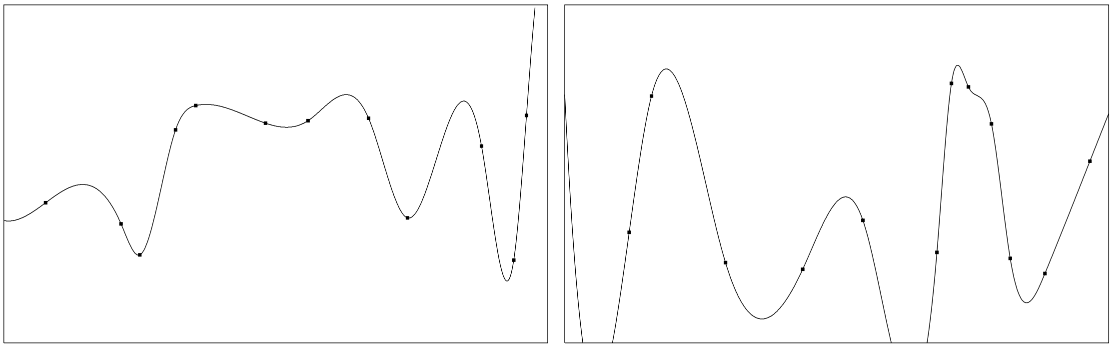


f) The above fj-curve (for $g=2$) with (i) 1st and (ii) 2nd derivatives und tangents



...F03...

g) Two splines each connecting a random series of points



F04 – Formulæ for the calculation of the acceleration of a series of beats based on initial and final tempo, the total duration and the number of beats

$$s = SQ^{\frac{t}{T}}$$

$$n = \frac{TS(Q^{\frac{t}{T}}-1)}{\ln(Q)}$$

$$T = \frac{N\ln(Q)}{(S'-S)}$$

$$t = \frac{T\ln(s/S)}{\ln(Q)}$$

$$= \frac{T\ln((n\ln(Q)/TS)+1)}{\ln(Q)}$$

where S = initial tempo

S' = final tempo

$Q = S'/S$

T = total time

t = current time [i.e. time at tempo s or at beat n - see below]

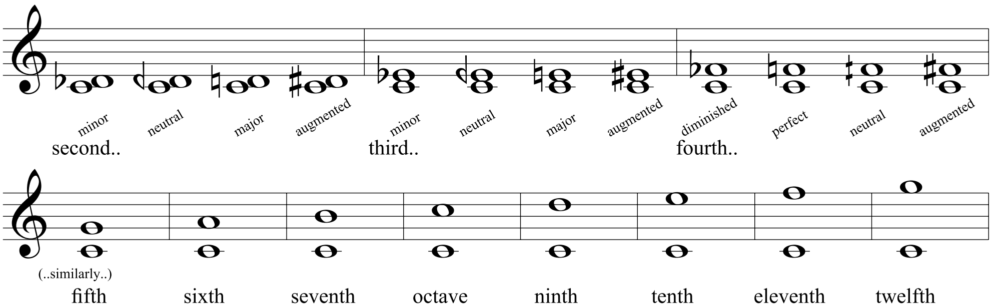
s = tempo at time t

n = beat at time t

N = total number of beats

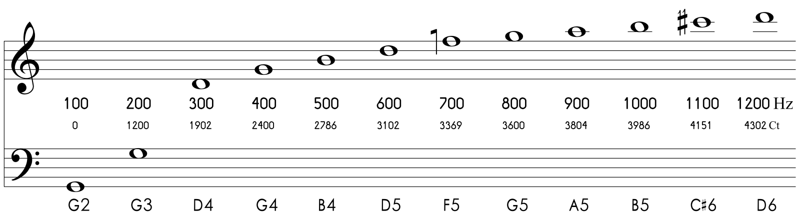
$\ln(x)$ = natural logarithm of x [see chapter 4]

Γ05 – A selection of commonly notatable intervals

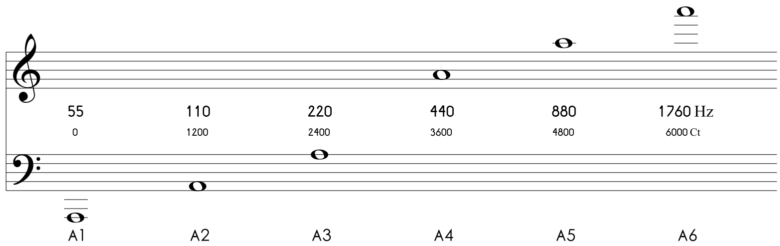


Γ06 – Examples of series of frequencies in Hertz

a) Arithmetic Series, e.g. the overtone row (note names approximate)

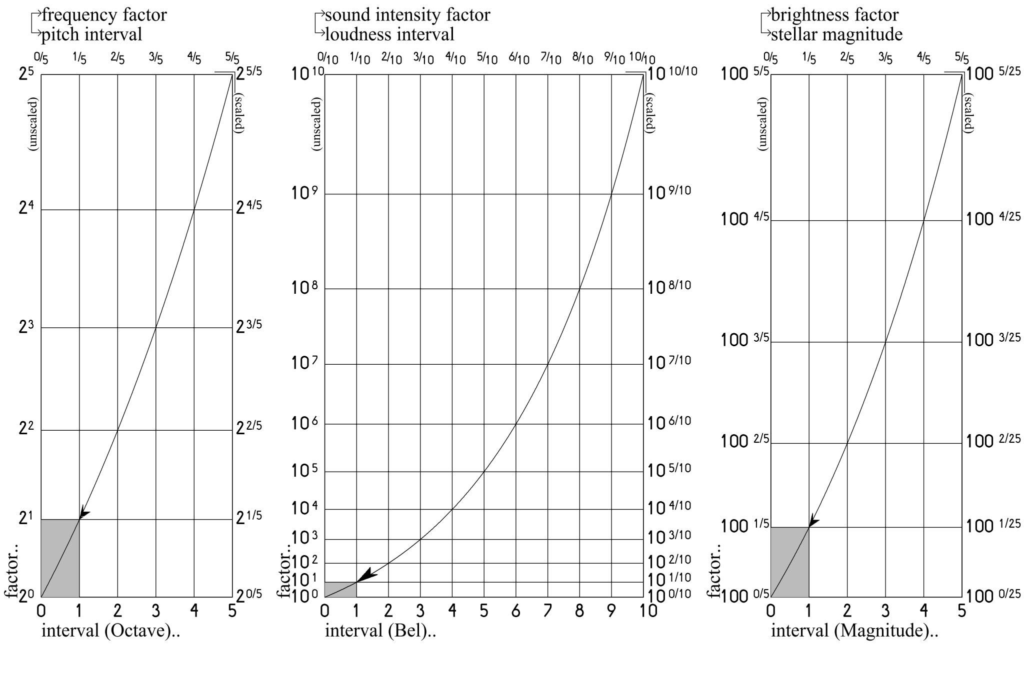


b) Geometric Series, e.g. octaves



Γ07 – The relationship between geometric factor and arithmetic interval in

a) pitch, b) loudness and c) stellar magnitude



T08 – Measurements of pitch and loudness

a) Number, note names, frequencies in 100-Ct-steps (A4 = 440 Hz)
└from 0-127 identical to the MIDI-scale (see Chapter 18)

No.	name,8^{ve}	frequency	No.	name,8^{ve}	frequency	No.	name,8^{ve}	frequency
0	C_{-1}	8.17580 Hz	50	D_3	146.832 Hz	100	E_7	2637.02 Hz
1	$C\sharp_{-1}$	8.66196 Hz	51	$E\flat_3$	155.563 Hz	101	F_7	2793.83 Hz
2	D_{-1}	9.17702 Hz	52	E_3	164.814 Hz	102	$F\sharp_7$	2959.96 Hz
3	$E\flat_{-1}$	9.72272 Hz	53	F_3	174.614 Hz	103	G_7	3135.96 Hz
4	E_{-1}	10.3009 Hz	54	$F\sharp_3$	184.997 Hz	104	$A\flat_7$	3322.44 Hz
5	F_{-1}	10.9134 Hz	55	G_3	195.998 Hz	105	A_7	3520.00 Hz
6	$F\sharp_{-1}$	11.5623 Hz	56	$A\flat_3$	207.652 Hz	106	$B\flat_7$	3729.31 Hz
7	G_{-1}	12.2499 Hz	57	A_3	220.000 Hz	107	B_7	3951.07 Hz
8	$A\flat_{-1}$	12.9783 Hz	58	$B\flat_3$	233.082 Hz	108	C_8	4186.01 Hz
9	A_{-1}	13.7500 Hz	59	B_3	246.942 Hz	109	$C\sharp_8$	4434.92 Hz
10	$B\flat_{-1}$	14.5676 Hz	60	C_4	261.626 Hz	110	D_8	4698.64 Hz
11	B_{-1}	15.4339 Hz	61	$C\sharp_4$	277.183 Hz	111	$E\flat_8$	4978.03 Hz
12	C_0	16.3516 Hz	62	D_4	293.665 Hz	112	E_8	5274.04 Hz
13	$C\sharp_0$	17.3239 Hz	63	$E\flat_4$	311.127 Hz	113	F_8	5587.65 Hz
14	D_0	18.3540 Hz	64	E_4	329.628 Hz	114	$F\sharp_8$	5919.91 Hz
15	$E\flat_0$	19.4454 Hz	65	F_4	349.228 Hz	115	G_8	6271.93 Hz
16	E_0	20.6017 Hz	66	$F\sharp_4$	369.994 Hz	116	$A\flat_8$	6644.88 Hz
17	F_0	21.8268 Hz	67	G_4	391.995 Hz	117	A_8	7040.00 Hz
18	$F\sharp_0$	23.1247 Hz	68	$A\flat_4$	415.305 Hz	118	$B\flat_8$	7458.62 Hz
19	G_0	24.4997 Hz	69	A_4	440.000 Hz	119	B_8	7902.13 Hz
20	$A\flat_0$	25.9565 Hz	70	$B\flat_4$	466.164 Hz	120	C_9	8372.02 Hz
21	A_0	27.5000 Hz	71	B_4	493.883 Hz	121	$C\sharp_9$	8869.84 Hz
22	$B\flat_0$	29.1352 Hz	72	C_5	523.251 Hz	122	D_9	9397.27 Hz
23	B_0	30.8677 Hz	73	$C\sharp_5$	554.365 Hz	123	$E\flat_9$	9956.06 Hz
24	C_1	32.7032 Hz	74	D_5	587.330 Hz	124	E_9	10548.1 Hz
25	$C\sharp_1$	34.6478 Hz	75	$E\flat_5$	622.254 Hz	125	F_9	11175.3 Hz
26	D_1	36.7081 Hz	76	E_5	659.255 Hz	126	$F\sharp_9$	11839.8 Hz
27	$E\flat_1$	38.8909 Hz	77	F_5	698.456 Hz	127	G_9	12543.9 Hz
28	E_1	41.2034 Hz	78	$F\sharp_5$	739.989 Hz	128	$A\flat_9$	13289.8 Hz
29	F_1	43.6535 Hz	79	G_5	783.991 Hz	129	A_9	14080.0 Hz
30	$F\sharp_1$	46.2493 Hz	80	$A\flat_5$	830.609 Hz	130	$B\flat_9$	14917.2 Hz
31	G_1	48.9994 Hz	81	A_5	880.000 Hz	131	B_9	15804.3 Hz
32	$A\flat_1$	51.9131 Hz	82	$B\flat_5$	932.328 Hz	132	C_{10}	16744.0 Hz
33	A_1	55.0000 Hz	83	B_5	987.767 Hz	133	$C\sharp_{10}$	17739.7 Hz
34	$B\flat_1$	58.2705 Hz	84	C_6	1046.50 Hz	134	D_{10}	18794.5 Hz
35	B_1	61.7354 Hz	85	$C\sharp_6$	1108.73 Hz	135	$E\flat_{10}$	19912.1 Hz
36	C_2	65.4064 Hz	86	D_6	1174.66 Hz	136	E_{10}	21096.2 Hz
37	$C\sharp_2$	69.2957 Hz	87	$E\flat_6$	1244.51 Hz	137	F_{10}	22350.6 Hz
38	D_2	73.4162 Hz	88	E_6	1318.51 Hz	138	$F\sharp_{10}$	23679.6 Hz
39	$E\flat_2$	77.7817 Hz	89	F_6	1396.91 Hz	139	G_{10}	25087.7 Hz
40	E_2	82.4069 Hz	90	$F\sharp_6$	1479.98 Hz	140	$A\flat_{10}$	26579.5 Hz
41	F_2	87.3071 Hz	91	G_6	1567.98 Hz	141	A_{10}	28160.0 Hz
42	$F\sharp_2$	92.4986 Hz	92	$A\flat_6$	1661.22 Hz	142	$B\flat_{10}$	29834.5 Hz
43	G_2	97.9989 Hz	93	A_6	1760.00 Hz	143	B_{10}	31608.5 Hz
44	$A\flat_2$	103.826 Hz	94	$B\flat_6$	1864.66 Hz	144	C_{11}	33488.1 Hz
45	A_2	110.000 Hz	95	B_6	1975.53 Hz	145	$C\sharp_{11}$	35479.4 Hz
46	$B\flat_2$	116.541 Hz	96	C_7	2093.00 Hz	146	D_{11}	37589.1 Hz
47	B_2	123.471 Hz	97	$C\sharp_7$	2217.46 Hz	147	$E\flat_{11}$	39824.3 Hz
48	C_3	130.813 Hz	98	D_7	2349.32 Hz	148	E_{11}	42192.3 Hz
49	$C\sharp_3$	138.591 Hz	99	$E\flat_7$	2489.02 Hz	149	F_{11}	44701.2 Hz

b) Correlation of Sound Intensity→Sound Pressure in 2 dB-Steps
($p = 20\sqrt{i}$, where p is the Pressure and i the Intensity; ** = growth)

Sound Intensity**		dB(SPL)	Sound pressure	
10.000 fW/M^2	$\times 10^{-2}$	−20	2.000 µPa	$\times 10^{-1}$
15.849 fW/ M^2		−18	2.518 µPa	
25.119 fW/ M^2		−16	3.170 µPa	
39.811 fW/ M^2		−14	3.991 µPa	
63.096 fW/ M^2		−12	5.024 µPa	
100.000 fW/ M^2		−10	6.325 µPa	
158.489 fW/ M^2		−8	7.962 µPa	
251.189 fW/ M^2		−6	10.024 µPa	
398.107 fW/ M^2		−4	12.619 µPa	
630.957 fW/ M^2		−2	15.887 µPa	
1.000 pW/ M^2	$\times 10^{0}$	0	20.000 µPa	$\times 10^{0}$
1.585 pW/ M^2		2	25.179 µPa	
2.512 pW/ M^2		4	31.698 µPa	
3.981 pW/ M^2		6	39.905 µPa	
6.310 pW/ M^2		8	50.238 µPa	
10.000 pW/ M^2		10	63.246 µPa	
15.849 pW/ M^2		12	79.621 µPa	
25.119 pW/ M^2		14	100.237 µPa	
39.811 pW/ M^2		16	126.191 µPa	
63.096 pW/ M^2		18	158.866 µPa	
100.000 pW/ M^2	$\times 10^{2}$	20	200.00 µPa	$\times 10^{1}$
158.489 pW/ M^2		22	251.785 µPa	
251.189 pW/ M^2		24	316.979 µPa	
398.107 pW/M^2		26	399.052 µPa	
630.957 pW/M^2		28	502.377 µPa	
1.000 nW/M^2		30	632.456 µPa	
1.585 nW/M^2		32	796.214 µPa	
2.512 nW/M^2		34	1.002 mPa	
3.981 nW/M^2		36	1.262 mPa	
6.310 nW/M^2		38	1.589 mPa	
10.000 nW/M^2	$\times 10^{4}$	40	2.000 mPa	$\times 10^{2}$
15.849 nW/M^2		42	2.518 mPa	
25.119 nW/M^2		44	3.170 mPa	
39.811 nW/M^2		46	3.991 mPa	
63.096 nW/M^2		48	5.024 mPa	
100.000 nW/M^2		50	6.325 mPa	

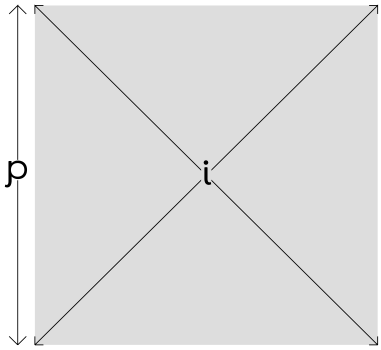
Sound Intensity**		dB(SPL)	Sound pressure	
158.489 nW/M^2		52	7.962 mPa	
251.189 nW/M^2		54	10.024 mPa	
398.107 nW/M^2		56	12.619 mPa	
630.957 nW/M^2		58	15.887 mPa	
1.000 µW/M^2	$\times 10^{6}$	60	20.000 mPa	$\times 10^{3}$
1.585 µW/M^2		62	25.179 mPa	
2.512 µW/M^2		64	31.698 mPa	
3.981 µW/M^2		66	39.905 mPa	
6.310 µW/M^2		68	50.238 mPa	
10.000 µW/M^2		70	63.246 mPa	
15.849 µW/M^2		72	79.621 mPa	
25.119 µW/M^2		74	100.237 mPa	
39.811 µW/M^2		76	126.191 mPa	
63.096 µW/M^2		78	158.866 mPa	
100.000 µW/M^2	$\times 10^{8}$	80	200.000 mPa	$\times 10^{4}$
158.489 µW/M^2		82	251.785 mPa	
251.189 µW/M^2		84	316.979 mPa	
398.107 µW/M^2		86	399.052 mPa	
630.957 µW/M^2		88	502.377 mPa	
1.000 mW/M^2		90	632.456 mPa	
1.585 mW/M^2		92	796.214 mPa	
2.512 mW/M^2		94	1.002 Pa	
3.981 mW/M^2		96	1.262 Pa	
6.310 mW/M^2		98	1.589 Pa	
10.000 mW/M^2	$\times 10^{10}$	100	2.000 Pa	$\times 10^{5}$
15.849 mW/M^2		102	2.518 Pa	
25.119 mW/M^2		104	3.170 Pa	
39.811 mW/M^2		106	3.991 Pa	
63.096 mW/M^2		108	5.024 Pa	
100.000 mW/M^2		110	6.325 Pa	
158.489 mW/M^2		112	7.962 Pa	
251.189 mW/M^2		114	10.024 Pa	
398.107 mW/M^2		116	12.619 Pa	
630.957 mW/M^2		118	15.887 Pa	
1.000 W/M^2	$\times 10^{12}$	120	20.000 Pa	$\times 10^{6}$
1.585 W/M^2		122	25.179 Pa	

Units:

M = metre (length)
kg = kilogramme (mass)
s = second (time)
sqm = square metre (area) = M^2
N = Newton (force) = kg.M.s^{-2}
Pa = Pascal (pressure) = N/sqm = kg.M^{-1}.s^{-2}
b = Bar (pressure) = 100000 Pa
J = Joule (work) = N.M = kg.M^2.s^{-2}
W = Watt (power) = J/s = N.M/s = kg.M^{-2}.s^{-3}

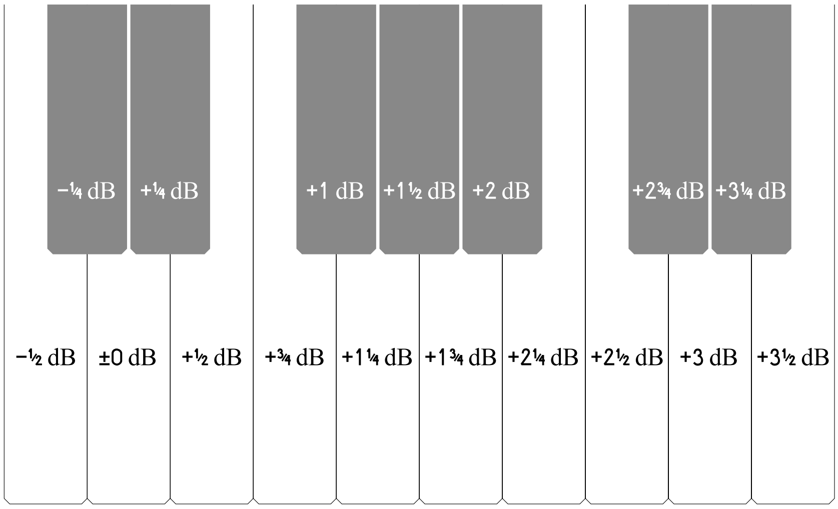
Powers	-th	-fold
10^{1}	d = deci-	da = deca-
10^{2}	c = centi-	h = hecto-
10^{3}	m = milli-	k = kilo-
10^{6}	µ = micro-	M = mega-
10^{9}	n = nano-	G = giga-
10^{12}	p = pico-	T = tera-
10^{15}	f = femto-	P = peta-
10^{18}	a = atto-	E = exa-
10^{21}	z = zepto-	Z = zetta-
10^{24}	y = yocto-	Y = yotta-

Γ09 – Showing sound pressure (p^2) as proportional to squared sound intensity (i)

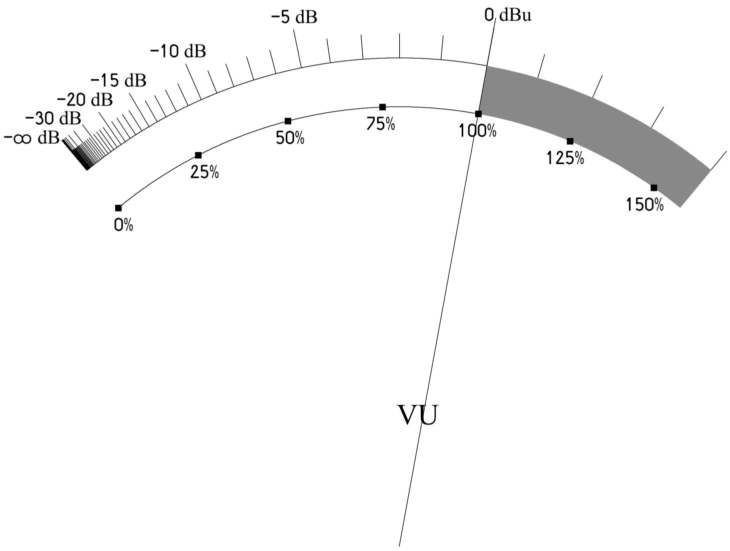


Γ10 – Illustration of the decibel-Scale

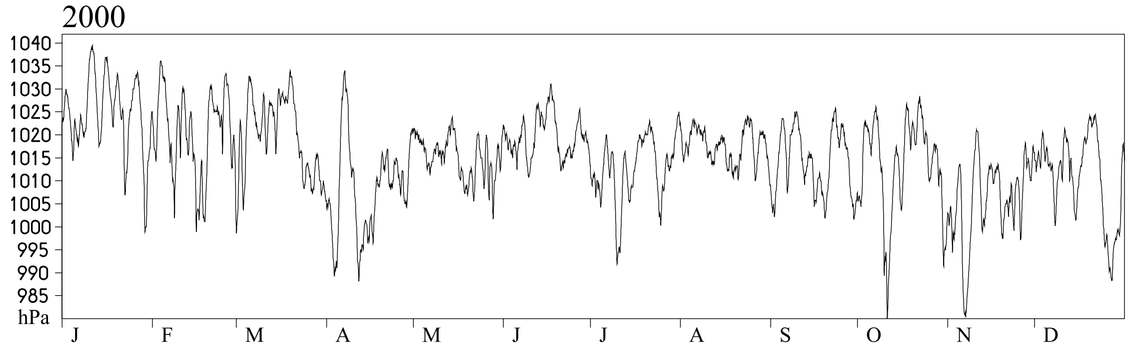
a) Sound intensity levels compared to pitch intervals

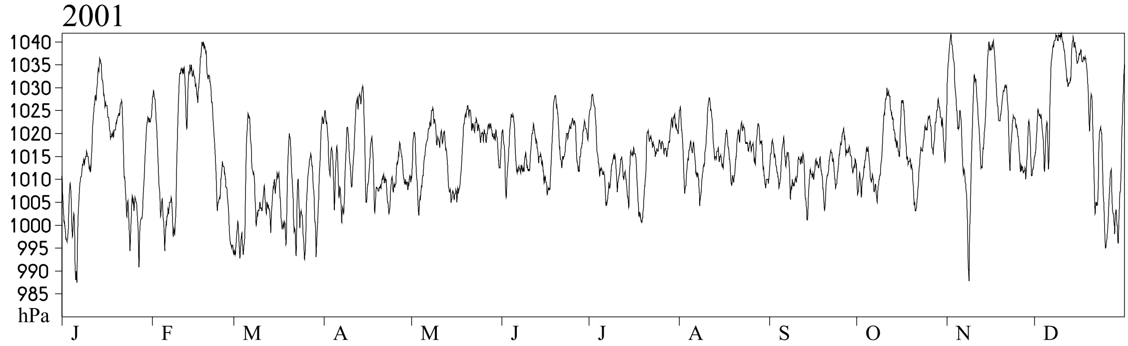


b) A VU-Meter, calibrated in % und in dBu

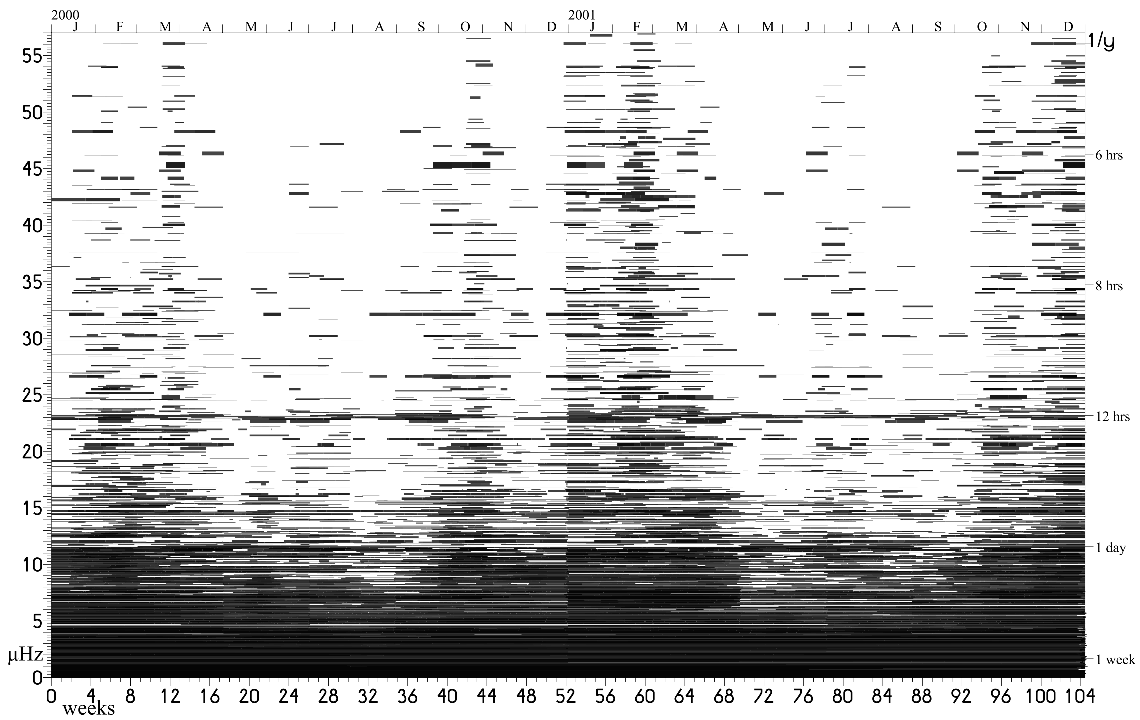


c) Air pressure as measured at Cologne-Bonn Airport in 2000 and 2001: an ‘infrasonic sound wave’ averaging 9.65 hPa = 153.67 dB(SPL)





d) Spectral representation of the wave shown in Γ08c in μHz against weeks

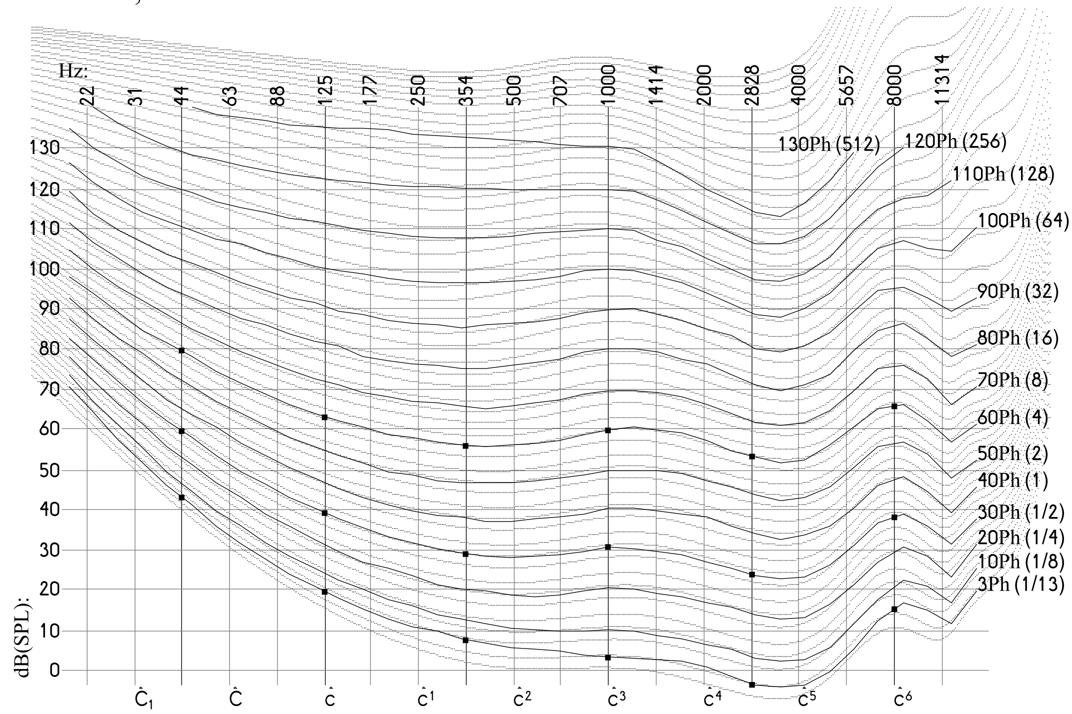


T11 – Examples of Loudness in dB(SPL)

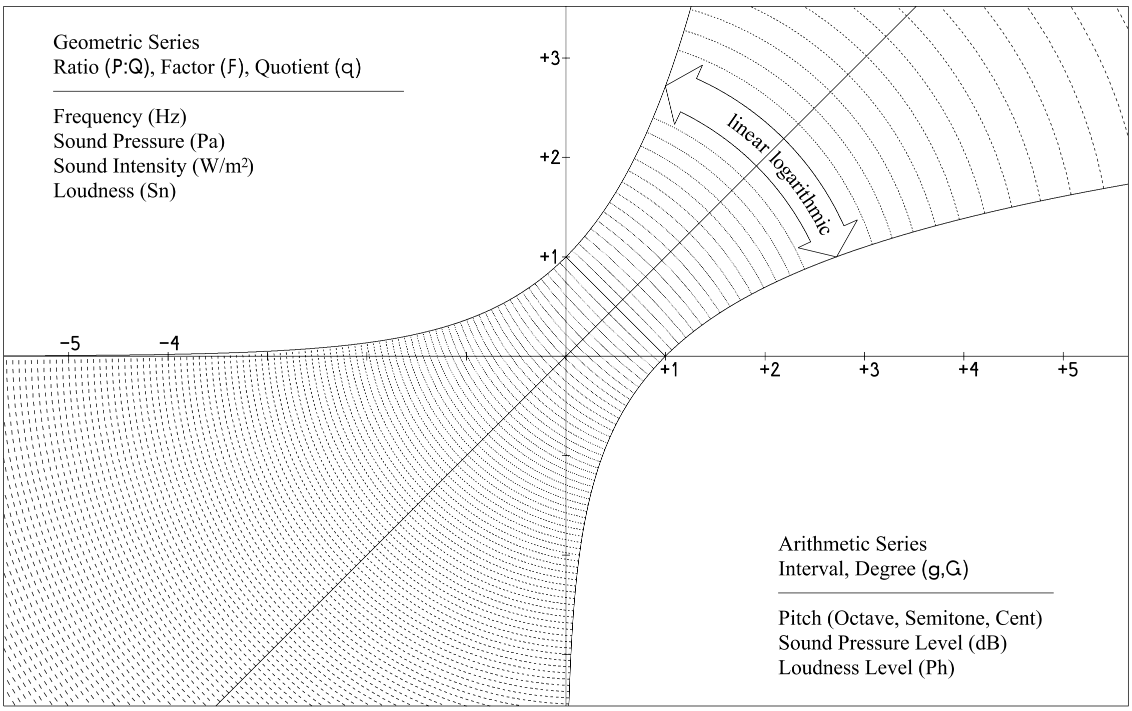
dB(SPL)	
0	‘practically inaudible’, →10: pin dropping
10	‘almost inaudible’, snow falling, →20: recording studio, →30: leaves rustling
20	→40: ‘very soft’, ‘quiet in the country’, light wind, clock ticking, bedroom
30	quiet garden, →35: very quiet room, whispering, 35: frequent upper permissible limit for residential areas and hospitals (nighttime), →40: background noise
40	enough to wake one up, →50: normal conversation, soft radio music, refrigerator, residential area without traffic (nighttime), library, →60: ‘quiet’
50	quiet river, residential area without traffic (daytime), car motor in neutral gear, photocopying machine, 50: frequent upper permissible limit for residential areas (daytime) and industrial areas (nighttime), →55: ‘disturbed sleep dangerous to health’, 55: EU-directive upper limit for ‘creative work’, →60: loud speech, TV at normal room level, →65: office noises
60	→70: typewriter, →80: ‘loud’, ‘disagreeable physiological stress reactions’, vacuum cleaner, automobile, lawn mower
70	‘stressful’, loud TV-set, main road with traffic, 70: EU-directive upper limit ‘routine work’, frequent upper permissible limit for industrial areas, →75: factory, →80: automobile in the city, →100: motorbicycle
80	80: frequent upper permissible limit for place of work, →85: train (medium speed), →90: truck in city, →110: ‘very loud’, loud cry, loud radio music, conveyor belt, automobile on motorway
90	fast train, loud factory, door slamming, 93: EU-directive ‘live entertainment’, →95: grinding lathe, →95: loud walkman, →100: dance floor, →105: pneumatic hammer
100	‘speech impossible’, circular saw, electric hammer drill, →110: car horn, →110: loud discotheque
110	→115: airplane noise, →140: ‘unbearable’, boiler room, planing mill
120	ear damage starts, thunder, commercial airplane, →130: running turbine
130	pain threshold, →135: jet plane in low flight
140	starting jet plane
150	(and higher) explosion
160	riveting machine

Γ12 – Objective vs. subjective, linear vs. logarithmic

a) Isophonic curves of equal subjective loudness (after Fletcher, Munson et al.), here algebraically and speculatively inter- und extrapolated; 44, 125, 354, 1000, 2828 and 8000 Hz marked for illustration.

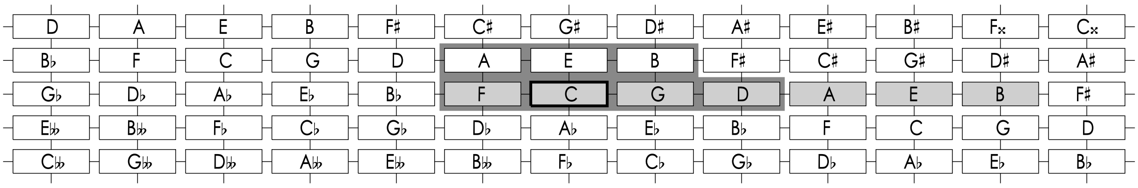


b) The symmetry of the linear and the logarithmic

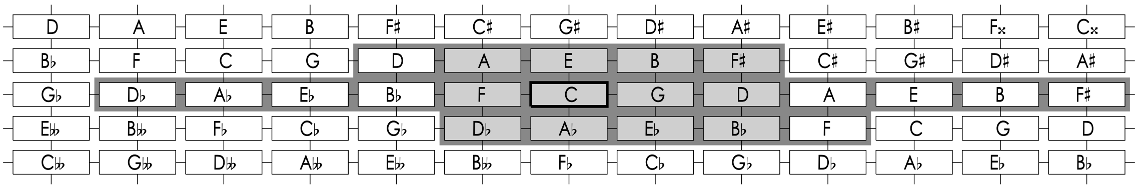


Γ13 – Network of fifths and thirds; derivation of intervals

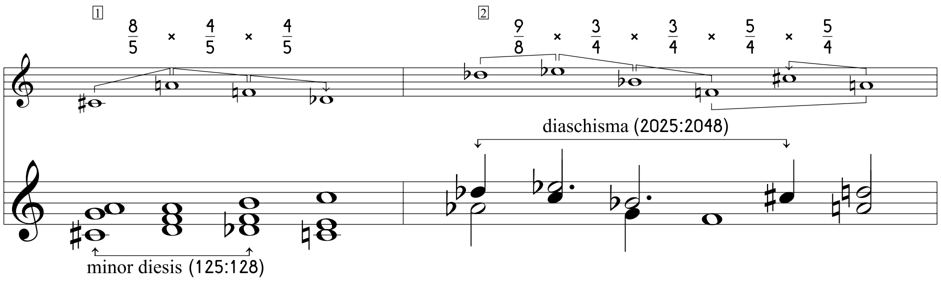
a) Two variants of the C-major scale, die Pythagorian 3-limit (grey-filled) and the Aristoxenian 5-limit (enclosed in darker grey).



b) The chromatic scale (grey-filled) and the North Indian *sruti* system (enclosed in darker grey)

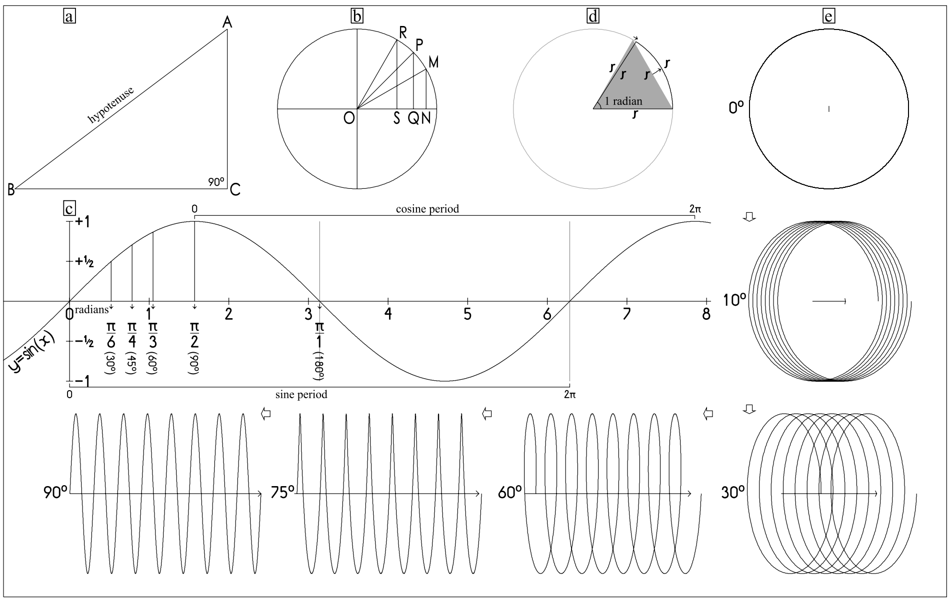


c) Modulations containing the minor diesis and the diaschisma



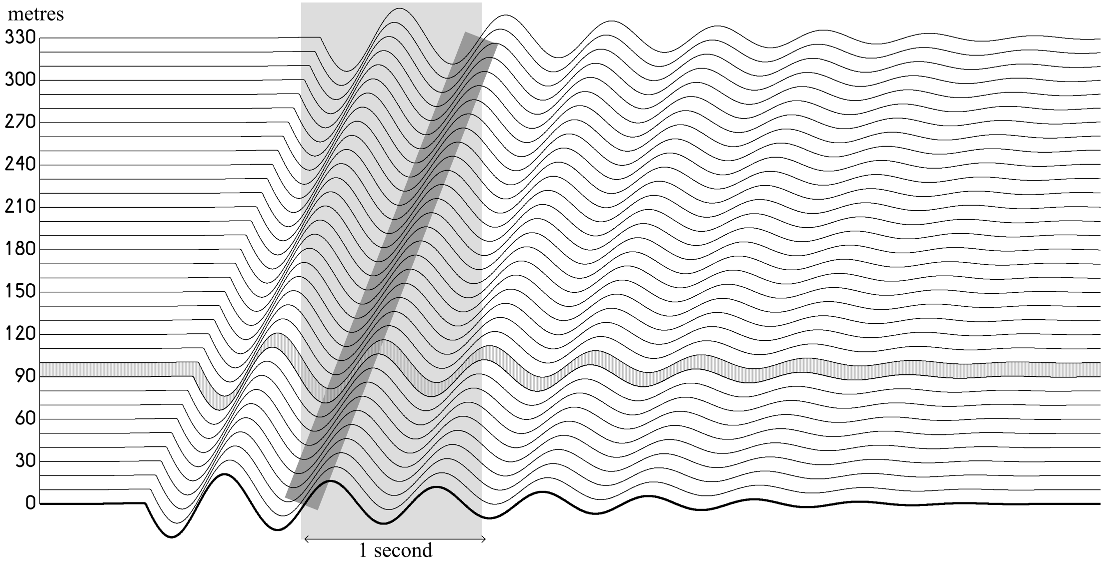
Γ14 – The derivation of trigonometric functions from the triangle and the circle

a) Right-angled triangle ABC
b) Sine exemplified by the radius of a circle, rotating through 30°, 45°, 60°
c) Sine traversing sines of the above three angles; cosine period for comparison
d) Pulling a side of an equilateral triangle (grey) to the circumference shrinks 60° to a radian
e) Transition of the circle through a spiral to a sine wave

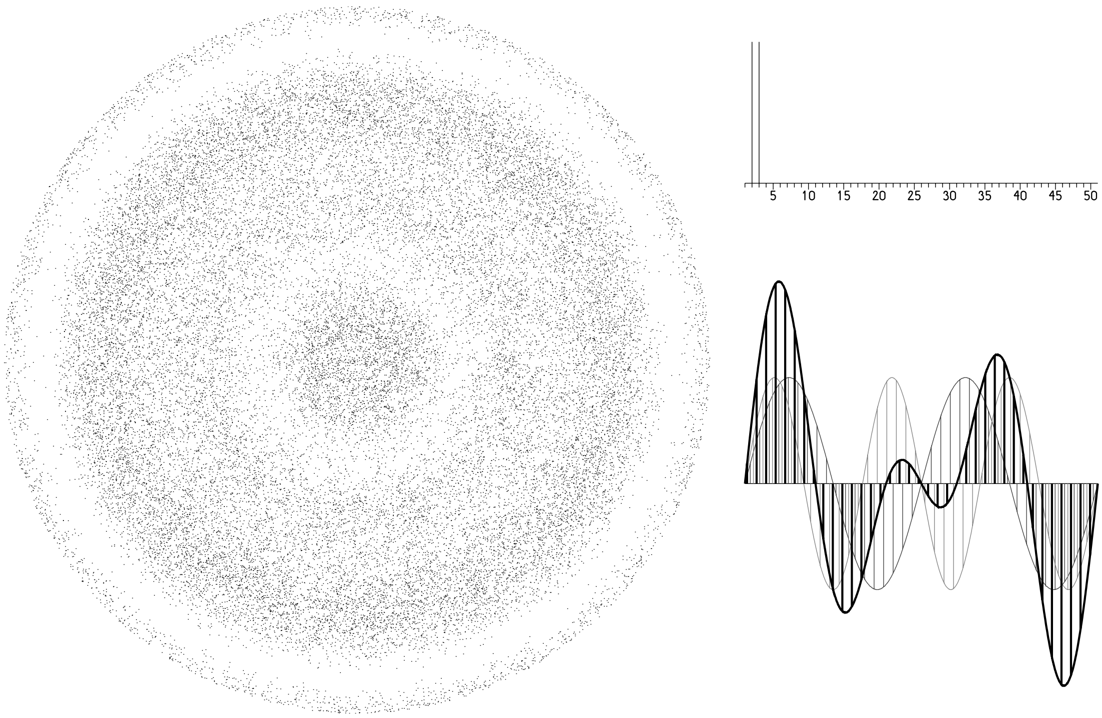


Γ15 – Sound wave and spectrum

a) Sound wave generation – a vibrating particle (thick line at bottom) passes movement on to air molecules (thin lines): a high pressure zone (marked by thick grey diagonal) moves 330m in 1 sec (s. light grey rectangle); the light grey strip shows the time-variant change of the molecular distance.

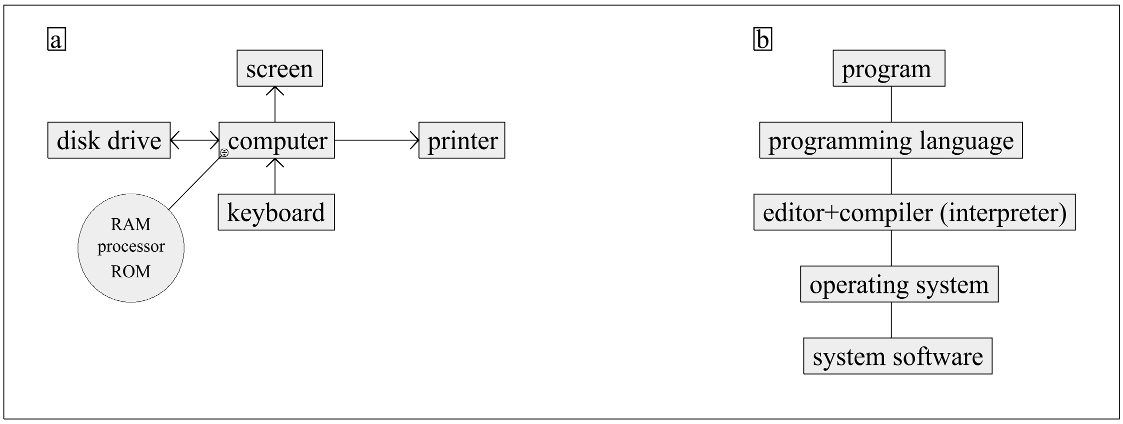


b) Addition of sound waves at the frequency ratio 2:3 (bottom right: components grey, summation curve black, vertical lines=values for RMS-evaluation), the spectrum (top right), a ‘molecular snapshot’ of the summation curve (left)



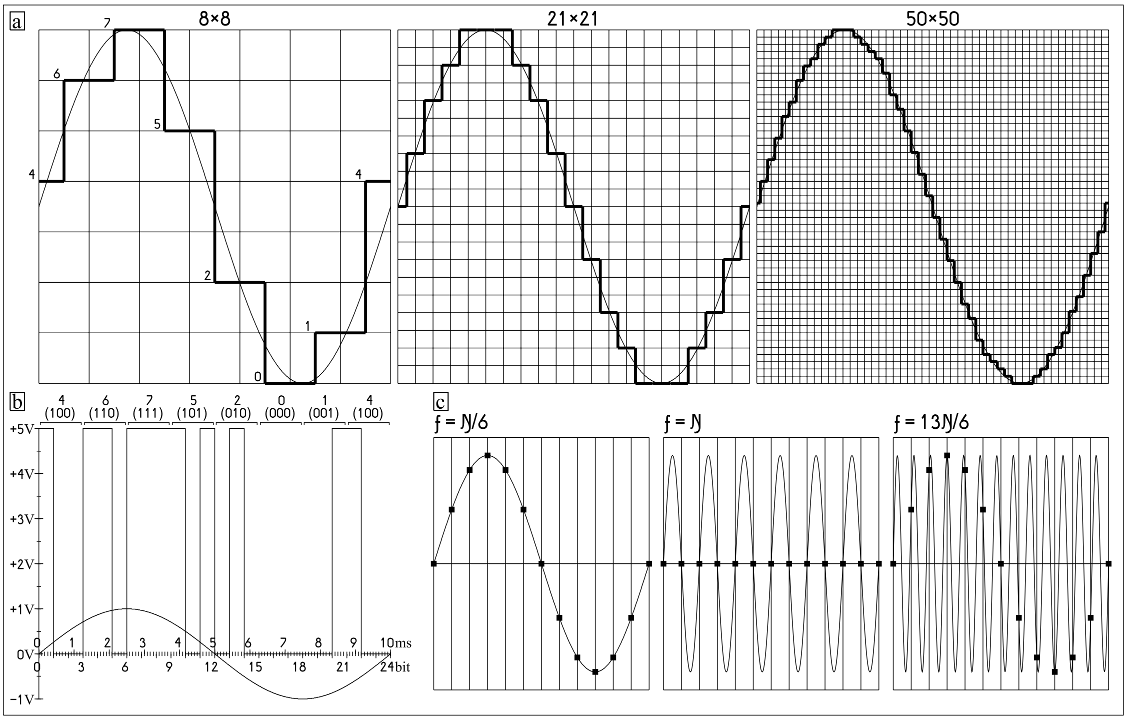
Γ16 – Hardware und Software

a) Hardware: machines surrounding the computer
b) Software: levels of working with the computer



Γ17 – Digitised sound waves; comparison with analogue

a) Stepwise improvement of the digitisation of a sine curve
b) Comparison of the leftmost (8×8) digital wave with an analogue 100 Hz sine
c) Sampling and aliasing of three frequencies with reference to the Nyquist limit Ŋ



T18 – ASCII-Code

Standard						Special* (excerpt)	
$20_h=032_d$	<empty>	$40_h=064_d$	@	$60_h=096_d$	`	$80_h=128_d$	Ç
$21_h=033_d$	!	$41_h=065_d$	A	$61_h=097_d$	a	$81_h=129_d$	ü
$22_h=034_d$	"	$42_h=066_d$	B	$62_h=098_d$	b	$82_h=130_d$	é
$23_h=035_d$	#	$43_h=067_d$	C	$63_h=099_d$	c	$83_h=131_d$	â
$24_h=036_d$	$	$44_h=068_d$	D	$64_h=100_d$	d	$84_h=132_d$	ä
$25_h=037_d$	%	$45_h=069_d$	E	$65_h=101_d$	e	$85_h=133_d$	à
$26_h=038_d$	&	$46_h=070_d$	F	$66_h=102_d$	f	$86_h=134_d$	å
$27_h=039_d$	'	$47_h=071_d$	G	$67_h=103_d$	g	$87_h=135_d$	ç
$28_h=040_d$	(	$48_h=072_d$	H	$68_h=104_d$	h	$88_h=136_d$	ê
$29_h=041_d$	)	$49_h=073_d$	I	$69_h=105_d$	i	$89_h=137_d$	ë
$2A_h=042_d$	*	$4A_h=074_d$	J	$6A_h=106_d$	j	$8A_h=138_d$	è
$2B_h=043_d$	+	$4B_h=075_d$	K	$6B_h=107_d$	k	$8B_h=139_d$	ï
$2C_h=044_d$	,	$4C_h=076_d$	L	$6C_h=108_d$	l	$8C_h=140_d$	î
$2D_h=045_d$	-	$4D_h=077_d$	M	$6D_h=109_d$	m	$8D_h=141_d$	ì
$2E_h=046_d$	.	$4E_h=078_d$	N	$6E_h=110_d$	n	$8E_h=142_d$	Ä
$2F_h=047_d$	/	$4F_h=079_d$	O	$6F_h=111_d$	o	$8F_h=143_d$	Å
$30_h=048_d$	0	$50_h=080_d$	P	$70_h=112_d$	p	$90_h=144_d$	É
$31_h=049_d$	1	$51_h=081_d$	Q	$71_h=113_d$	q	$91_h=145_d$	æ
$32_h=050_d$	2	$52_h=082_d$	R	$72_h=114_d$	r	$92_h=146_d$	Æ
$33_h=051_d$	3	$53_h=083_d$	S	$73_h=115_d$	s	$93_h=147_d$	ô
$34_h=052_d$	4	$54_h=084_d$	T	$74_h=116_d$	t	$94_h=148_d$	ö
$35_h=053_d$	5	$55_h=085_d$	U	$75_h=117_d$	u	$95_h=149_d$	ò
$36_h=054_d$	6	$56_h=086_d$	V	$76_h=118_d$	v	$96_h=150_d$	û
$37_h=055_d$	7	$57_h=087_d$	W	$77_h=119_d$	w	$97_h=151_d$	ù
$38_h=056_d$	8	$58_h=088_d$	X	$78_h=120_d$	x	$98_h=152_d$	ÿ
$39_h=057_d$	9	$59_h=089_d$	Y	$79_h=121_d$	y	$99_h=153_d$	Ö
$3A_h=058_d$	:	$5A_h=090_d$	Z	$7A_h=122_d$	z	$9A_h=154_d$	Ü
$3B_h=059_d$	;	$5B_h=091_d$	[	$7B_h=123_d$	{	$9B_h=155_d$	¢
$3C_h=060_d$	<	$5C_h=092_d$	\	$7C_h=124_d$	|	$9C_h=156_d$	£
$3D_h=061_d$	=	$5D_h=093_d$	]	$7D_h=125_d$	}	$9D_h=157_d$	¥
$3E_h=062_d$	>	$5E_h=094_d$	^	$7E_h=126_d$	~	$9E_h=158_d$	₧
$3F_h=063_d$	?	$5F_h=095_d$	_	$7F_h=127_d$	<del>	$9F_h=159_d$	ƒ

...etc...

* OEM extended ASCII

L19 – Programming the Fibonacci Series in a number of languages

a) Basic language structure (command separators: in Basic ‘:’, in Fortran ‘$’)

1) Basic

```
a=1 : b=1
print a,b
for i=3 to 10 do
c=a+b
print c
a=b : b=c
next i
```

2) Fortran

```
    a=1 $ b=1
    print a,' ',b,' '
    do 111 i=3,10
    c=a+b
    print c,' '
    a=b $ b=c
111 continue
```

3) Pascal

```
a:=1; b:=1;
write(a,' ',b,' ');
for i:=3 to 10 do
begin c:=a+b;
      write(c,' ');
      a:=b; b:=c;
end;
```

4) C

```
a=1; b=1;
printf("%d %d", a, b);
for (i=3; i<=10; i++)
{ c = a+b;
  printf(" %d", c);
  a=b; b=c;
}
```

b) Comparison and description of the individual statements

Language	Initial assignment	Write statement	Loop start with counter `i` (counts from 3 to 10)
Basic:	`a=1 b=1`	`print a,b`	`for i=3 to 10 do`
Fortran:	`a=1 b=1`	`print a,' ',b,' '`	`do 111 i=3,10`
Pascal:	`a:=1; b:=1;`	`write(a,' ',b,' ');`	`for i:=3 to 10 do`
C:	`a=1; b=1;`	`printf("%d %d", a, b);`	`for (i=3; i<=10; i++)`

Language	Block start	Assignment	Write statement	Assignments	Block/Loop end
Basic:	loop start	`c=a+b`	`print c`	`a=b b=c`	`next i`
Fortran:	loop start	`c=a+b`	`print c,' '`	`a=b b=c`	`111 continue`
Pascal:	`begin`	`c:=a+b;`	`write(c,' ');`	`a:=b; b:=c;`	`end;`
C:	`{`	`c=a+b;`	`printf(" %d", c);`	`a=b; b=c;`	`}`

L20 – A C-Program

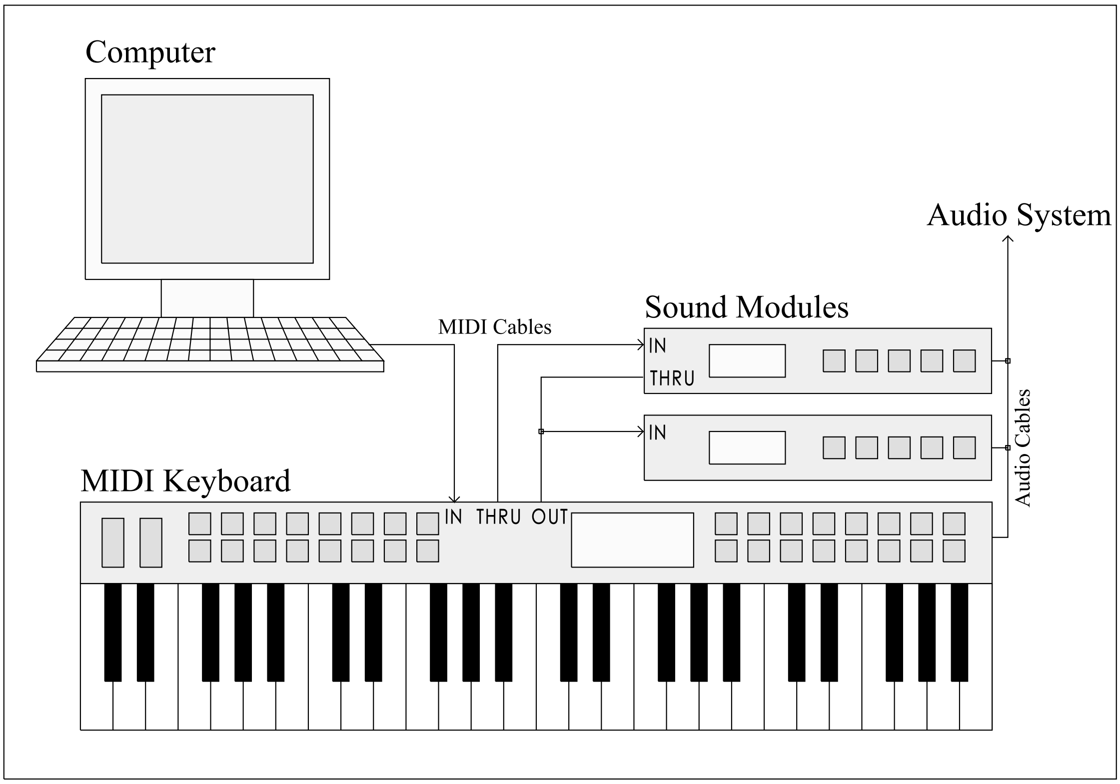
```
#include <stdio.h>
#include <math.h>
         /* Conversion of number ratios into cents & decibels */
int main ()
{  int p, q; float nLog2, nLog10, nLog_Quotient, ct, db;
   nLog2 = log(2); nLog10 = log(10); p = 1;
   printf("Enter P:Q.. (whole numbers>0; ");
   printf("program exit through invalid input)\n");
   while (p > 0)
   { printf("P: "); p = 0; scanf("%d", &p);
     if (p > 0)
     { printf("Q: "); scanf("%d", &q);
       nLog_Quotient = log(1.0 * q / p);
       ct = 1200 * nLog_Quotient / nLog2;
       db = 20 * nLog_Quotient / nLog10;
       printf("-----The ratio %d:%d corresponds to ",p,q);
       printf("%9.3f Ct or %6.3f dB\n Enter P:Q..\n",ct,db);
     }
   }
   printf("Program done.\n"); return 0;
}
```

Examples of screen display (keyboard input in bold):

Enter P:Q.. (whole numbers>0; program exit through invalid input)
P: **1**
Q: **2**
----- The ratio 1:2 corresponds to 1200.000 Ct or 6.021 dB
Enter P:Q..
P: **3**
Q: **2**
----- The ratio 3:2 corresponds to -701.955 Ct or -3.522 dB
Enter P:Q..
P: **x** Program done.

```
/* The numbers 0 - 255 from decimal into binary, octal, hexadecimal */

#include <stdio.h>

const int final_number = 255;

char hex_char(int num)
{ char c;
  if (num<10) c = num + 48; else c = num + 55;
  return c;
}

void convert (int base, int number,  char *code)
{ int i, power, buffer, remainder, divisor;

  power = 1; buffer = final_number;
  while (buffer>0)
  { power *= base;
    buffer /= base;
  }

  remainder = number; divisor = power / base; i = 0;
  while (divisor>0)
  { buffer = remainder / divisor;
    code[i] = hex_char(buffer);
    remainder %= divisor;
    divisor /= base;
    i++;
  } ;

  code[i] = '\0';
}
/*---------------------------main part---------------------------*/
int main ()
{ int counter;
  char digits[8];
  FILE *outfile;
  outfile = fopen("number_systems.txt", "w");

  for (counter=0; counter<=final_number; counter++)
  { fprintf(outfile, "%3dd = ", counter);
    convert( 2,counter,digits); fprintf(outfile, "%sb = ", digits);
    convert( 8,counter,digits); fprintf(outfile, "%so = ", digits);
    convert(16,counter,digits); fprintf(outfile, "%sh\n", digits);
  }

  fclose(outfile); printf("%s done","number_systems.txt");

  return 0;
}
```

Г22 – A possible network of MIDI-compatible devices



T23 – MIDI-Code

Status Bytes	Data Bytes	Command type
1000nnnn.b (=8mh)	pitch number [0–127], force of attack* [0–127]	Note Off
1001nnnn.b (=9mh)	pitch number [0–127], force of attack* [0=off, else 1–127]	Note On
1010nnnn.b (=Amh)	pitch number [0–127], force of attack* [0–127]	After-touch
1011nnnn.b (=Bmh)	control nummer [0–121: e.g. 7=Volume,....], control value [0–127]	Control Change
1100nnnn.b (=Cmh)	program number [0–127] (only 1 data byte!)	Program Change
1101nnnn.b (=Dmh)	pressure value [0–127] (only 1 data byte!)	Channel Pressure
1110nnnn.b (=Emh)	lower byte [0–127], upper byte [0–127]	Pitch Wheel
11110nnn.b (=Fmh)	brand dependent	System Exclusive

* 'velocity' for technocrats

L24 – Programming of MIDI in C (Schumann's *The Happy Farmer*)

a) More cumbersome solution

```
#include <stdio.h>
#include <time.h>
/*-----------------------MIDI-Library-------------------------*/
int channel_number;
void send(int value) { printf("%d ", value); }
void choose_channel(int value) { channel_number = value - 1; }
void play(int pitch, int force)
{ send(0x90 + channel_number); send(pitch); send(force); }
void damp(int pitch)
{ send(0x80 + channel_number); send(pitch); send(0); }
void wait(float seconds)
{ float clock_ticks, start;
  clock_ticks = seconds * CLOCKS_PER_SEC; start = clock();
  while (clock() – start < clock_ticks);
  printf("(%5.3f\")\n",seconds);
}
/* --------------------------M A I N-------------------------*/
int main ()
{ choose_channel(1);                          /* choose channel 1 */
  play(48,64);                                /* \c; [\]='depress'*/
  wait(0.25);                                 /* wait 0.25 secs. */
  damp(48);                                   /* /c; [/]='release'*/
  play(53,64);                                /* \f */
  wait(0.25);                                 /* wait 0.25 secs. */
  choose_channel(2);                          /* choose channel 2 */
  play(60,64); play(65,64); play(69,64);      /* \c'\f'\a' */
  wait(0.25);                                 /* wait 0.25 secs. */
  damp(60); damp(65); damp(69);               /* /c'/f'/a' */
  play(60,64); play(65,64); play(69,64);      /* \c'\f'\a' */
  wait(0.25);                                 /* wait 0.25 secs. */
  damp(60); damp(65); damp(69);               /* /c'/f'/a' */
  choose_channel(1);                          /* choose channel 1 */
  damp(53);                                   /* /f */
  play(57,64);                                /* \a */
  wait(0.25);                                 /* wait 0.25 secs. */
  damp(57);                                   /* /a */
  play(60,64);                                /* \c' */
  wait(0.5);                                  /* wait 0.5 secs. */
  damp(60);                                   /* /c' */
}
```

Screen display:

```
144 48 64 (0.250")
128 48 0  144 53 64 (0.250")
145 60 64 145 65 64 145 69 64 (0.250")
129 60 0  129 65 0  129 69 0  145 60 64 145 65 64 145 69 64 (0.250")
129 60 0  129 65 0  129 69 0  128 53 0  144 57 64 (0.250")
128 57 0  144 60 64 (0.500")
128 60 0
```

...L24...

b) More elegant solution

```
#include <stdio.h>
#include <time.h>
#define FILENAME "HAPPY_FARMER.TXT"
#define write printf
/*-----------------------MIDI-Library-----------------------*/
int  channel_number;
void send(int value) { write("%d ", value); }
void choose_channel(int value) { channel_number = value - 1; }
void play(int pitch, int force)
{ send(0x90 + channel_number); send(pitch); send(force); }
void damp(int pitch)
{ send(0x80 + channel_number); send(pitch); send(0); }
void wait(float seconds)
{ float clock_ticks, start;
  clock_ticks = seconds * CLOCKS_PER_SEC; start = clock();
  while (clock() – start < clock_ticks);
  write("(%5.3f\")\n",seconds);
}
/*------------------------read file-------------------------*/
int execute (FILE *score)
{ char instruction[8];
  int number1,number2;
  int input_amount;
  input_amount = fscanf(score, "%s %d %d",
                             instruction, &number1, &number2);
  if (input_amount != 3) return 0;
  if (instruction[0] == 'P') play(number1,number2);
  if (instruction[0] == 'D') damp(number1);
  if (instruction[0] == 'C') choose_channel(number1);
  if (instruction[0] == 'W') wait(number1 / 1000.0);
  return 1;
}
/*----------------------------MAIN--------------------------*/
int main ()
{ FILE *score;
  score = fopen(FILENAME, "r");
  while(execute(score));
  fclose(score);
  wait(2);
}
```

input ‘score’ file HAPPY_FARMER.TXT (read from left to right from column to column) :

```
C 1   0        C 2   0        P 60  64       C 1   0
P 48  64       P 60  64       P 65  64       D 53  0
W 250 0        P 65  64       P 69  64       P 57  64
D 48  0        P 69  64       W 250 0        W 250 0
P 53  64       W 250 0        D 60  0        D 57  0
W 250 0        D 60  0        D 65  0        P 60  64
               D 65  0        D 69  0        W 500 0
               D 69  0                       D 60
```

F25 – Formulæ for Harmonicity

a) for the indigestibility ξ of the natural number N
b) for the harmonicity H of an interval $P{:}Q$
c) for the max. power η of a prime p with min. harmonicity h und pitch range ω

a

$$\xi(N) = 2\sum_{r=1}^{\infty}\left(\frac{n_r(p_r-1)^2}{p_r}\right)$$

whereby:

1. $N = \prod_{r=1}^{\infty} p_r^{\,n_r}$
2. $N,\ n,\ p$ $\in$ natural numbers
3. p $\in$ prime numbers

b

$$H(P,Q) = \frac{\operatorname{sgn}\big(\xi(Q)-\xi(P)\big)}{\xi(P)+\xi(Q)}$$

whereby $\operatorname{sgn}(x)=-1$ for $x<0$, else $\operatorname{sgn}(x)=+1$

c

$$\eta = \left\lfloor \frac{\omega+(1/h)}{1+(\log(256)/\log(27))} \right\rfloor$$

for $p=2$, else

$$\eta = \left\lfloor \frac{\omega+(1/h)}{\xi(p)+(\log(p)/\log(2))} \right\rfloor$$

whereby:

1. p is the prime number for which η (the maximum power) is needed
2. h is the minimum harmonicity
3. ω is the pitch range in octaves

T26 – Basic Tables of Harmonicity

a) the Indigestibility $\xi(N)$ of the natural numbers 1–16

N	ξ(N)								
1	0.0000000	21	12.952381	41	78.048781	61	118.03279	81	10.666667
2	1.0000000	22	19.181818	42	13.952381	62	59.064516	82	79.048781
3	2.6666667	23	42.086957	43	82.046512	63	15.619048	83	162.02410
4	2.0000000	24	5.6666667	44	20.181818	64	6.0000000	84	14.952381
5	6.4000000	25	12.800000	45	11.733333	65	28.553846	85	36.517647
6	3.6666667	26	23.153846	46	43.086957	66	21.848485	86	83.046512
7	10.285714	27	8.0000000	47	90.042553	67	130.02985	87	56.735632
8	3.0000000	28	12.285714	48	6.6666667	68	32.117647	88	21.181818
9	5.3333333	29	54.068966	49	20.571429	69	44.753623	89	174.02247
10	7.4000000	30	10.066667	50	13.800000	70	17.685714	90	12.733333
11	18.181818	31	58.064516	51	32.784314	71	138.02817	91	32.439560
12	4.6666667	32	5.0000000	52	24.153846	72	8.3333333	92	44.086957
13	22.153846	33	20.848485	53	102.03774	73	142.02740	93	60.731183
14	11.285714	34	31.117647	54	9.0000000	74	71.054054	94	91.042553
15	9.0666667	35	16.685714	55	24.581818	75	15.466667	95	40.505263
16	4.0000000	36	7.3333333	56	13.285714	76	36.105263	96	7.6666667
17	30.117647	37	70.054054	57	36.771930	77	28.467533	97	190.02062
18	6.3333333	38	35.105263	58	55.068966	78	25.820513	98	21.571429
19	34.105263	39	24.820513	59	114.03390	79	154.02532	99	23.515152
20	8.4000000	40	9.4000000	60	11.066667	80	10.400000	100	14.800000

. . .T26. . .

b) Maximum powers of the primes 2−23 at different minimum harmonicities

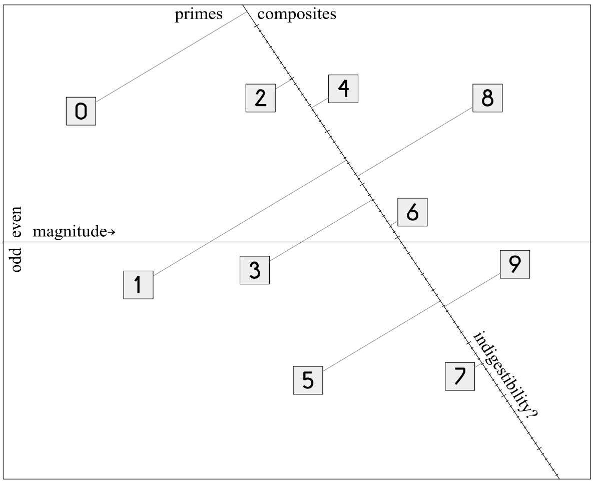
Minimum Harmonicity	Maximum Powers of Primes 2	3	5	7	11	13	17	19	23	Intervals in an octave
0.10	4	2	1	0	0	0	0	0	0	8
0.09	4	2	1	0	0	0	0	0	0	9
0.08	5	3	1	1	0	0	0	0	0	13
0.07	5	3	1	1	0	0	0	0	0	18
0.06	6	4	2	1	0	0	0	0	0	23
0.05	7	4	2	1	0	0	0	0	0	37
0.04	9	6	2	1	1	1	0	0	0	77
0.03	12	8	3	2	1	1	1	0	0	213
0.02	19	11	5	3	2	1	1	1	1	1117
0.01	37	23	11	7	4	3	2	2	2...	?????

c) Complete Intraoctavic Intervals upwards of absolute Harmonicity 0.05

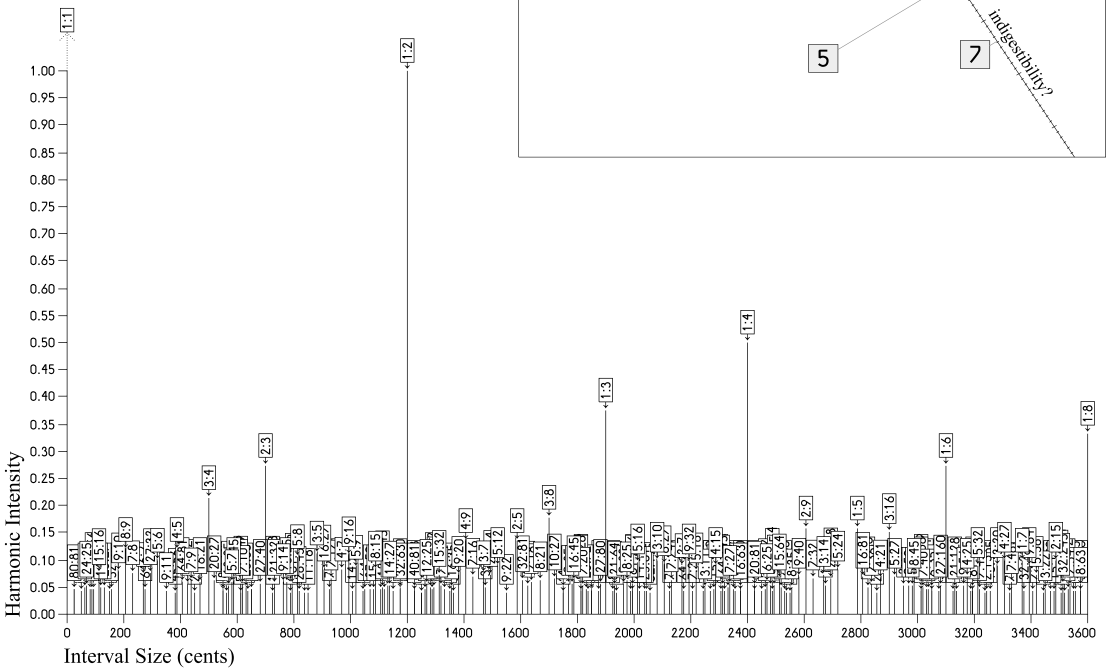
Interval size (Ct)	Prime Decomposition as Powers of 2	3	5	7	11	13	ratio	Harmonicity
0.000	0	0	0	0	0	0	1:1	+∞
70.672	−3	−1	+2	0	0	0	24:25	+0.054152
111.731	+4	−1	−1	0	0	0	15:16	−0.076531
182.404	+1	−2	+1	0	0	0	9:10	+0.078534
203.910	−3	+2	0	0	0	0	8:9	+0.120000
231.174	+3	0	0	−1	0	0	7:8	−0.075269
266.871	−1	−1	0	+1	0	0	6:7	+0.071672
294.135	+5	−3	0	0	0	0	27:32	−0.076923
315.641	+1	+1	−1	0	0	0	5:6	−0.099338
386.314	−2	0	+1	0	0	0	4:5	+0.119048
407.820	−6	+4	0	0	0	0	64:81	+0.060000
427.373	+5	0	−2	0	0	0	25:32	−0.056180
435.084	0	+2	0	−1	0	0	7:9	−0.064024
470.781	−4	+1	0	+1	0	0	16:21	+0.058989
498.045	+2	−1	0	0	0	0	3:4	−0.214286
519.551	−2	+3	−1	0	0	0	20:27	−0.060976
568.717	−1	−2	+2	0	0	0	18:25	+0.052265
582.512	0	0	−1	+1	0	0	5:7	+0.059932
590.224	−5	+2	+1	0	0	0	32:45	+0.059761
609.776	+6	−2	−1	0	0	0	45:64	−0.056391
617.488	+1	0	+1	−1	0	0	7:10	−0.056543
680.449	+3	−3	+1	0	0	0	27:40	+0.057471
701.955	−1	+1	0	0	0	0	2:3	+0.272727
729.219	+5	−1	0	−1	0	0	21:32	−0.055703
764.916	+1	−2	0	+1	0	0	9:14	+0.060172
772.627	−4	0	+2	0	0	0	16:25	+0.059524
792.180	+7	−4	0	0	0	0	81:128	−0.056604
813.686	+3	0	−1	0	0	0	5:8	−0.106383
884.359	0	−1	+1	0	0	0	3:5	+0.110294
905.865	−4	+3	0	0	0	0	16:27	+0.083333
933.129	+2	+1	0	−1	0	0	7:12	−0.066879
968.826	−2	0	0	+1	0	0	4:7	+0.081395
996.090	+4	−2	0	0	0	0	9:16	−0.107143
1017.596	0	+2	−1	0	0	0	5:9	−0.085227
1088.269	−3	+1	+1	0	0	0	8:15	+0.082873
1129.328	+4	+1	−2	0	0	0	25:48	−0.051370
1137.039	−1	+3	0	−1	0	0	14:27	−0.051852
1200.000	+1	0	0	0	0	0	1:2	+1.000000

Γ27 – Graphs on Harmonicity

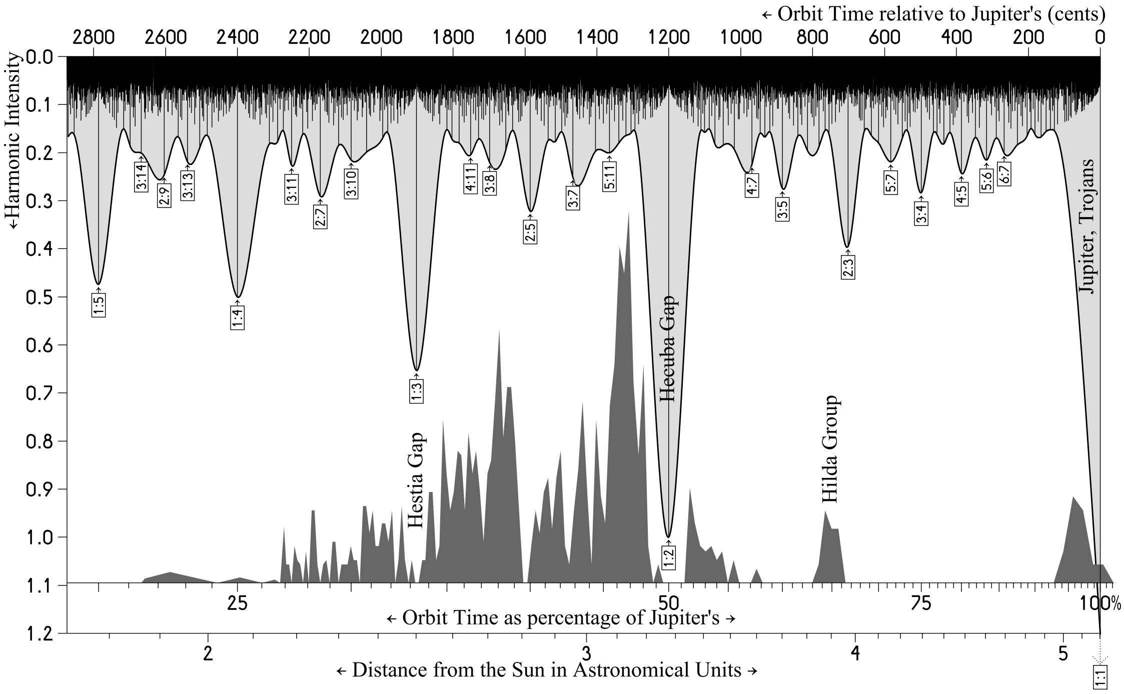
a) Multidimensional scaling of numerical similarity (Stanford 1975)



b) All 240 intervals of harmonic intensity ⩾0.04 in a three-8[ve] pitch range

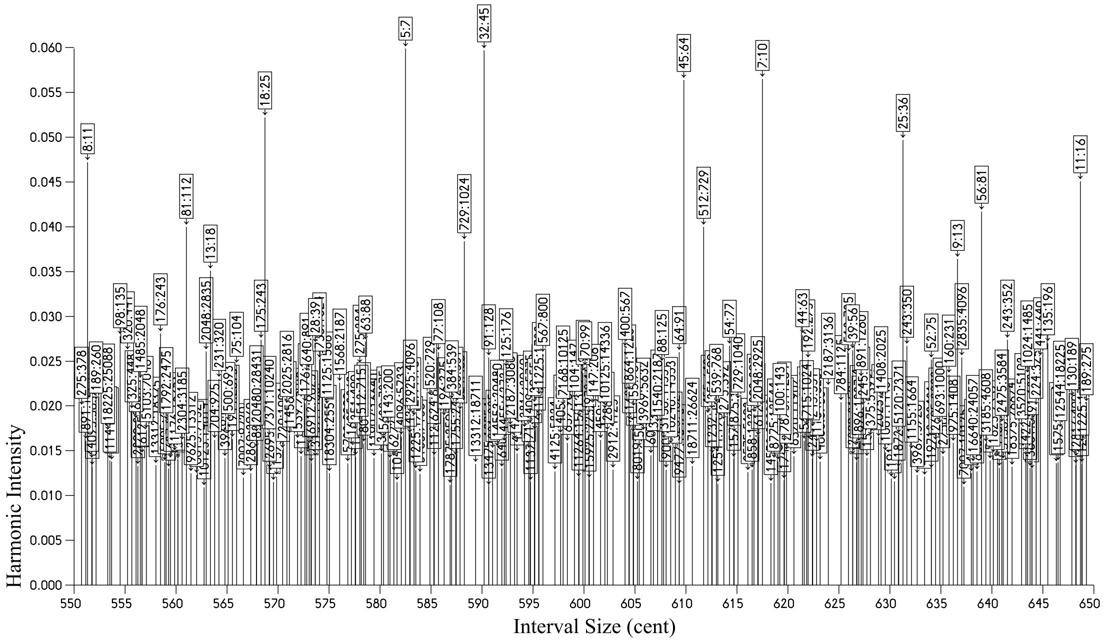


c) Asteroid Belt Density as against harmonic intensity of orbital intervals: prime enmity=1.2, display threshold |Ӿ|=0.02, 6438 intervals shown

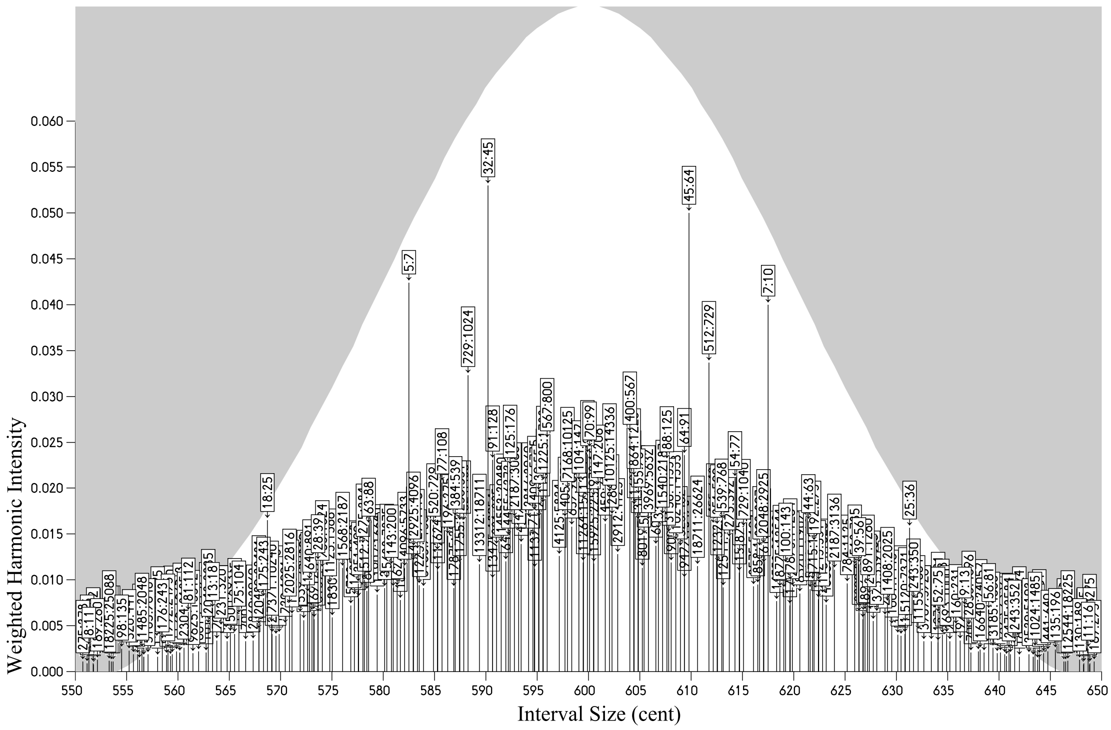


Г28 – Interval Size, Ratios and Harmonic Intensity

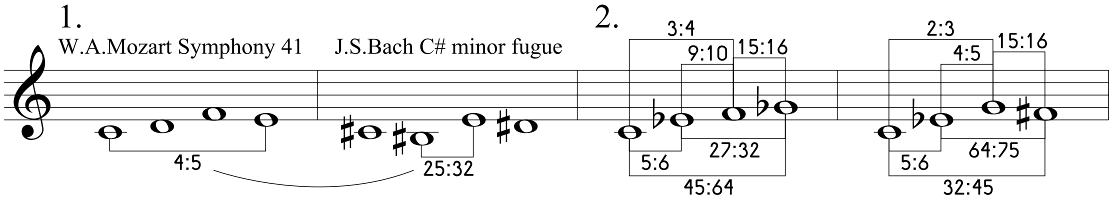
a$_1$) Unweighted...



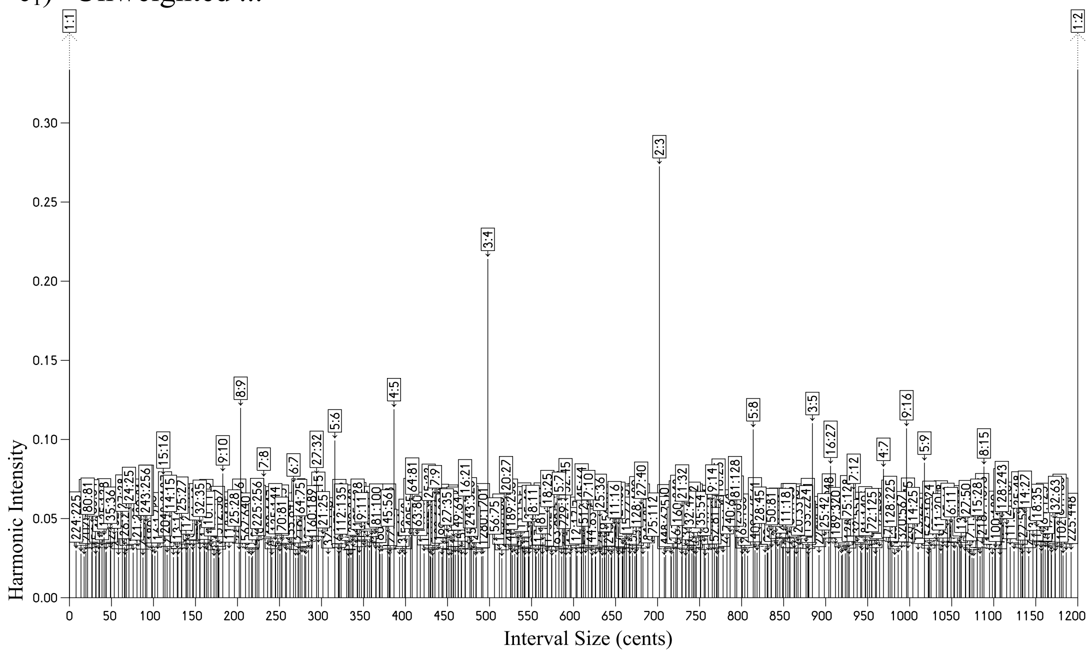
a$_2$) ...and Weighted Harmonic Intensity of 256 intervals (Ӿ≥0.02) from 550–650 Ct



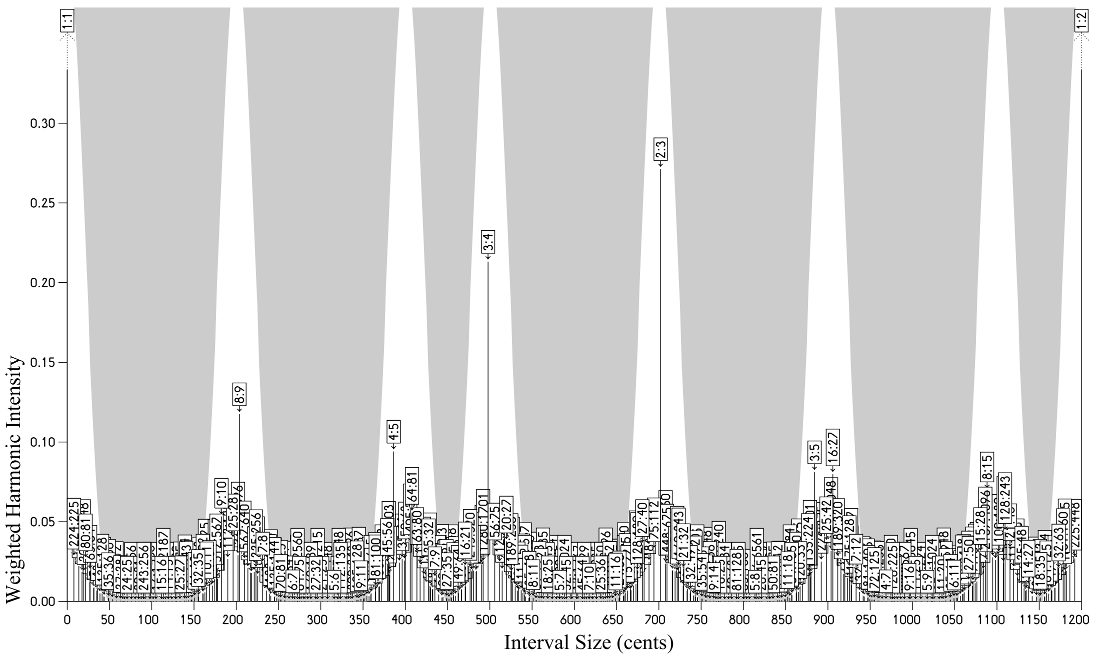
b) Two intervallic comparisons (left: 4:5 vs. 25:32, right: G♭ vs. F♯)



c1) Unweighted ...



c2) ... and Weighted Harmonic Intensity of 416 intervals (Ӈ≥0.025) from 0–1200 Ct
in tuning a major scale (nominal tolerance 50 Ct)



T29 – Rationalisations of scales

a) Best tunings of selected scales at given Minimum Harmonicities (MH), Nominal Tolerance (NT) and Alternative Tunings (AT), with the Specific Harmonicity (SH) of each tuning

1. <u>Major Scale</u>:

		Tuning							
MH=0.03									
AT=2									
NT=10	SH=0.1255	1/1	9/8	512/405	4/3	3/2	27/16	256/135	2/1
NT=20	SH=0.1957	1/1	9/8	81/64	4/3	3/2	27/16	15/8	2/1
NT=30-50	SH=0.2252	1/1	9/8	5/4	4/3	3/2	5/3	15/8	2/1
AT=3									
NT=10	SH=0.1842	1/1	9/8	81/64	4/3	3/2	27/16	243/128	2/1
NT=20	SH=0.1957	1/1	9/8	81/64	4/3	3/2	27/16	15/8	2/1
NT=30-50	SH=0.2252	1/1	9/8	5/4	4/3	3/2	5/3	15/8	2/1
MH=0.04									
AT=2									
NT=10	SH=0.1842	1/1	9/8	81/64	4/3	3/2	27/16	243/128	2/1
NT=20	SH=0.2252	1/1	9/8	5/4	4/3	3/2	5/3	15/8	2/1
NT=30-50	SH=0.2252	1/1	9/8	5/4	4/3	3/2	5/3	15/8	2/1
AT=3									
NT=10	SH=0.1842	1/1	9/8	81/64	4/3	3/2	27/16	243/128	2/1
NT=20-50	SH=0.2252	1/1	9/8	5/4	4/3	3/2	5/3	15/8	2/1
MH=0.05									
AT=2									
NT=10	SH=0.2196	1/1	9/8	81/64	4/3	3/2	27/16	27/16	2/1
NT=20-50	SH=0.2252	1/1	9/8	5/4	4/3	3/2	5/3	15/8	2/1
AT=3									
NT=10	SH=0.2196	1/1	9/8	81/64	4/3	3/2	27/16	27/16	2/1
NT=20-50	SH=0.2252	1/1	9/8	5/4	4/3	3/2	5/3	15/8	2/1

2. <u>Minor Scale</u>:

		Tuning							
MH=0.03									
AT=2									
NT=10	SH=0.1204	1/1	9/8	32/27	4/3	3/2	405/256	256/135	2/1
NT=20	SH=0.1618	1/1	9/8	32/27	4/3	3/2	128/81	15/8	2/1
NT=30-50	SH=0.2032	1/1	9/8	6/5	4/3	3/2	8/5	15/8	2/1
AT=3									
NT=10	SH=0.1514	1/1	9/8	32/27	4/3	3/2	128/81	256/135	2/1
NT=20	SH=0.1618	1/1	9/8	32/27	4/3	3/2	128/81	15/8	2/1
NT=30-50	SH=0.2032	1/1	9/8	6/5	4/3	3/2	8/5	15/8	2/1
MH=0.04									
AT=2									
NT=10	SH=0.1514	1/1	9/8	32/27	4/3	3/2	128/81	256/135	2/1
NT=20	SH=0.2032	1/1	9/8	6/5	4/3	3/2	8/5	15/8	2/1
NT=30-50	SH=0.2032	1/1	9/8	6/5	4/3	3/2	8/5	15/8	2/1
AT=3									
NT=10	SH=0.1514	1/1	9/8	32/27	4/3	3/2	128/81	256/135	2/1
NT=20-50	SH=0.2032	1/1	9/8	6/5	4/3	3/2	8/5	15/8	2/1
MH=0.05									
AT=2									
NT=10	SH=0.1690	1/1	9/8	32/27	4/3	3/2	128/81	128/81	2/1
NT=20-50	SH=0.2032	1/1	9/8	6/5	4/3	3/2	8/5	15/8	2/1
AT=3									
NT=10	SH=0.1690	1/1	9/8	32/27	4/3	3/2	128/81	128/81	2/1
NT=20-50	SH=0.2032	1/1	9/8	6/5	4/3	3/2	8/5	15/8	2/1

. . . Γ29a . . .

3. Whole Tone Scale:

MH=0.03

		Tuning						
AT=2								
NT=10	SH=0.0977	1/1	9/8	512/405	64/45	405/256	16/9	2/1
NT=20	SH=0.1229	1/1	9/8	81/64	64/45	128/81	16/9	2/1
NT=30	SH=0.1546	1/1	9/8	5/4	64/45	8/5	16/9	2/1
NT=40–50	SH=0.1615	1/1	10/9	5/4	64/45	8/5	16/9	2/1
AT=3								
NT=10	SH=0.1299	1/1	9/8	512/405	64/45	128/81	16/9	2/1
NT=20	SH=0.1414	1/1	10/9	512/405	64/45	128/81	16/9	2/1
NT=30–50	SH=0.1615	1/1	10/9	5/4	64/45	8/5	16/9	2/1

MH=0.04

AT=2								
NT=10	SH=0.1229	1/1	9/8	81/64	64/45	128/81	16/9	2/1
NT=20–50	SH=0.1615	1/1	10/9	5/4	64/45	8/5	16/9	2/1
AT=3								
NT=10	SH=0.1229	1/1	9/8	81/64	64/45	128/81	16/9	2/1
NT=20–50	SH=0.1615	1/1	10/9	5/4	64/45	8/5	16/9	2/1

MH=0.05

AT=2								
NT=10	SH=0.1229	1/1	9/8	81/64	64/45	128/81	16/9	2/1
NT=20–50	SH=0.1615	1/1	10/9	5/4	64/45	8/5	16/9	2/1
AT=3								
NT=10	SH=0.1229	1/1	9/8	81/64	64/45	128/81	16/9	2/1
NT=20–50	SH=0.1615	1/1	10/9	5/4	64/45	8/5	16/9	2/1

4. Bohlen-Pierce Scale (AT=2; values for NT=40 and NT=50 identical):

MH=0.03

NT	SH	Tuning
10	0.0796	1/1 49/45 32/27 9/7 7/5 32/21 5/3 1024/567 63/32 32/15 7/3 1024/405 224/81 3/1
20	0.0861	1/1 12/11 32/27 9/7 7/5 32/21 5/3 9/5 63/32 32/15 7/3 81/32 224/81 3/1
30	0.0959	1/1 35/32 32/27 32/25 7/5 32/21 5/3 9/5 2/1 32/15 7/3 81/32 224/81 3/1
40	0.1200	1/1 35/32 6/5 9/7 45/32 3/2 5/3 9/5 2/1 15/7 75/32 5/2 25/9 3/1

MH=0.04

NT	SH	
10	0.0905	1/1 35/32 32/27 9/7 45/32 32/21 5/3 9/5 63/32 32/15 7/3 81/32 96/35 3/1
20	0.0912	1/1 35/32 75/64 9/7 45/32 243/160 5/3 9/5 63/32 15/7 75/32 81/32 96/35 3/1
30	0.1058	1/1 35/32 6/5 32/25 7/5 32/21 5/3 9/5 2/1 32/15 7/3 5/2 96/35 3/1
40	0.1200	1/1 35/32 6/5 9/7 45/32 3/2 5/3 9/5 2/1 15/7 75/32 5/2 25/9 3/1

MH=0.05

NT	SH	
10	0.1102	1/1 1/1 32/27 9/7 45/32 32/21 5/3 9/5 9/5 32/15 7/3 81/32 81/32 3/1
20	0.1074	1/1 1/1 32/27 32/25 7/5 32/21 5/3 9/5 9/5 32/15 7/3 64/25 25/9 3/1
30	0.1424	1/1 1/1 6/5 32/25 7/5 3/2 5/3 9/5 2/1 32/15 7/3 5/2 14/5 3/1
40	0.1366	1/1 16/15 6/5 32/25 7/5 3/2 5/3 9/5 2/1 32/15 7/3 5/2 14/5 3/1

Based on [3–5–7] grid of other authorities:

1/1 27/25 25/21 9/7 7/5 75/49 5/3 9/5 49/25 15/7 7/3 63/25 25/9 3/1

. . . Γ29 . . .

b) Rationalised Tuning Network of Equal-Tempered Scales with intra-degree relationships and harmonicities

Scale degree:	1	2	3	4	5	6	7	8	9	10	11	12	13
Input size (Ct):	0	100	200	300	400	500	600	700	800	900	1000	1100	1200
Tuning:	1/1	16/15	9/8	6/5	5/4	4/3	45/32	3/2	8/5	5/3	9/5	15/8	2/1
Deviation (Ct):	0	+12	+4	+16	−14	−2	−10	+2	+14	−16	+18	−12	0
Note name:	C	D♭´₁	D²	E♭´¹	\`E	F₁	\`F♯²	G¹	A♭´	\`A₁	B♭´²	\`B¹	Ĉ

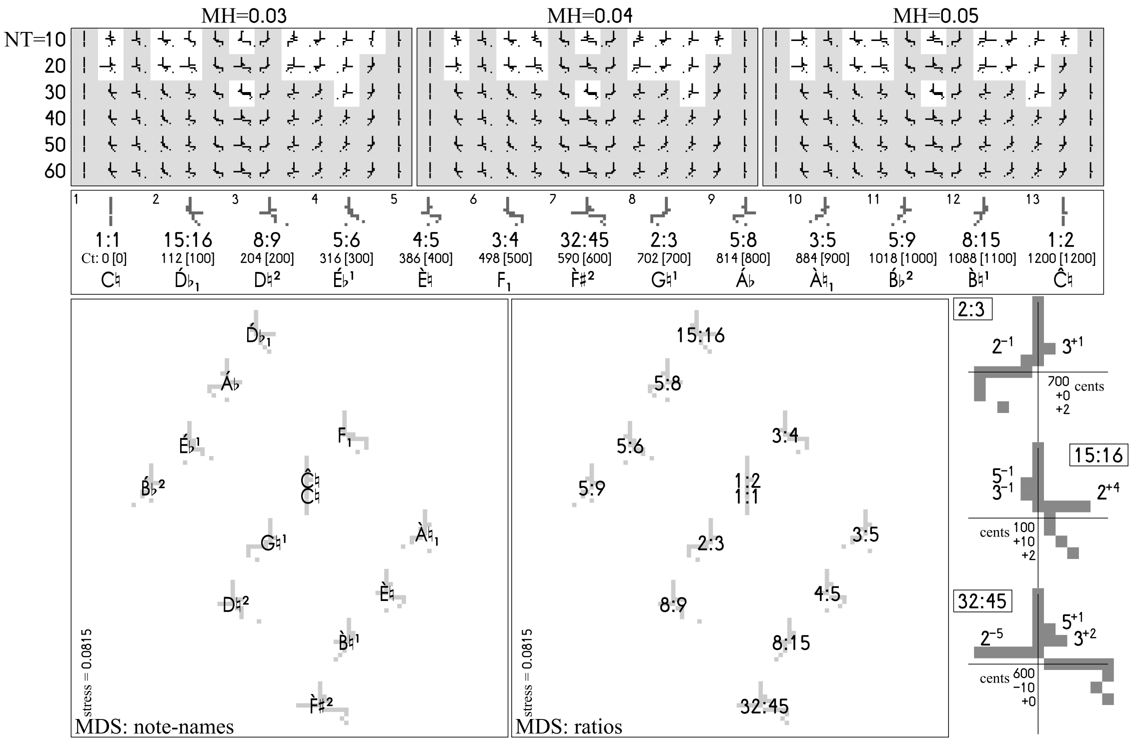
12 tone matrix:

	2→	3→	4→	5→	6→	7→	8→	9→	10→	11→	12→	13→	
	16/15	9/8	6/5	5/4	4/3	45/32	3/2	8/5	5/3	9/5	15/8	2/1	→1
	−0.077	+0.120	−0.099	+0.119	−0.214	+0.060	+0.273	−0.106	+0.110	−0.085	+0.083	+1.000	
		135/128	9/8	75/64	5/4	675/512	45/32	3/2	25/16	27/16	225/128	15/8	→2
		+0.047	+0.120	+0.047	+0.119	+0.034	+0.060	+0.273	+0.060	+0.083	+0.040	+0.083	
			16/15	10/9	32/27	5/4	4/3	64/45	40/27	8/5	5/3	16/9	→3
13 tone			−0.077	+0.079	−0.077	+0.119	−0.214	−0.056	+0.057	−0.106	+0.110	−0.107	
matrix:				25/24	10/9	75/64	5/4	4/3	25/18	3/2	25/16	5/3	→4
13→	256/243			+0.054	+0.079	+0.047	+0.119	−0.214	+0.052	+0.273	+0.060	+0.110	
	−0.047				16/15	9/8	6/5	32/25	4/3	36/25	3/2	8/5	→5
12→	10/9	135/128			−0.077	+0.120	−0.099	−0.056	−0.214	−0.050	+0.273	−0.106	
	+0.079	+0.047				135/128	9/8	6/5	5/4	27/20	45/32	3/2	→6
11→	32/27	9/8	16/15			+0.047	+0.120	−0.099	+0.119	−0.061	+0.060	+0.273	
	−0.077	+0.120	−0.077				16/15	256/225	32/27	32/25	4/3	64/45	→7
10→	5/4	1215/1024	9/8	135/128			−0.077	−0.038	−0.077	−0.056	−0.214	−0.056	
	+0.119	+0.034	+0.120	+0.047				16/15	10/9	6/5	5/4	4/3	→8
9→	21/16	5103/4096	189/160	567/512	21/20			−0.077	+0.079	−0.099	+0.119	−0.214	
	+0.059	+0.026	+0.034	+0.033	+0.047				25/24	9/8	75/64	5/4	→9
8→	48/35	729/560	216/175	81/70	192/175	256/245			+0.054	+0.120	+0.047	+0.119	
	−0.043	−0.027	−0.029	−0.035	−0.031	−0.029				27/25	9/8	6/5	→10
7→	35/24	2835/2048	21/16	315/256	7/6	10/9	1225/1152			−0.048	+0.120	−0.099	
	+0.045	+0.026	+0.059	+0.033	+0.072	+0.079	+0.022				25/24	10/9	→11
6→	32/21	81/56	48/35	9/7	128/105	512/441	10/9	256/245			+0.054	+0.079	
	−0.056	−0.042	−0.043	−0.064	−0.038	−0.029	+0.079	−0.029				16/15	→12
5→	8/5	243/160	36/25	27/20	32/25	128/105	7/6	192/175	21/20			−0.077	
	−0.106	+0.040	−0.050	−0.061	−0.056	−0.038	+0.072	−0.031	+0.047				
4→	12/7	729/448	54/35	81/56	48/35	64/49	5/4	288/245	9/8	15/14			
	−0.067	−0.031	−0.039	−0.042	−0.043	−0.038	+0.119	−0.027	+0.120	−0.049			
3→	16/9	27/16	8/5	3/2	64/45	256/189	35/27	128/105	7/6	10/9	28/27		
	−0.107	+0.083	−0.106	+0.273	−0.056	−0.038	+0.041	−0.038	+0.072	+0.079	+0.049		
2→	256/135	9/5	128/75	8/5	1024/675	4096/2835	112/81	2048/1575	56/45	32/27	448/405	16/15	
	−0.045	−0.085	−0.045	−0.106	−0.032	−0.025	+0.040	−0.025	+0.040	−0.077	−0.030	−0.077	
1→	2/1	243/128	9/5	27/16	8/5	32/21	35/24	48/35	21/16	5/4	7/6	9/8	135/128
	+1.000	+0.049	−0.085	+0.083	−0.106	−0.056	+0.045	−0.043	+0.059	+0.119	+0.072	+0.120	+0.047
	→14	→13	→12	→11	→10	→9	→8	→7	→6	→5	→4	→3	→2

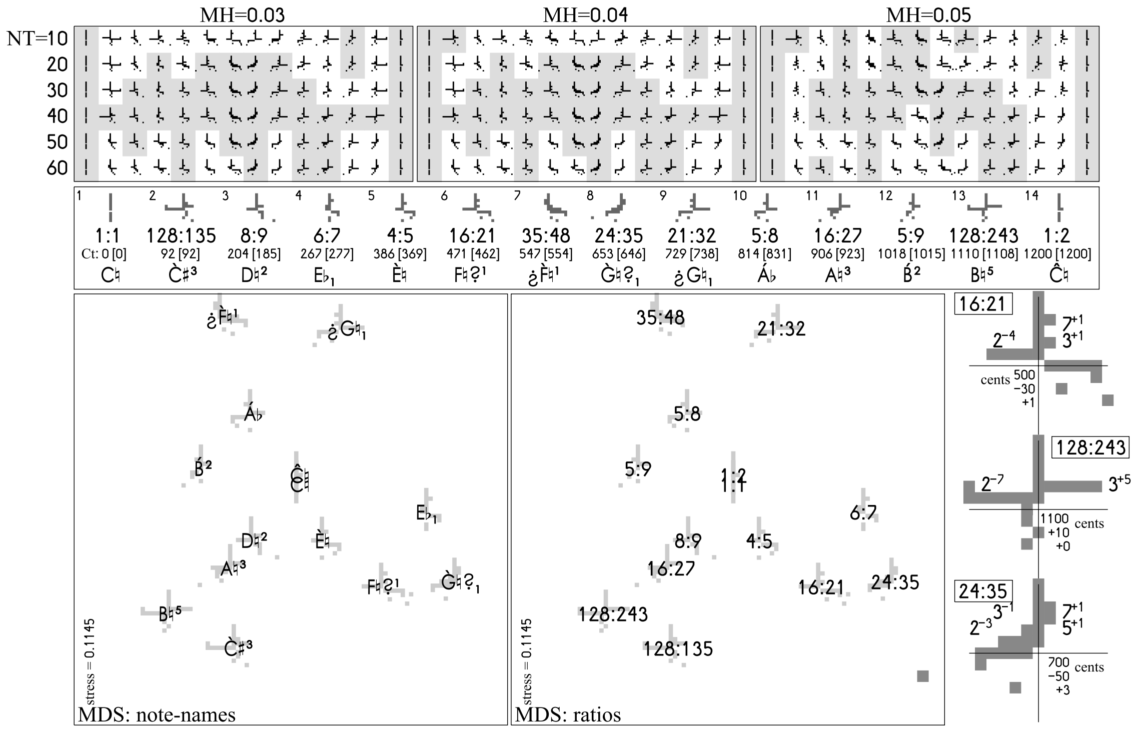
Scale degree:	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Input size (Ct):	0	92	185	277	369	462	554	646	738	831	923	1015	1108	1200
Tuning:	1/1	135/128	9/8	7/6	5/4	21/16	48/35	35/24	32/21	8/5	27/16	9/5	243/128	2/1
Deviation (Ct):	0	0	+19	−10	+17	+9	−7	+7	−9	−17	−17	+3	+2	0
Note name:	C	C♯\`³	D²	E♭?₁	\`\`E	F?¹	¿F¹	\`G?₁	¿G₁	A♭´	A³	B♭´²	B⁵	Ĉ

Γ30 – Ratioglyphic representation of tunings of equal-tempered scales – varied (guaranteed) minimum harmonicity (MH) und nominal tolerance (NT), ‘best tuning’ in grey fields with details in box thereunder – multidimensional scaling (MDS) bottom left (note-names) and middle (ratios) – bottom right: rational-intervallic explanation of selected ratioglyphs

a) 12-tone-scale

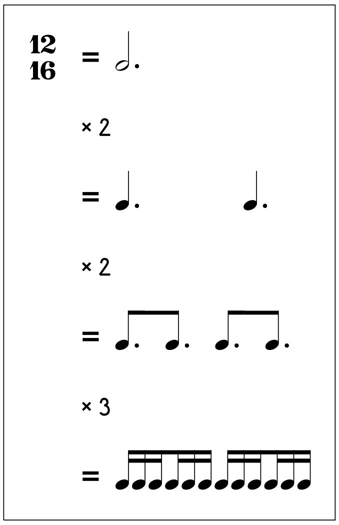


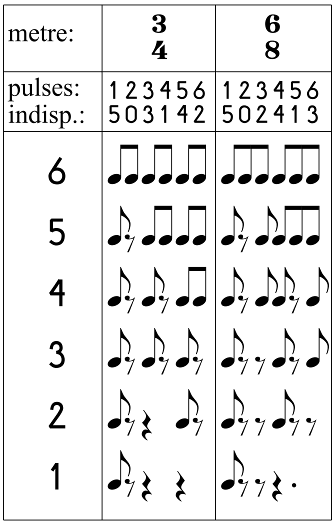
b) 13-tone-scale



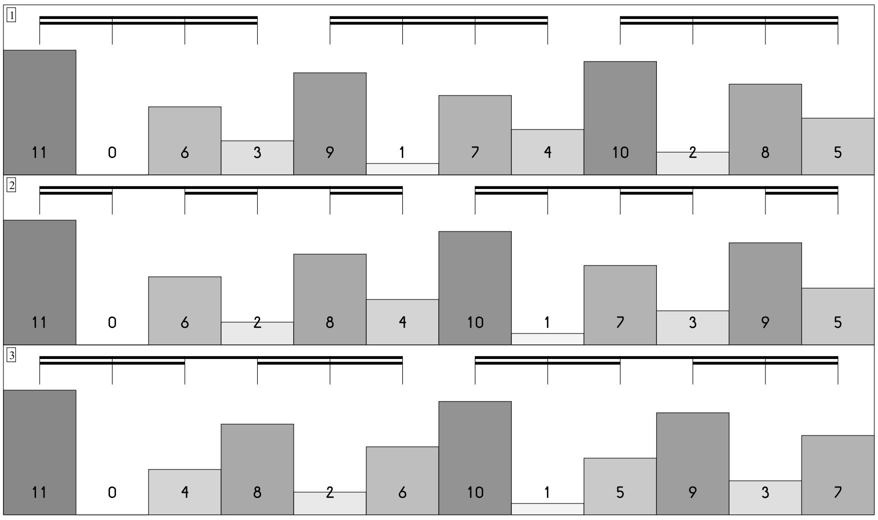
Γ31 – Graphs on Metricity

a) Stratification of a $^{12}/_{16}$-bar (2×2×3) b) Dilution (2nd level) of $^{3}/_{4}$ & $^{6}/_{8}$

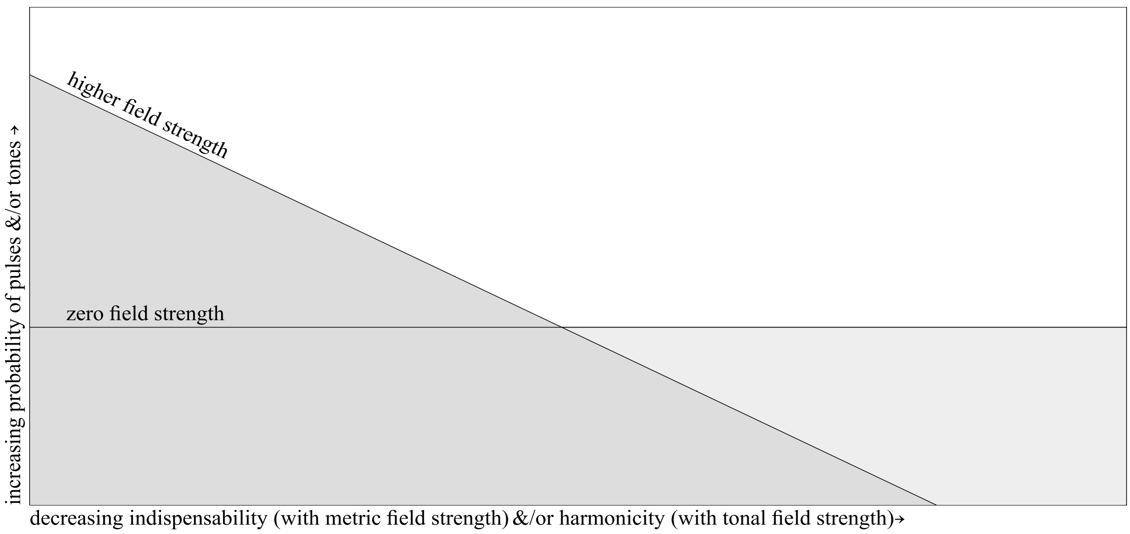




c) Indispensabilities for 1) $^{3}/_{4}$, 2) $^{6}/_{8}$ and 3) $^{12}/_{16}$ shown in grey shades and size



d) Metric/tonal field strength as relation 'oftenness'↔relevance of pulses/pitches



F32 – Formulæ for Metricity

a) Formula for the indispensability Ψ of the n^{th} pulse of a metre of stratification $p_1 \times p_2 \times p_3 \times \ldots \times p_z$

$$\Psi_z(n) = \sum_{r=0}^{z-1} \left\{ \prod_{i=0}^{z-r-1} p_i \Psi_{p_{z-r}} \left(1 + \left(\left[1 + \frac{(n-2) \bmod \prod_{j=1}^{z} p_j}{\prod_{k=0}^{r} p_{z+1-k}} \right] \bmod p_{z-r} \right) \right) \right\}$$

whereby (all variables being whole numbers):

1. $p_0 = p_{z+1} = 1$
2. n is the position in the bar of the pulse in question, starting at 1
3. p_j is the stratification divisor on level j
4. z is the number of levels in the stratification
5. $\Psi_p(x)$ is the Indispensability of the xth pulse of a first-order bar with the prime stratification p
6. $u \bmod v$ is the remainder of the division $(u+mv)/v$, by sufficiently large m never negative
7. $[x]$ is the whole-number component of x

b) Formula for the basic indispensability Ψ of the n^{th} pulse of a 1st-order metre with prime divisor p

If $p=2$, then $\Psi_p(n) = p-n$;

otherwise if $n=p-1$, then $\Psi_p(n) = [p/4]$

or else $\Psi_p(n) = \left[q + 2\sqrt{\frac{q+1}{p}} \right]$

whereby

1. n is the position in the bar of the pulse in question, starting at 1
2. $q = \Psi_{p-1}(n-[n/p])$
3. $\Psi_{p-1}(x)$ gives the Indispensabilities for a bar of pulses numbering $p-1$, factorised and stratified with primes in decreasing order of size

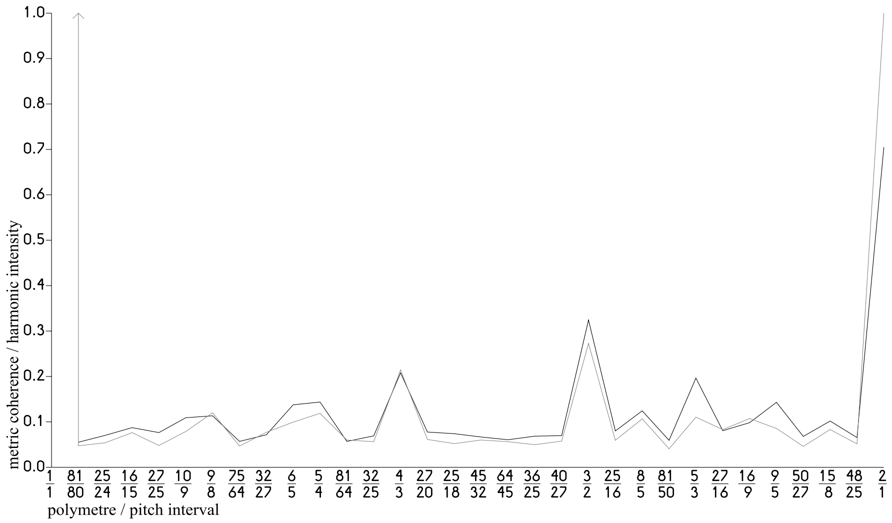
See also 5. and 7. in the previous diagramme

Γ33 – Graphs on Metric Coherence

a) 2×2×3-indispensability series, five times repeated, against a 3×5-Series, four times repeated; the printed size of the 'all-pulse counter' (1–60) reflects the squared product of the corresponding relative indispensabilities

all-pulse counter:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
2×2×3 pulses:	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8
Indispensability:	11	0	4	8	2	6	10	1	5	9	3	7	11	0	4	8	2	6	10	1
3×5 pulses:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	1	2	3	4	5
Indispensability:	14	0	9	3	6	12	1	10	4	7	13	2	11	5	8	14	0	9	3	6
all-pulse counter:	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
2×2×3 pulses:	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4
Indispensability:	5	9	3	7	11	0	4	8	2	6	10	1	5	9	3	7	11	0	4	8
3×5 pulses:	6	7	8	9	10	11	12	13	14	15	1	2	3	4	5	6	7	8	9	10
Indispensability:	12	1	10	4	7	13	2	11	5	8	14	0	9	3	6	12	1	10	4	7
all-pulse counter:	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
2×2×3 pulses:	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12
Indispensability:	2	6	10	1	5	9	3	7	11	0	4	8	2	6	10	1	5	9	3	7
3×5 pulses:	11	12	13	14	15	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Indispensability:	13	2	11	5	8	14	0	9	3	6	12	1	10	4	7	13	2	11	5	8

b) Graphic comparison of metric coherence of selected bar-tempo-ratios with the harmonic intensity (in grey) of the corresponding pitch intervals: (the metric stratification is derived from the prime decomposition of the tempo- ratio numbers in order of falling primes)



F34 – Formula for the Metric Coherence of two metres

Conventions:

- $\mathcal{M}$ is the Metric Coherence of metres 1 and 2
- m is the distinguishing number of the metres (= 1 or 2)
- u_m is the number of levels in the original metre m
- z_m is the number of levels in the expanded metre m
- v_m is the bar tempo of metre m
- q_{m_j} is the stratification divisor on the jth level of metre m
- Ω_j is the number of pulses in the stratification on level j
- Ω_0 is the number of pulses in the entire cycle

Based on the statements at left:

$$\Omega_{u_m} = \prod_{j=1}^{u_m} q_{m_j}$$

$$\Omega_{z_m} = \mathrm{kgV}(v_1\Omega_{u_1}, v_2\Omega_{u_2})$$

$$\Omega_0 = \mathrm{kgV}(\Omega_{z_1}, \Omega_{z_2})$$

The Formula:

$$\mathcal{M} = \frac{1}{-2\,\log_e\left[\dfrac{18\sum\limits_{n=1}^{\Omega_0}\left[\prod\limits_{i=1}^{2}\left\{\Psi_{z_i}\left(1+(n-1)\bmod \Omega_{z_i}\right)\right\}\right]^2-2}{7\Omega_0\prod\limits_{i=1}^{2}(\Omega_{z_i}-1)^2}\right]}$$

T35 – Tables on Metric Coherence

a) Intrametric Coherences of selected stratifications (S) upto 3rd order: bar-tempo (T) of each metre at left

Metre 1		Metre 2		Coherence
T	S	T	S	
1	2×2×2	1	2×2×2	0.46382
1	2×2×2	1	2×2×3	0.24638
1	2×2×2	1	2×3×2	0.18956
1	2×2×2	1	3×2×2	0.16747
1	2×2×3	1	2×2×2	0.24638
1	2×2×3	1	2×2×3	0.41454
1	2×2×3	1	2×3×2	0.26958
1	2×2×3	1	3×2×2	0.20797
1	2×3×2	1	2×2×2	0.18956
1	2×3×2	1	2×2×3	0.26958
1	2×3×2	1	2×3×2	0.41454
1	2×3×2	1	3×2×2	0.36421
1	3×2×2	1	2×2×2	0.16747
1	3×2×2	1	2×2×3	0.20797
1	3×2×2	1	2×3×2	0.36421
1	3×2×2	1	3×2×2	0.41454
1	2×2×2	2	2×2×2	0.42381
1	2×2×2	2	2×2×3	0.39233
1	2×2×2	2	2×3×2	0.25708
1	2×2×2	2	3×2×2	0.19808
1	2×2×3	2	2×2×2	0.18485
1	2×2×3	2	2×2×3	0.25708
1	2×2×3	2	2×3×2	0.39233
1	2×2×3	2	3×2×2	0.34635
1	2×3×2	2	2×2×2	0.16281
1	2×3×2	2	2×2×3	0.19808
1	2×3×2	2	2×3×2	0.34635
1	2×3×2	2	3×2×2	0.39233
1	3×2×2	2	2×2×2	0.15474
1	3×2×2	2	2×2×3	0.17378
1	3×2×2	2	2×3×2	0.32603
1	3×2×2	2	3×2×2	0.37737
1	2×2×2	3	2×2×2	0.17633
1	2×2×2	3	2×2×3	0.19094
1	2×2×2	3	2×3×2	0.13830
1	2×2×2	3	3×2×2	0.11984
1	2×2×3	3	2×2×2	0.33779
1	2×2×3	3	2×2×3	0.35407
1	2×2×3	3	2×3×2	0.20399
1	2×2×3	3	3×2×2	0.15166
1	2×3×2	3	2×2×2	0.39582
1	2×3×2	3	2×2×3	0.37899
1	2×3×2	3	2×3×2	0.24842
1	2×3×2	3	3×2×2	0.18120
1	3×2×2	3	2×2×2	0.41227
1	3×2×2	3	2×2×3	0.38555
1	3×2×2	3	2×3×2	0.25318
1	3×2×2	3	3×2×2	0.19492
2	2×2×2	1	2×2×2	0.42381
2	2×2×2	1	2×2×3	0.18485
2	2×2×2	1	2×3×2	0.16281
2	2×2×2	1	3×2×2	0.15474
2	2×2×3	1	2×2×2	0.39233
2	2×2×3	1	2×2×3	0.25708
2	2×2×3	1	2×3×2	0.19808
2	2×2×3	1	3×2×2	0.17378
2	2×3×2	1	2×2×2	0.25708
2	2×3×2	1	2×2×3	0.39233
2	2×3×2	1	2×3×2	0.34635
2	2×3×2	1	3×2×2	0.32603
2	3×2×2	1	2×2×2	0.19808
2	3×2×2	1	2×2×3	0.34635
2	3×2×2	1	2×3×2	0.39233
2	3×2×2	1	3×2×2	0.37737
2	2×2×2	3	2×2×2	0.16149
2	2×2×2	3	2×2×3	0.13297
2	2×2×2	3	2×3×2	0.11317
2	2×2×2	3	3×2×2	0.10609
2	2×2×3	3	2×2×2	0.18416
2	2×2×3	3	2×2×3	0.19395
2	2×2×3	3	2×3×2	0.14147
2	2×2×3	3	3×2×2	0.12338
2	2×3×2	3	2×2×2	0.35240
2	2×3×2	3	2×2×3	0.23679
2	2×3×2	3	2×3×2	0.17158
2	2×3×2	3	3×2×2	0.15084
2	3×2×2	3	2×2×2	0.41575
2	3×2×2	3	2×2×3	0.24132
2	3×2×2	3	2×3×2	0.18509
2	3×2×2	3	3×2×2	0.16105
3	2×2×2	1	2×2×2	0.17633
3	2×2×2	1	2×2×3	0.33779
3	2×2×2	1	2×3×2	0.39582
3	2×2×2	1	3×2×2	0.41227
3	2×2×3	1	2×2×2	0.19094
3	2×2×3	1	2×2×3	0.35407
3	2×2×3	1	2×3×2	0.37899
3	2×2×3	1	3×2×2	0.38555
3	2×3×2	1	2×2×2	0.13830
3	2×3×2	1	2×2×3	0.20399
3	2×3×2	1	2×3×2	0.24842
3	2×3×2	1	3×2×2	0.25318
3	3×2×2	1	2×2×2	0.11984
3	3×2×2	1	2×2×3	0.15166
3	3×2×2	1	2×3×2	0.18120
3	3×2×2	1	3×2×2	0.19492
3	2×2×2	2	2×2×2	0.16149
3	2×2×2	2	2×2×3	0.18416
3	2×2×2	2	2×3×2	0.35240
3	2×2×2	2	3×2×2	0.41575
3	2×2×3	2	2×2×2	0.13297
3	2×2×3	2	2×2×3	0.19395
3	2×2×3	2	2×3×2	0.23679
3	2×2×3	2	3×2×2	0.24132
3	2×3×2	2	2×2×2	0.11317
3	2×3×2	2	2×2×3	0.14147
3	2×3×2	2	2×3×2	0.17158
3	2×3×2	2	3×2×2	0.18509
3	3×2×2	2	2×2×2	0.10609
3	3×2×2	2	2×2×3	0.12338
3	3×2×2	2	2×3×2	0.15084
3	3×2×2	2	3×2×2	0.16105

. . . T35 . . .

b) Tabled comparison of metric coherence of selected bar-tempo-ratios with the harmonicity of the corresponding pitch intervals (see Γ33b).

Ratio	prime decomposition	metric coherence	harmonicity
1 / 1	2 / 2*	∞	∞
81 / 80	(3×3×3×3) / (5×2×2×2×2)	0.05538	0.04747
25 / 24	(5×5) / (3×2×2×2)	0.07005	0.05415
16 / 15	(2×2×2×2) / (5×3)	0.08712	0.07653
27 / 25	(3×3×3) / (5×5)	0.07607	0.04808
10 / 9	(5×2) / (3×3)	0.10889	0.07853
9 / 8	(3×3) / (2×2×2)	0.11304	0.12000
75 / 64	(5×5×3) / (2×2×2×2×2×2)	0.05688	0.04658
32 / 27	(2×2×2×2×2) / (3×3×3)	0.07138	0.07692
6 / 5	(3×2) / 5	0.13783	0.09934
5 / 4	5 / (2×2)	0.14370	0.11905
81 / 64	3×3×3×3) / (2×2×2×2×2×2)	0.05682	0.06000
32 / 25	(2×2×2×2×2) / (5×5)	0.06920	0.05618
4 / 3	(2×2) / 3	0.20797	0.21429
27 / 20	(3×3×3) / (5×2×2)	0.07750	0.06098
25 / 18	(5×5) / (3×3×2)	0.07424	0.05227
45 / 32	(5×3×3) / (2×2×2×2×2)	0.06668	0.05976
64 / 45	(2×2×2×2×2×2) / (5×3×3)	0.06082	0.05639
36 / 25	(3×3×2×2) / (5×5)	0.06834	0.04967
40 / 27	(5×2×2×2) / (3×3×3)	0.06952	0.05747
3 / 2	3 / 2	0.32447	0.27273
25 / 16	(5×5) / (2×2×2×2)	0.08001	0.05952
8 / 5	(2×2×2) / 5	0.12442	0.10639
81 / 50	(3×3×3×3) / (5×5×2)	0.05961	0.04087
5 / 3	5 / 3	0.19613	0.11029
27 / 16	(3×3×3) / (2×2×2×2)	0.08047	0.08333
16 / 9	(2×2×2×2) / (3×3)	0.09837	0.10714
9 / 5	(3×3) / 5	0.14284	0.08523
50 / 27	(5×5×2) / (3×3×3)	0.06810	0.04587
15 / 8	(5×3) / (2×2×2)	0.10209	0.08287
48 / 25	(3×2×2×2×2) / (5×5)	0.06562	0.05137
2 / 1	(2×2) / 2*	0.70507	1.00000

* set at 2 / 2 and to (2×2)/2 to avoid a one-beat metre

T36 – On Stochastics

a) Statistical list of all characters and bigrammes in Chapter 24 with occurrence tally (‘ _’ represents a space)

Characters	50 × AL	22 × OT	12 × MO	7 × KE	4 × EY	2 × AK
926 × _	47 × T_	22 × LL	12 × IL	7 × IV	4 × EP	2 × _K
565 × E	46 × Y_	22 × HI	12 × E	7 × IE	4 × EF	1 × ZE
404 × T	46 × TO	22 × FR	12 × EM	7 × IB	4 × CS	1 × YM
332 × O	46 × TI	22 × _G	12 × BA	7 × CR	4 × CI	1 × YI
311 × I	46 × ON	21 × ME	12 × AM	7 × AV	4 × AG	1 × YE
310 × S	45 × F_	21 × IO	11 × YN	6 × UN	3 × YS	1 × YA
302 × A	43 × OF	21 × _N	11 × UR	6 × SC	3 × W_	1 × H
294 × R	43 × AN	20 × RI	11 × UE	6 × PH	3 × UP	1 × C
270 × N	43 × _C	20 × HA	11 × OS	6 × IM	3 × TC	1 × UD
239 × H	41 × NG	20 × EA	11 × MU	6 × _U	3 × SS	1 × TM
177 × L	41 × _F	20 × CA	11 × EE	5 × T	3 × RM	1 × TL
145 × D	40 × O_	19 × VE	11 × C_	5 × UM	3 × PT	1 × SM
139 × C	39 × AT	19 × HO	11 × AP	5 × TU	3 × OP	1 × SL
103 × M	38 × TE	19 × FO	11 × AI	5 × P_	3 × NY	1 × SK
98 × F	38 × NT	19 × ET	10 × WS	5 × OD	3 × NL	1 × SF
88 × U	38 × L_	19 × AB	10 × TT	5 × NU	3 × K_	1 × RV
86 × G	37 × ST	18 × OW	10 × MA	5 × NI	3 × HT	1 × RU
83 × B	37 × RA	17 × TA	10 × GI	5 × LD	3 × HS	1 × RP
65 × P	37 × DE	17 × SH	10 × AD	5 × IR	3 × FU	1 × RL
64 × Y	37 × _B	17 × RS	9 × UT	5 × GU	3 × FA	1 × PU
52 × W	36 × ED	17 × OU	9 × UL	5 × GH	3 × EC	1 × OI
31 × V	35 × LE	17 × NO	9 × QU	5 × FT	3 × CY	1 × OG
15 × X	33 × SE	17 × LY	9 × PO	5 × EI	3 × CL	1 × OE
12 × K	33 × IC	17 × _D	9 × PL	5 × DO	3 × BU	1 × NV
9 × Q	33 × BE	16 × NE	9 × OL	5 × CU	3 × _V	1 × NN
5 × Z	33 × _L	16 × M_	9 × OB	5 × BY	2 × I	1 × NF
Bigrammes	32 × ND	16 × LO	9 × EQ	5 × BL	2 × F	1 × LR
189 × E_	32 × IT	16 × IG	9 × DU	5 × _Z	2 × WN	1 × LC
168 × _T	31 × _M	16 × GR	9 × CC	4 × ZO	2 × UI	1 × HU
156 × TH	30 × SI	15 × US	8 × VI	4 × A	2 × SU	1 × HF
148 × S_	30 × _W	15 × NA	8 × TY	4 × WO	2 × RC	1 × GT
139 × HE	29 × RD	15 × GE	8 × TR	4 × VA	2 × PA	1 × GN
104 × _A	29 × H_	15 × BI	8 × PE	4 × UC	2 × OY	1 × GM
87 × _O	28 × RO	14 × TS	8 × OC	4 × TW	2 × O	1 × FI
86 × ER	28 × _E	14 × PR	8 × MB	4 × SP	2 × OK	1 × EO
85 × IN	27 × CH	14 × NS	8 × LA	4 × RT	2 × ML	1 × EK
77 × RE	27 × _R	14 × CE	8 × ID	4 × PP	2 × LF	1 × EH
69 × ES	26 × G_	13 × WH	8 × EV	4 × PI	2 × KI	1 × DT
64 × N_	26 × AR	13 × WE	8 × CT	4 × OV	2 × IF	1 × BR
64 × D_	25 × AS	13 × NC	8 × BO	4 × OO	2 × HR	1 × BB
64 × _I	25 × A_	13 × EL	7 × WI	4 × MI	2 × GA	1 × AY
62 × IS	24 × LI	13 × AC	7 × SO	4 × LU	2 × EG	1 × A
61 × OR	24 × CO	13 × _H	7 × SA	4 × LS	2 × EB	1 × AU
59 × R_	23 × OM	12 × UA	7 × RY	4 × IK	2 × DY	1 × AO
54 × EN	23 × DI	12 × SY	7 × RR	4 × IA	2 × DS	1 × AF
54 × _S	23 × _P	12 × MP	7 × MM	4 × GL	2 × DD	

...T36...

b) Examples of Markov Syntheses of various orders

Original Text:

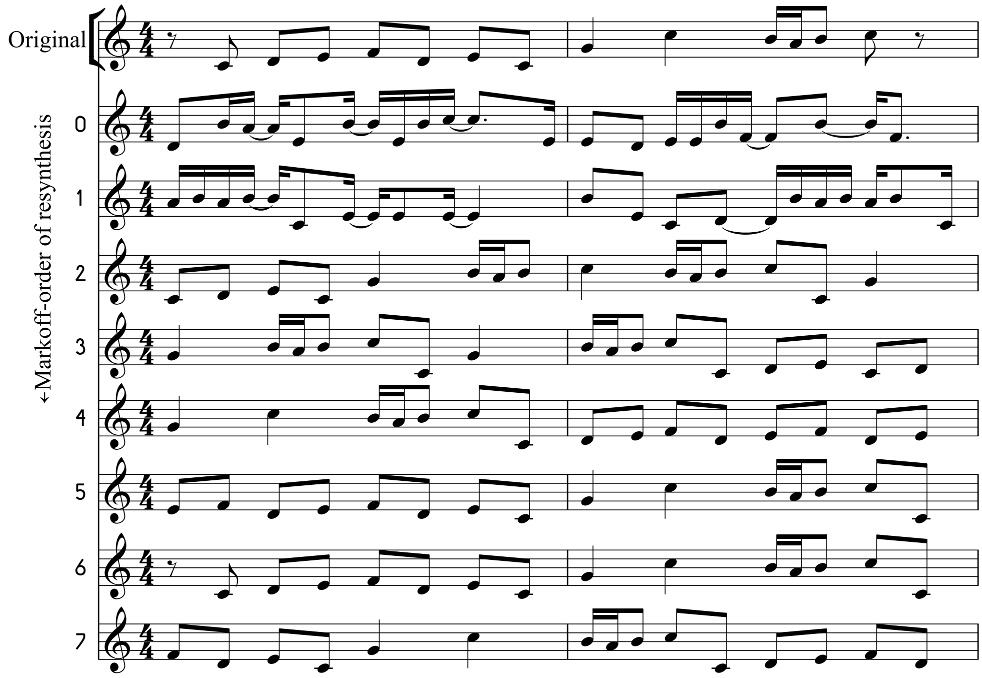
```
AT THE START OF THE TWENTIETH CENTURY, THE RUSSIAN MATHEMATICIAN
ANDREI ANDREYEVITCH MARKOV DEVELOPED A METHOD
```

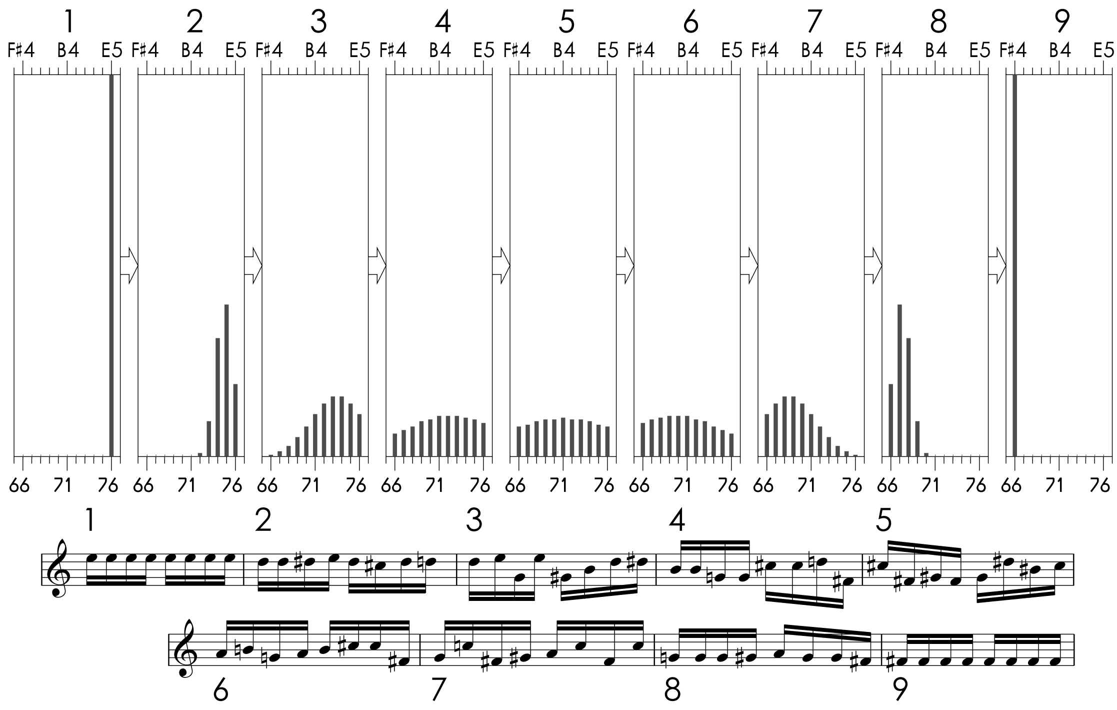
Markovsyntheses:

Order

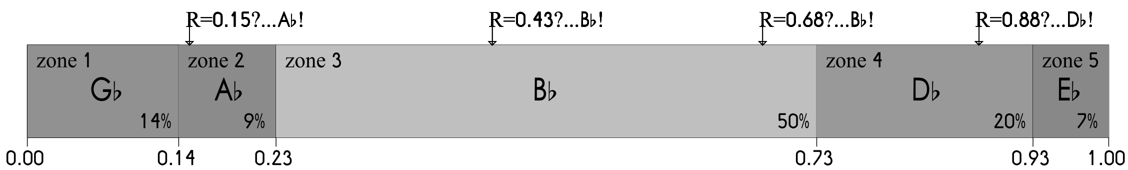
```
0:    RYHAEDERINKTCRT NITTDDT OHEIDADTHOVHR OTHSUMIRTSTTRAAVE EOATSSIE
      TH E ORDSLCRDPRTODSUTOYIKT  DTEFMATRE MAOTHA TINAAEELCTNRCHANOARHO
      T EEETI FHAN  RACH  TAHNMA ITEE OEC,RTE LTI KOT TAH

1:    IELOD ATUSTICIANDRE MATHENTHELOPE STHEV CENT D AN THEDEMAN SSTHE
      ARKOF D ANDREITWEMA MAND DEN TH ANDREVE SIAND ARETIEVEY,
      ATIETURURY, TH MARYELOF AN DE TUSIAN MANDRETHEIE CH STI TH

2:    T TWENTIETHOD ATIETH CENTURY, THE STARKOV DEVITCH MATHE START THE
      THE START TH METHE STARKOV DEVELOPED ANDREYEVELOPED ANDREI A
      MATICIANDREI AN MAT THEMATHE START OF TH CENTURY, THE STARKOV

3:    MATHE TWENTURY, THE RUSSIAN ANDREI ANDREYEVITCH MARKOV DEVELOPED A
      METHOD AT THE RUSSIAN MATICIAN ANDREI ANDREYEVITCH MARKOV
      DEVELOPED A METH CENTURY, THE START OF THE START OF THEMATHE

4:    VELOPED A METHOD AT THE START OF THE RUSSIAN ANDREI ANDREI ANDREI
      ANDREYEVITCH MARKOV DEVELOPED A METHOD AT THE START OF THE RUSSIAN
      MATHEMATICIAN ANDREYEVITCH MARKOV DEVELOPED A METHOD AT THE

5:    THOD AT THE TWENTIETH CENTURY, THE RUSSIAN MATHEMATICIAN ANDREI
      ANDREYEVITCH MARKOV DEVELOPED A METHOD AT THE TWENTIETH CENTURY,
      THE TWENTIETH CENTURY, THE START OF THE TWENTIETH CENTURY, THE

6:    START OF THE TWENTIETH CENTURY, THE RUSSIAN MATHEMATICIAN ANDREI
      ANDREYEVITCH MARKOV DEVELOPED A METHOD AT THE START OF THE
      TWENTIETH CENTURY, THE RUSSIAN MATHEMATICIAN ANDREI ANDREI ANDREI

7:    NTURY, THE RUSSIAN MATHEMATICIAN ANDREI ANDREYEVITCH MARKOV
      DEVELOPED A METHOD AT THE START OF THE TWENTIETH CENTURY, THE
      RUSSIAN MATHEMATICIAN ANDREI ANDREYEVITCH MARKOV DEVELOPED A

1.2:  D ANTHE METHENDREY, TCH CITH STHE MAT TH SIELOF STWE THELOV T ATHE
      THETHELOD TICE MA OVIATWETITCHELOPEVIANTHEMATHETHOD A TARTH
      METATIET SSTH OF MANTWEITURKOD AN TUSSSIATI RENTICICIANTURT

1.4:  ARY, THE MELOPEYEDREYEN MATITHE RUSSTIETH AT TH AN TCH
      OPELOPEMETHE STANTUREY, AT OF TITHE RUSTHEY, TH MATH
      RURTUSSSIATITH TH AN THENTH CE RUSSIATCHOVELOPEI AN ARKOVEY,

1.6:  IT MENTURYEV THODREV MATHEMARKOD ANDREVI ANTIETHE STARTH MAN ATHE
      ST OF RUSSIANTHEMETHE THOVELOVIT DRUSSICEMATHE RKOD ATWENTH CHE
      THEMAREYEVELOPEDRE STAN DEVITH TWELOV DEVIANDEVIAN T OV AN

1.8:  IAN MARKOVETHOD AT TH CENTICIANTICH MATHEY, THE RUSSSIAN CEMAT THE
      SIANDREYEVELOPED AND A MAT AN ANDREYEVELOPED AN MA METH MARKOV
      DEVELOPED AN MARKOD ATHE METH MART T OF THEMART OF THOD
```

Γ37 – Examples of the stochastic generation of notes

a) Markov resyntheses (orders 1-8) of a J.S.Bach-Phrase

b) nine-stage conversion of probabilities into pitches – above: probability (y, dark grey) vs. MIDI-pitch (x), below: resultant pitches for each of the 9 stages

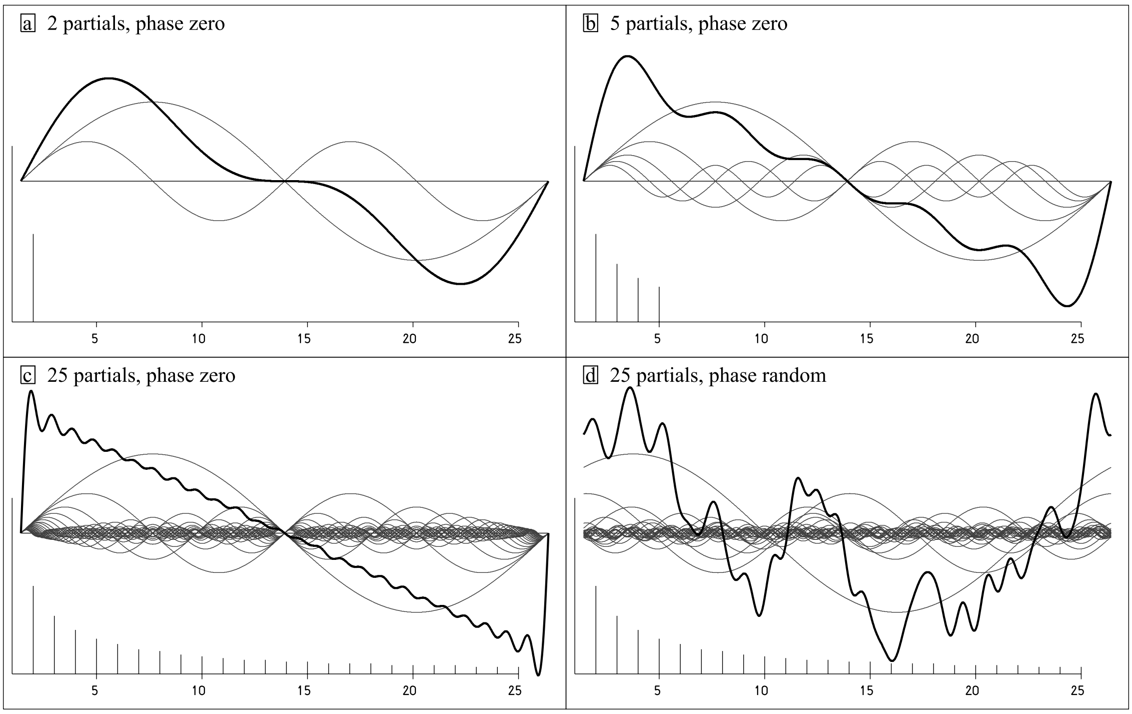


c) Method for converting probability (%) into notes by random numbers, of which here four (R=..) are employed for a set of five pitches and their probabilities

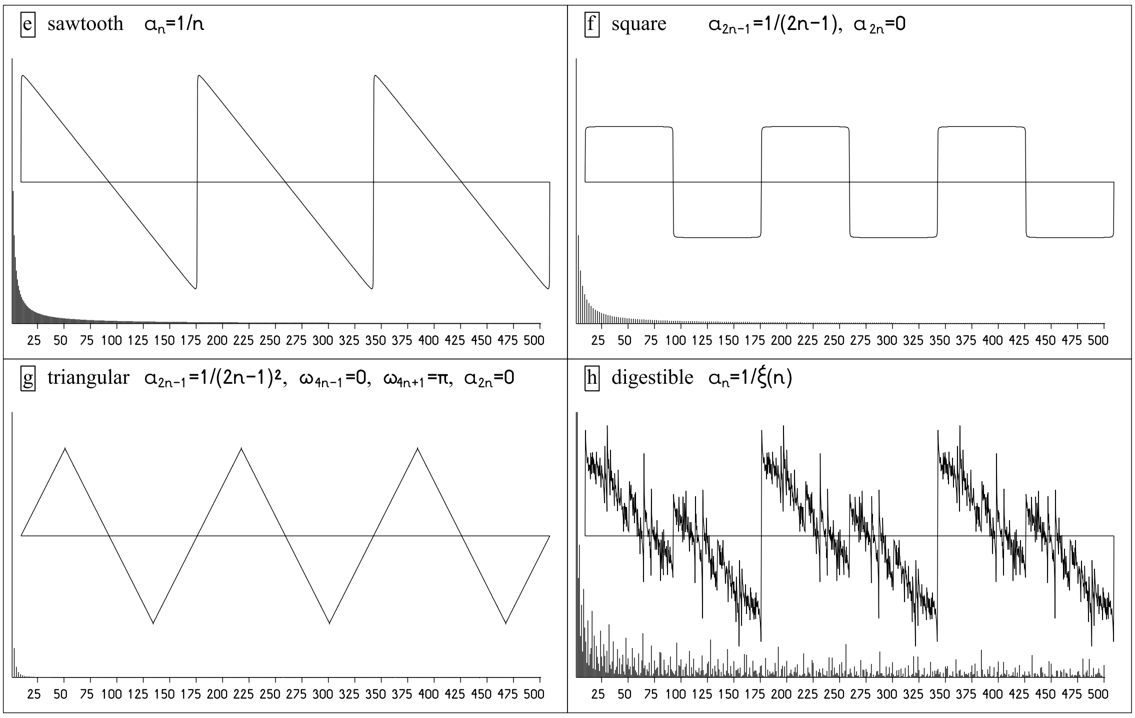


Γ38 – Examples of Fourier analysis and -synthesis

a-d) Wave synthesis of a sawtooth sprectrum

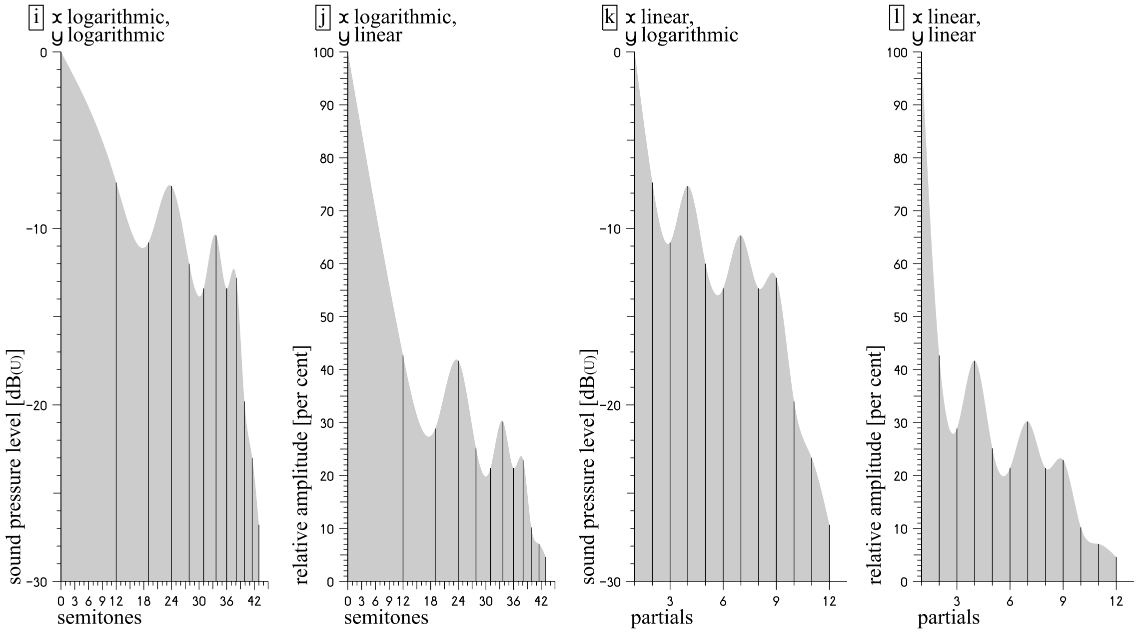


e-h) Wave-form syntheses mit 500 partials (a_t = amplitude, ω = phase, t = partial)

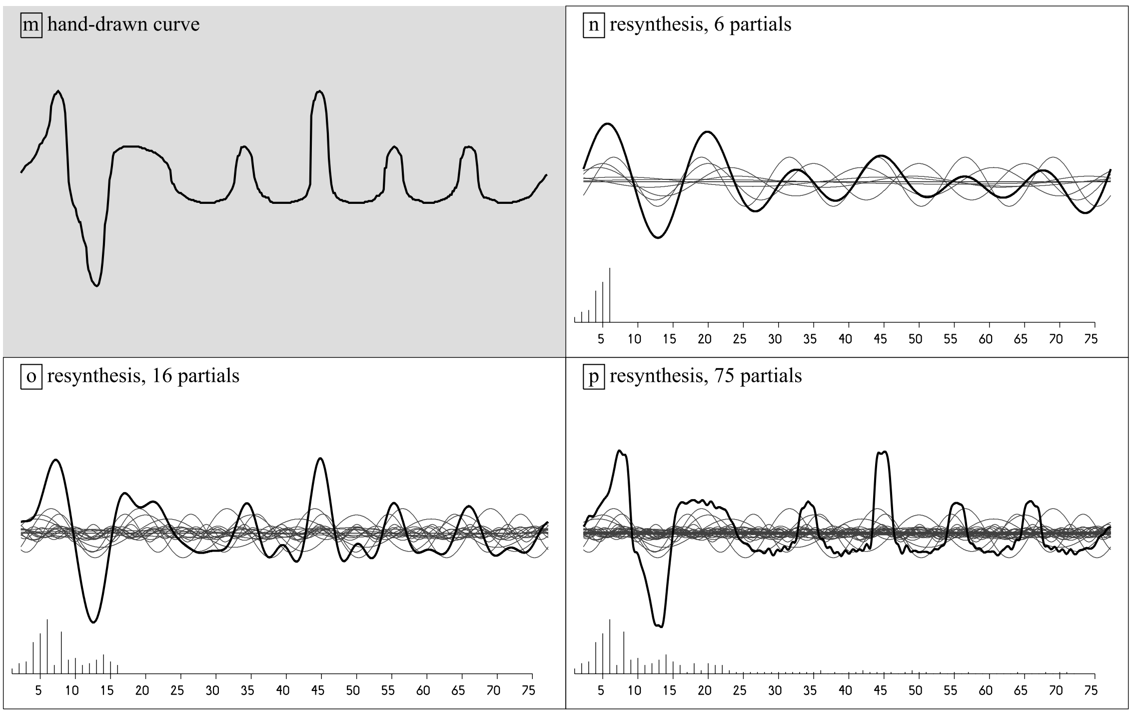


...Γ38...

i-l) Bassoon spectrum in four renditions
(pitch and loudness alternately linear and logarithmic)



m-p) hand-drawn curve with three Fourier-resynthesized approximations

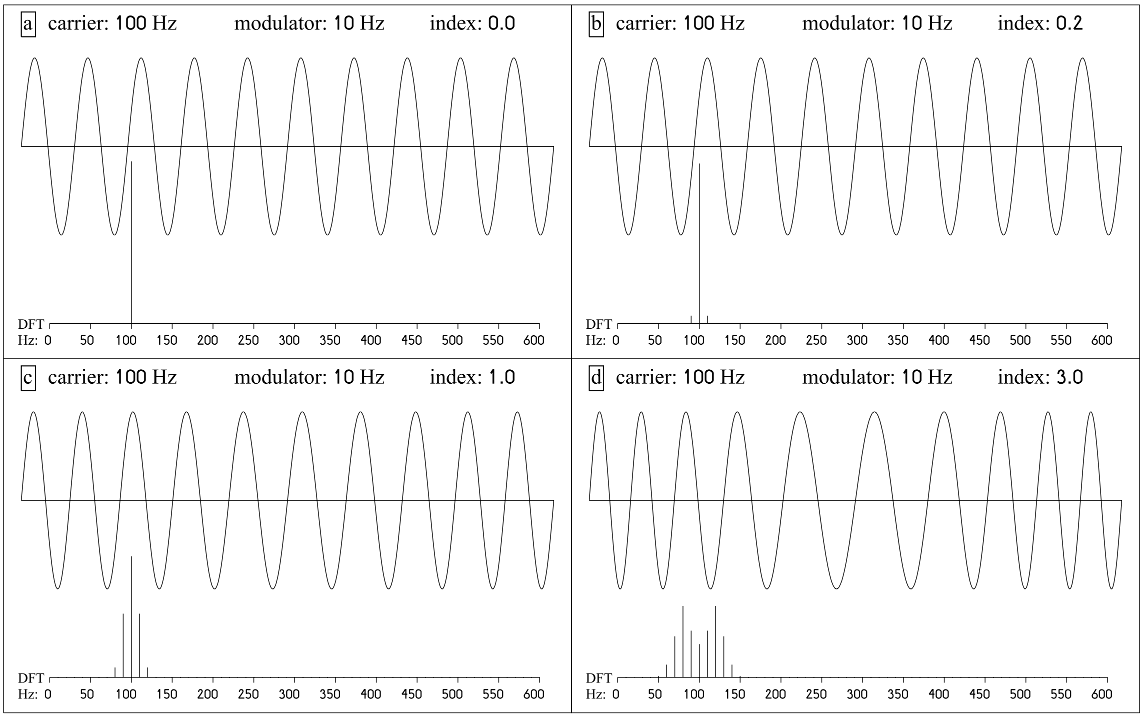


T39 – Fourier Analysis (DFT) of a hand-drawn curve

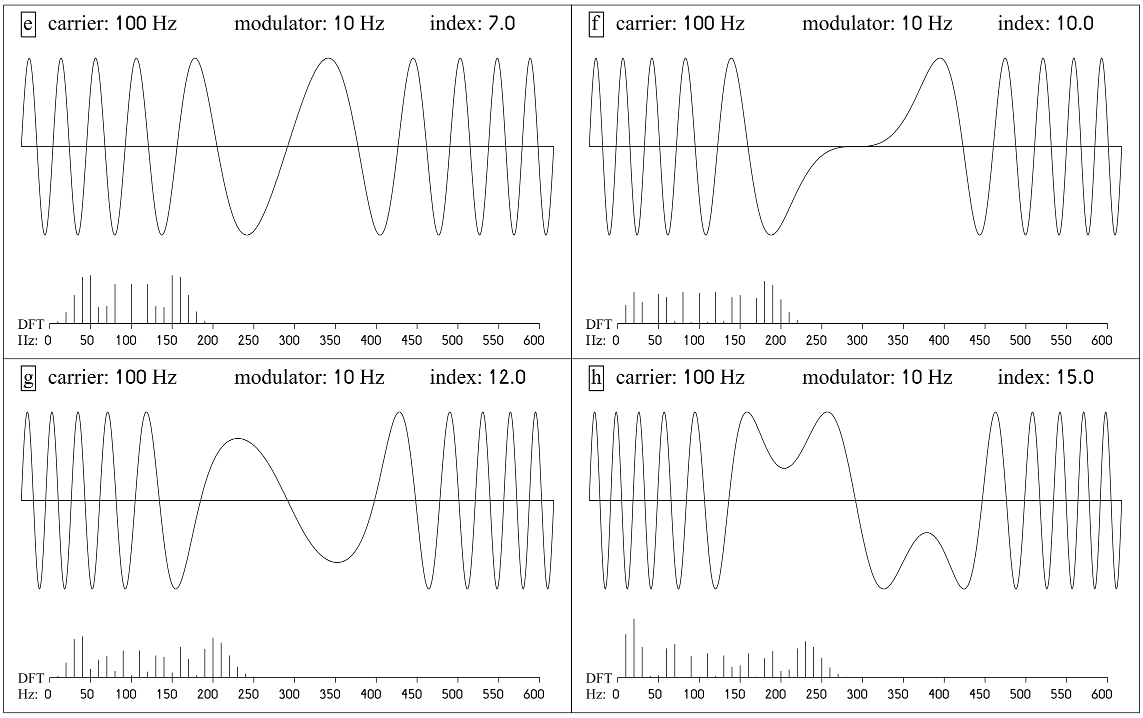
Partial	Amplitude /100	Phase /1000	Partial	Amplitude /100	Phase /1000	Partial	Amplitude /100	Phase /1000
1	3	61	34	1	456	67	0	0
2	6	107	35	1	947	68	1	469
3	7	374	36	2	240	69	0	0
4	18	138	37	0	0	70	1	524
5	23	68	38	1	79	71	1	986
6	31	906	39	0	0	72	0	0
7	5	260	40	1	925	73	0	0
8	24	670	41	1	616	74	0	0
9	8	535	42	2	896	75	0	0
10	9	510	43	1	336	76	0	0
11	5	234	44	1	683	77	1	640
12	6	236	45	1	389	78	0	0
13	8	935	46	1	550	79	1	438
14	11	326	47	0	0	80	0	0
15	7	652	48	1	494	81	1	325
16	5	206	49	2	47	82	0	0
17	1	44	50	1	298	83	0	0
18	6	11	51	1	917	84	0	0
19	2	927	52	0	0	85	0	0
20	6	868	53	1	739	86	0	0
21	5	445	54	0	0	87	0	0
22	5	651	55	1	628	88	0	0
23	2	357	56	0	0	89	0	0
24	1	499	57	1	432	90	0	0
25	1	163	58	0	0	91	0	0
26	1	310	59	0	0	92	0	0
27	1	34	60	0	0	93	0	0
28	1	330	61	0	0	94	0	0
29	1	16	62	0	0	95	0	0
30	1	972	63	0	0	96	0	0
31	1	799	64	1	896	97	0	0
32	1	642	65	0	0	98	0	0
33	1	733	66	0	0	99	0	0

Γ40 – Examples of Frequency Modulation and Phase Distortion

a-d) Modulator frequency < carrier frequency; index 0 to 3

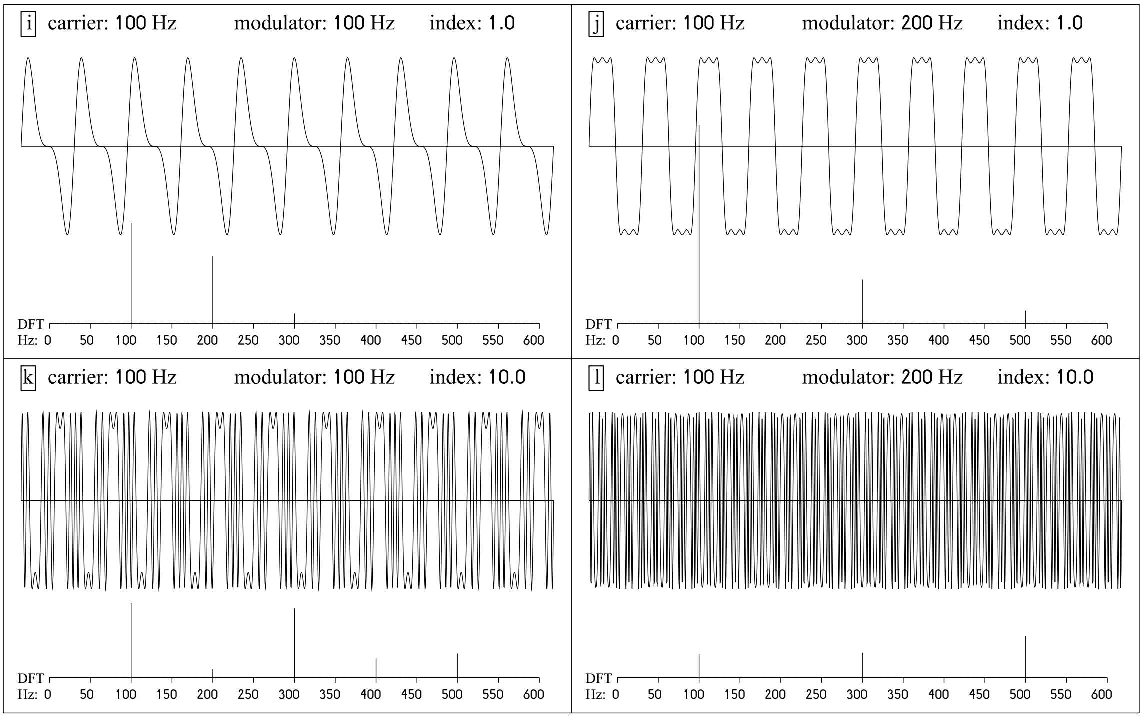


e-h) Modulator frequency < carrier frequency; index 7 to 15

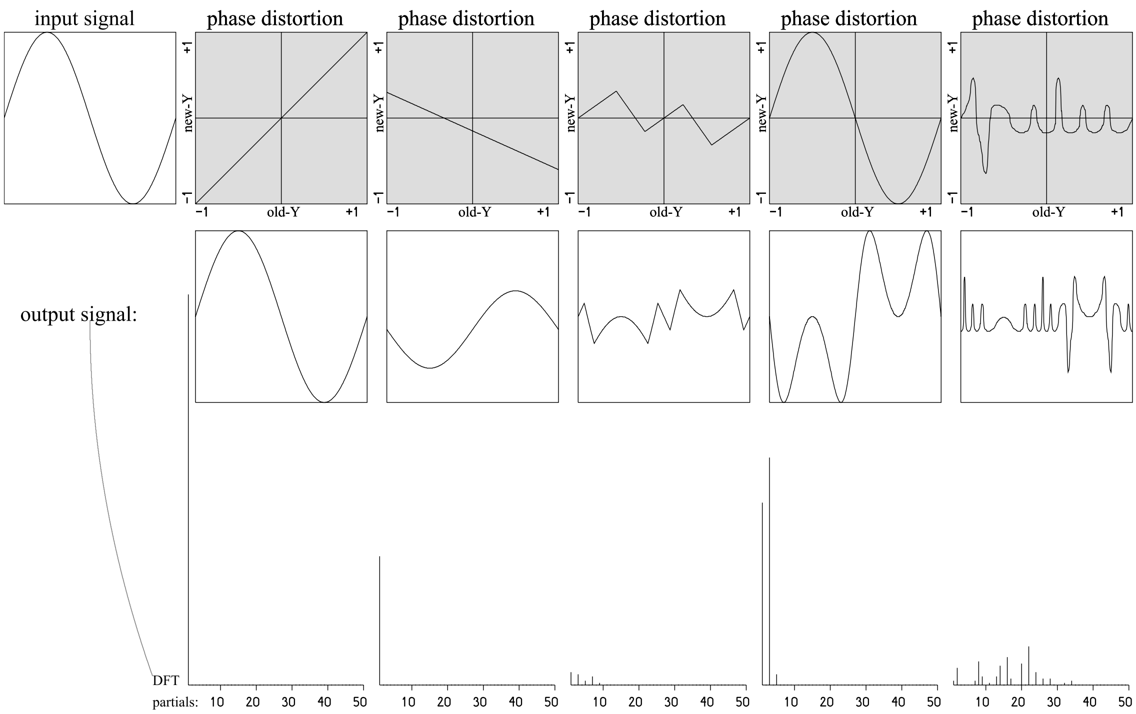


...Γ40...

i-l) Modulator frequency ≥ carrier frequency; index 1 and 10

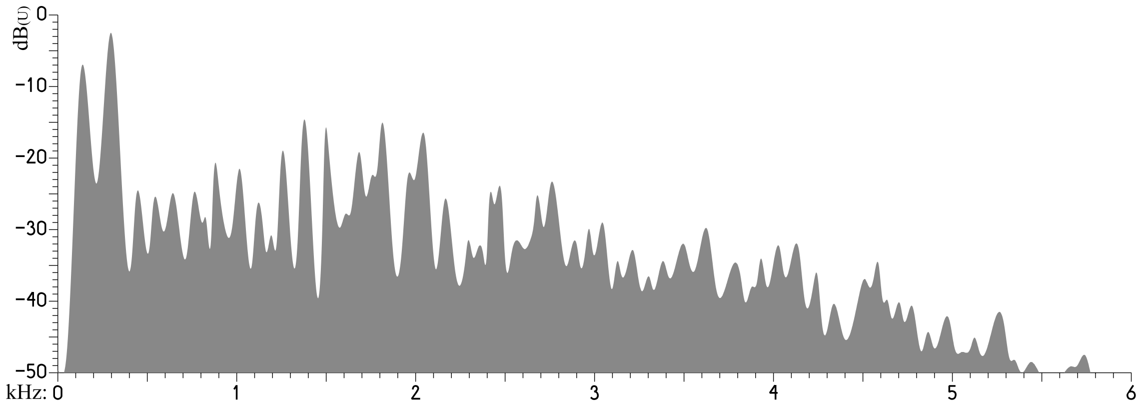


m) Examples of phase distortion (with sine input and output signals and DFTs)

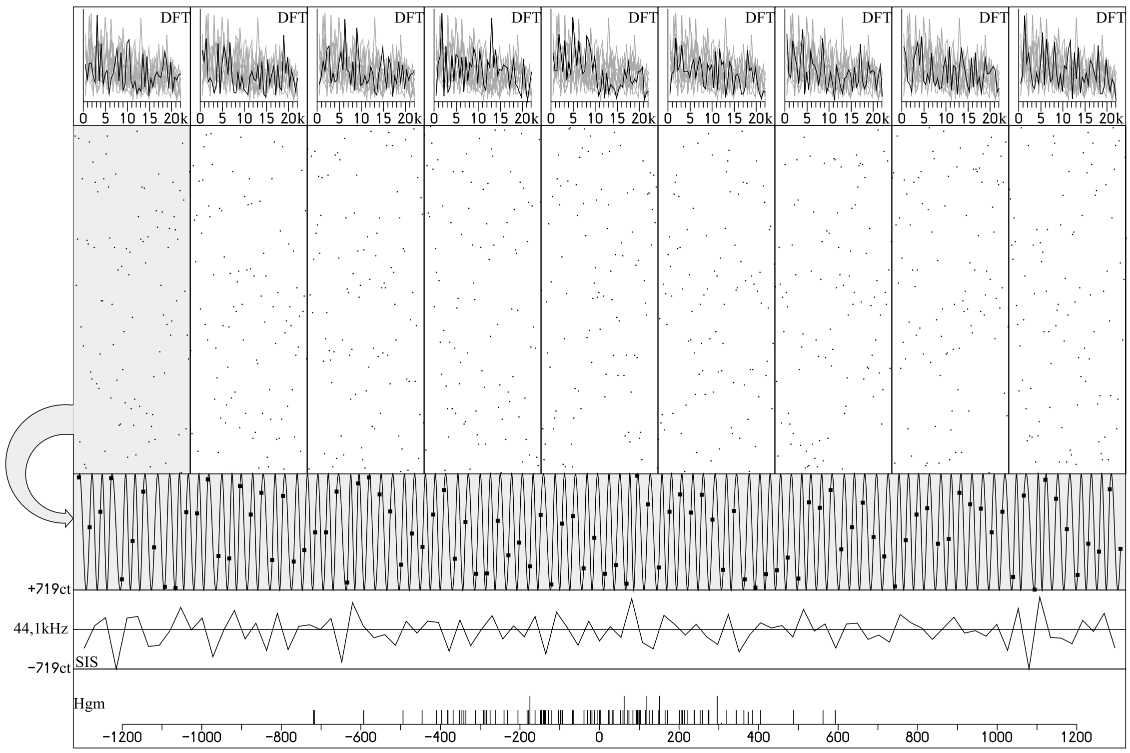


Γ41 – On complex tones and noise

a) A noise spectrum – a forcefully struck 70 cm Ø tamtam (92dB(SPL))

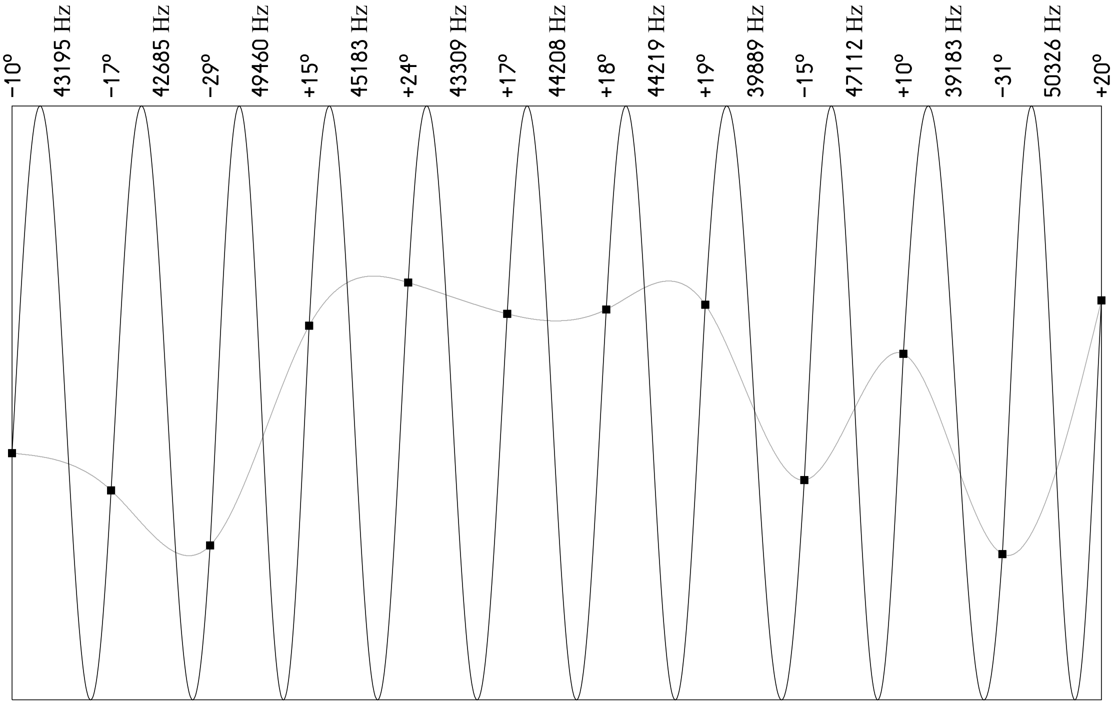


b) Random numbers as samples in the nine grey boxes (white noise – central diagramme from left to right) with ISIS-Analysis (SIS) and -histogramme (Hgm) at the bottom as well as Fourier-analysis of the samples (DFT, above)

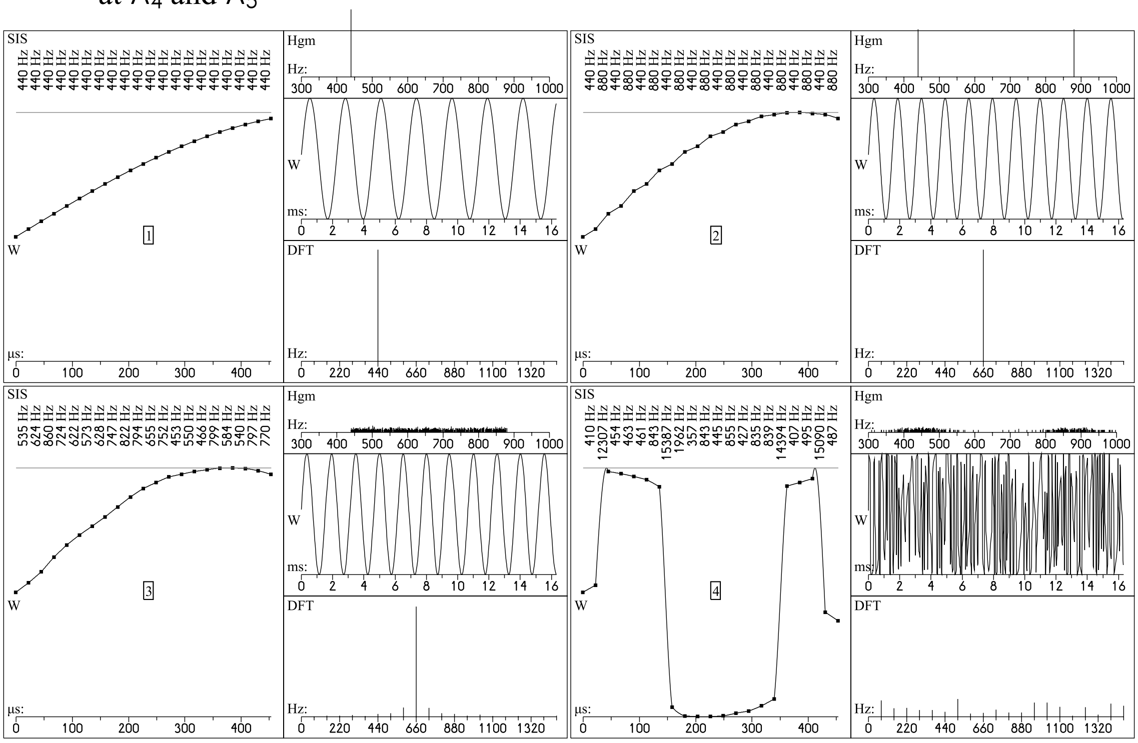


...Γ41...

c) ISIS-Analysis of 12 random samples: 11 sine frequencies with contigually connecting phases (a spline connects the samples for purely optical reasons as in some sound editors – s. also Γ02g)

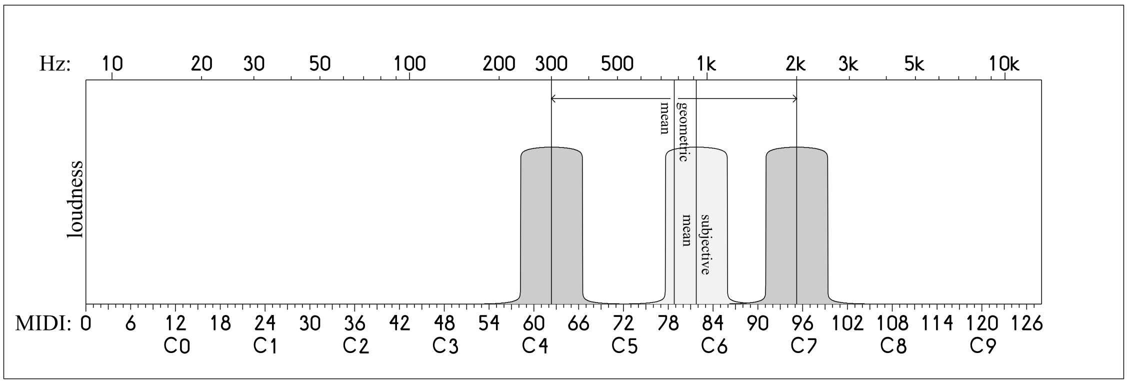


d) Four ISIS-syntheses (with DFTs and histogrammes) derived from
1) the pitch A_4, 2) an octave-tremolo A_4 - A_5, 3) randomly distributed frequencies between A_4 and A_5, 4) random frequencies with probability maxima at A_4 and A_5

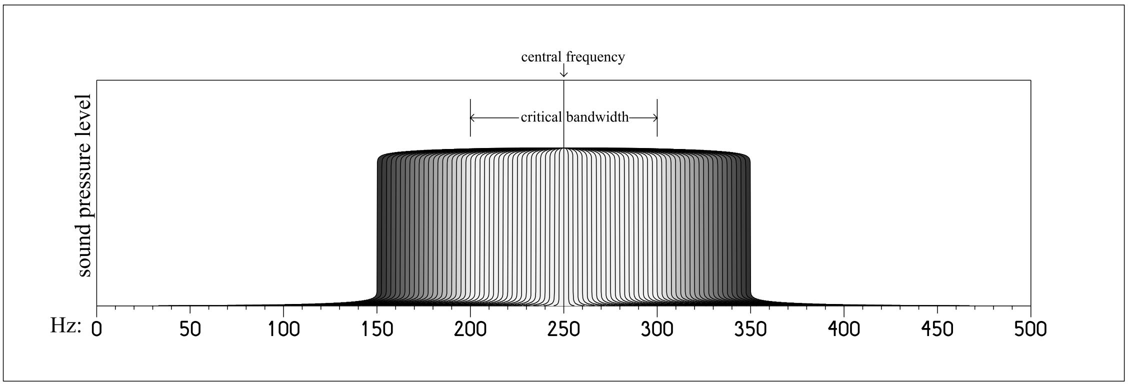


Γ42 – The Critical Bandwidth

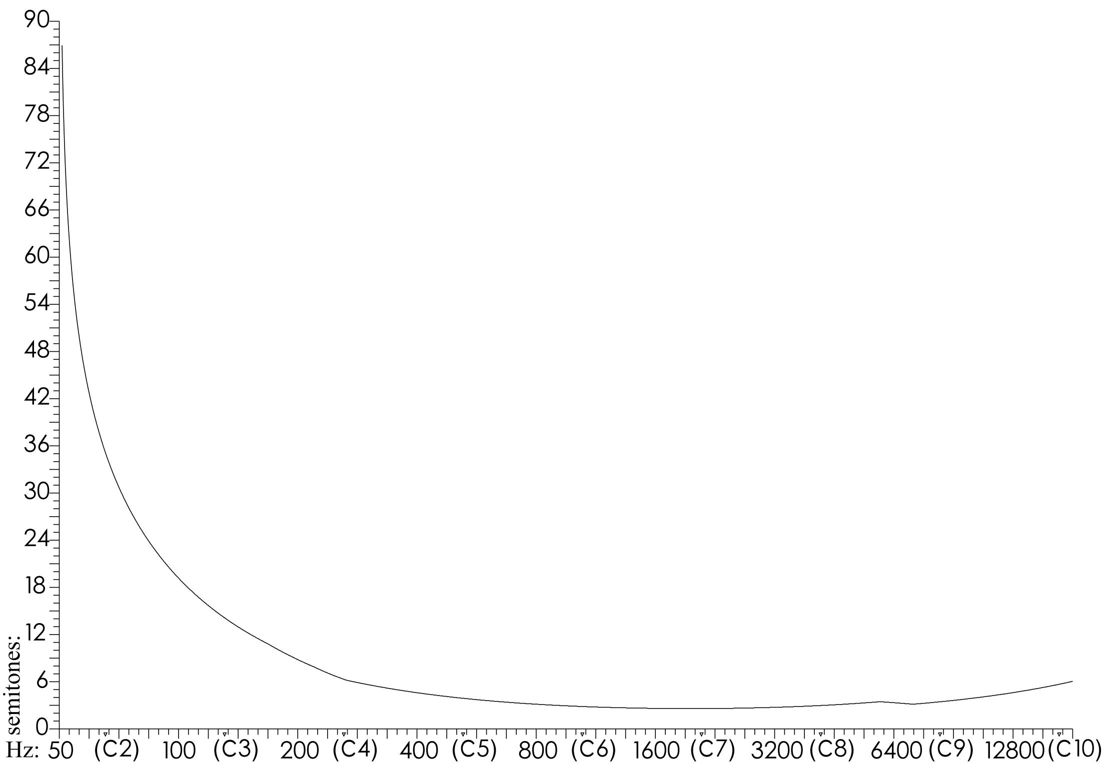
a) The geometric centre of two noise bands compared to their subjective centre



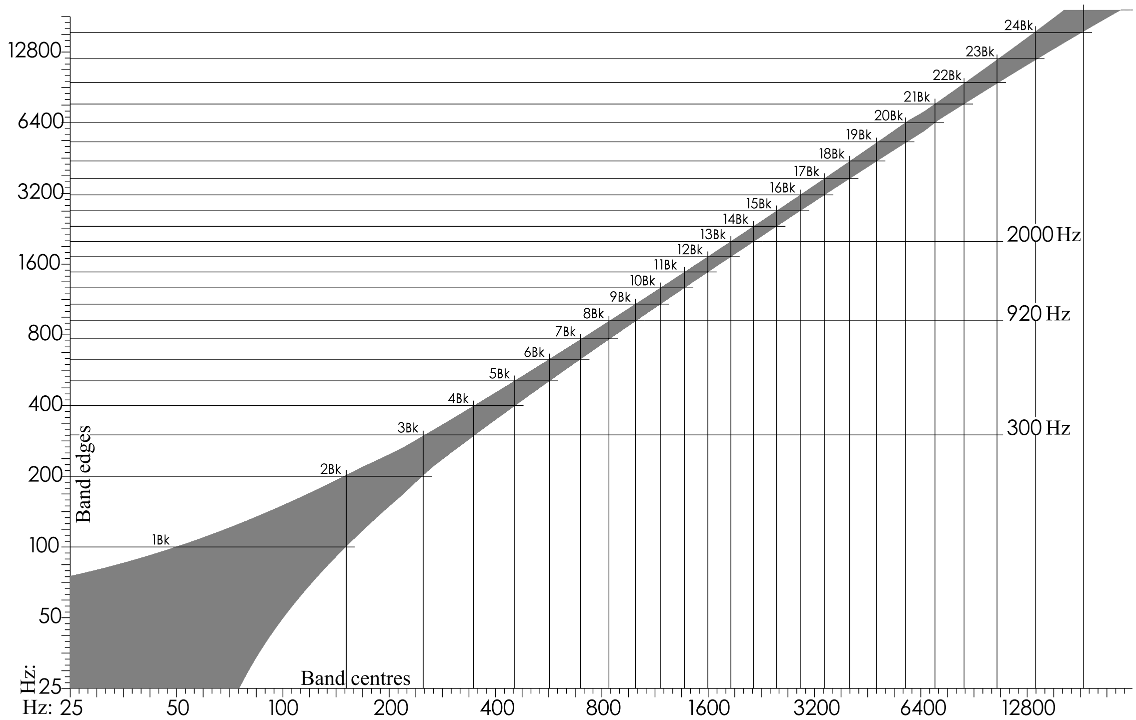
b) Loudness change (in grey shades) at constant SPL of a widening noise band



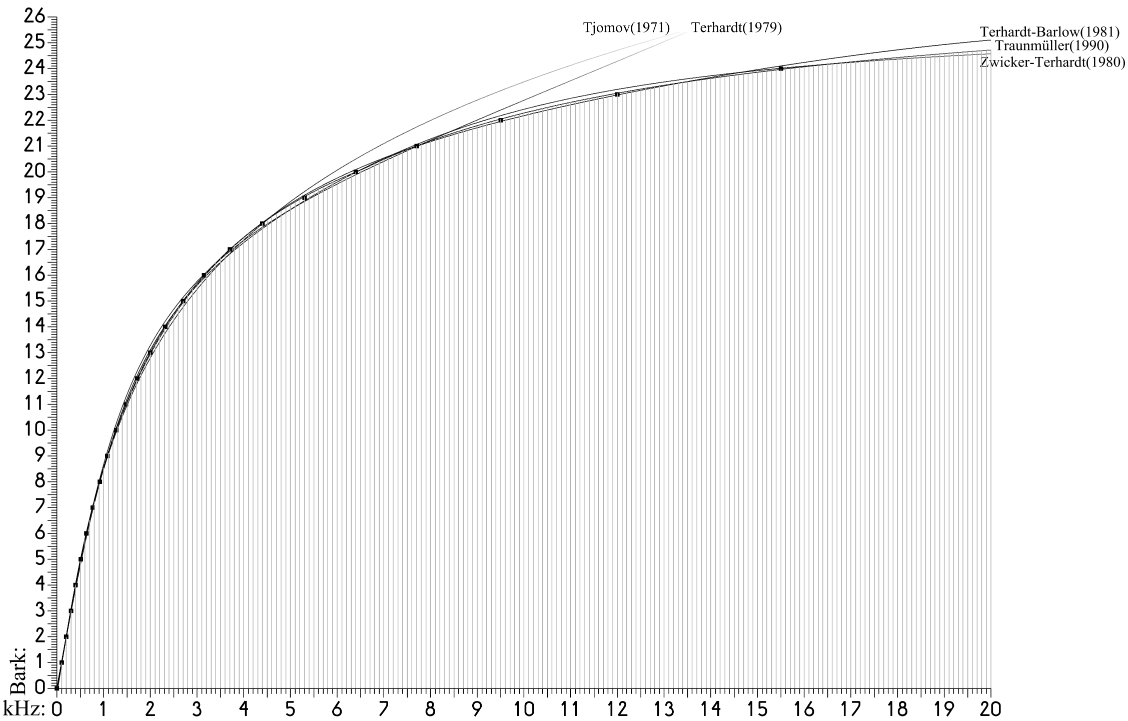
c) Critical Bandwidth in semitones (y) as an empirical function of the central band frequency (x)



d) Critical Band limits (y in Hz) as related to the central frequencies (x)



e) Critical Bandwidth in Bark (y) as algebraic approximations of band delimiting frequencies (x): the vertical grey lines mark 200 partials of a 100 Hz spectrum

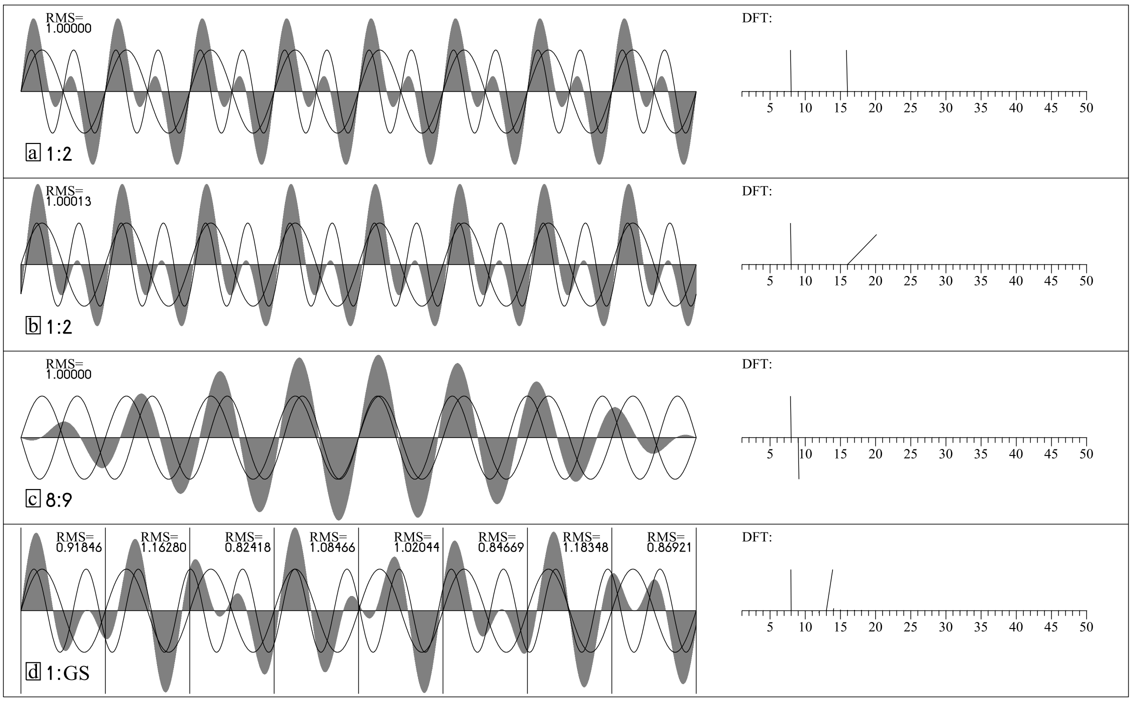


T43 – Conversion Bark→Hertz
according to the formulæ of Terhardt [1979] (for Bark ≤ 2.19)
and Traunmüller [1990] (for Bark ≥ 2.19)

Bark→	+0.0	+0.1	+0.2	+0.3	+0.4	+0.5	+0.6	+0.7	+0.8	+0.9
↓0:	0.000	10.03	20.05	30.08	40.11	50.15	60.19	70.24	80.30	90.36
1:	100.4	110.5	120.6	130.7	140.9	151.0	161.2	171.4	181.6	191.8
2:	202.0	212.3	222.2	231.3	240.5	249.7	259.1	268.5	278.0	287.5
3:	297.2	306.9	316.8	326.7	336.7	346.7	356.9	367.2	377.5	388.0
4:	398.5	409.1	419.9	430.7	441.6	452.7	463.8	475.0	486.4	497.8
5:	509.3	521.0	532.8	544.7	556.6	568.8	581.0	593.3	605.8	618.4
6:	631.1	643.9	656.9	670.0	683.2	696.6	710.1	723.7	737.5	751.4
7:	765.5	779.7	794.1	808.6	823.2	838.1	853.0	868.2	883.5	899.0
8:	914.6	930.4	946.4	962.6	978.9	995.4	1012	1029	1046	1063
9:	1081	1099	1117	1135	1153	1172	1190	1209	1229	1248
10:	1268	1288	1308	1328	1349	1370	1391	1413	1435	1457
11:	1479	1502	1525	1548	1571	1595	1620	1644	1669	1694
12:	1720	1746	1772	1799	1826	1853	1881	1909	1938	1967
13:	1997	2027	2057	2088	2120	2152	2184	2217	2251	2285
14:	2319	2354	2390	2426	2463	2501	2539	2578	2617	2658
15:	2698	2740	2783	2826	2870	2915	2960	3007	3054	3102
16:	3152	3202	3253	3305	3359	3413	3468	3525	3583	3642
17:	3702	3764	3827	3892	3958	4025	4094	4164	4237	4311
18:	4386	4464	4543	4625	4708	4794	4882	4972	5065	5160
19:	5258	5359	5462	5568	5678	5790	5906	6026	6149	6276
20:	6407	6543	6657	6775	6895	7019	7147	7278	7413	7552
21:	7695	7843	7995	8152	8314	8482	8655	8833	9018	9209
22:	9407	9613	9825	10046	10275	10513	10761	11018	11287	11566
23:	11858	12162	12480	12813	13161	13527	13910	14313	14737	15183
24:	15654	16152	16678	17236	17829	18459	19131	19848	20616	21441

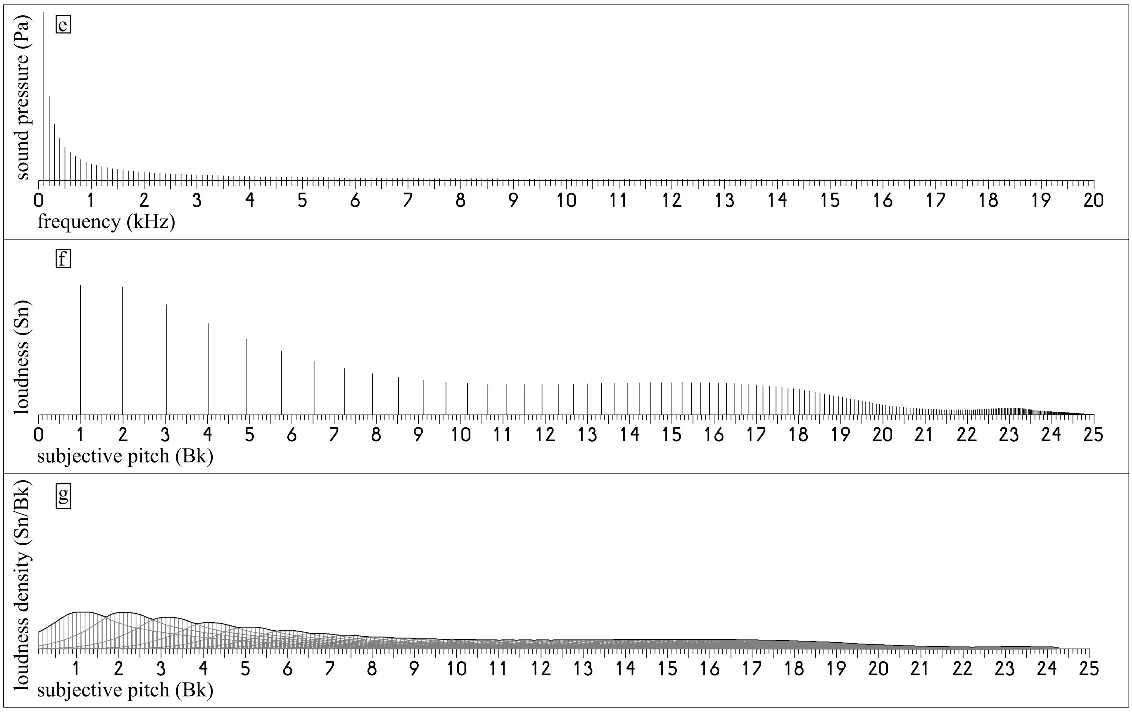
Γ44 – Summation (grey) of two sine curves, with RMS value and DFT spectrum (slant=phase):

a) Octave-related curves of equal phase
b) Octave-related curves with the shorter period's phase shifted by one-eighth
c) Fluctuating amplitude through frequency ratio 8:9 with opposite initial phase
d) Curves of equal initial phase in the frequency ratio of the Golden Section (1:1.618034)



Spectra of a 100-Hz sawtooth tone:

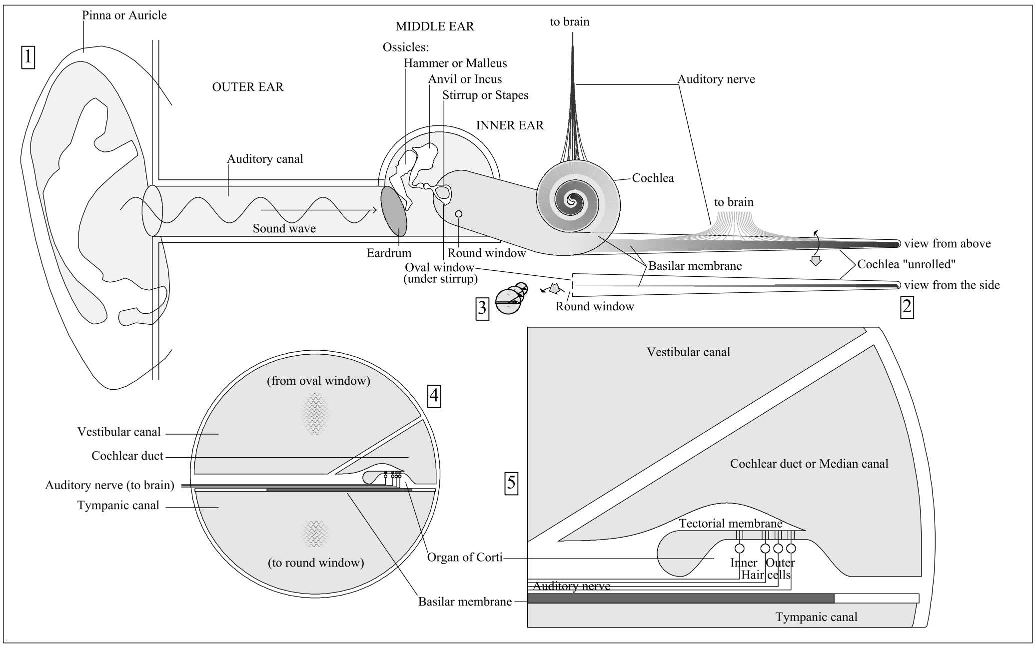
e) Physical, in sound pressure (y in Pascal) vs. frequency (x in Kilohertz)
f) Subjective, in loudness (y in Sone) vs. subjective pitch (x in Bark)
g) Subjective, in loudness density (y in Sone/Bark) vs. subjective pitch (x in Bark)



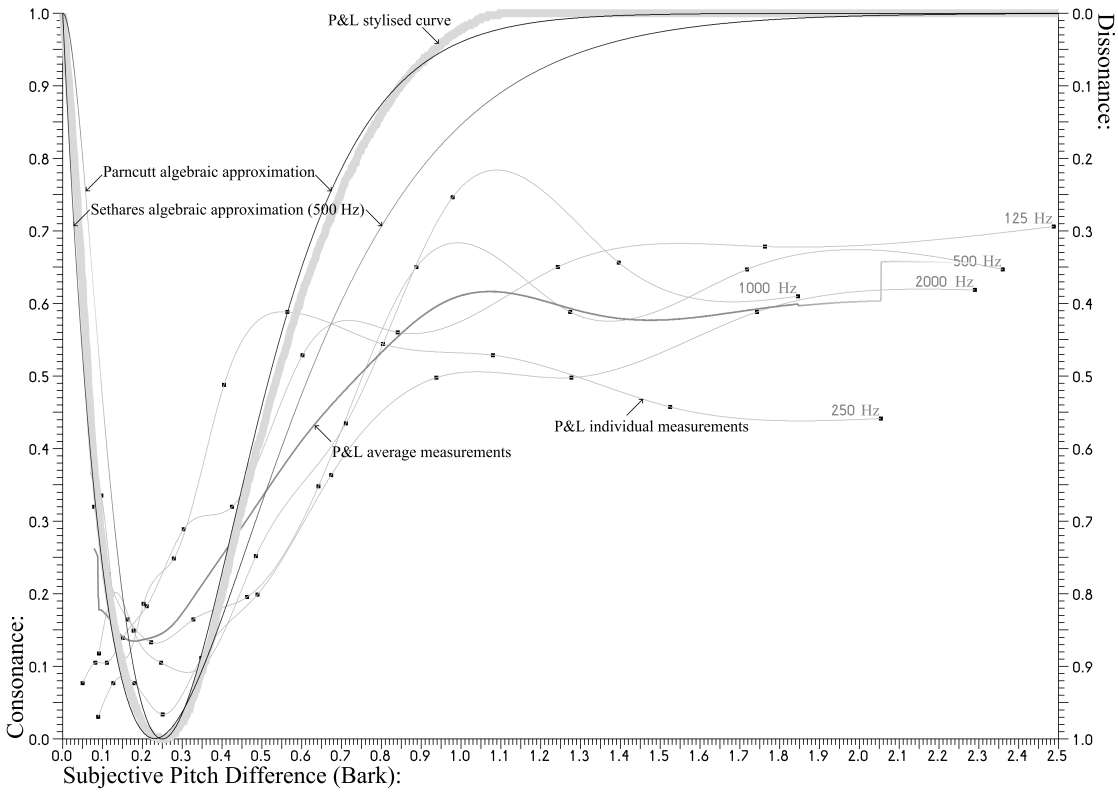
Γ45 – On physiological Phonetics

a) Depiction of the Ear

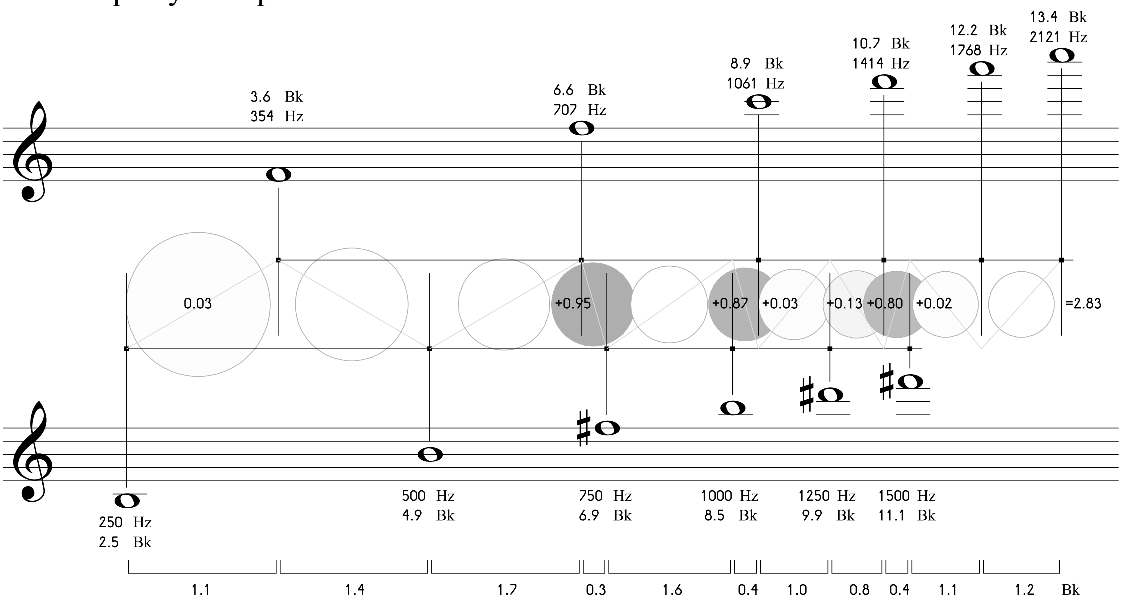
1. Pinna, ossicles, cochlea (also ‘unrolled’) and basilar membrane ‘from above’
2. Cochlea tilted by 90˚ around its longitudinal axis, membrane ‘from the side’
3. Cochea horizontally turned 90˚ showing cross-sections viewed ‘from the front’
4. Enlargement of frontmost cochlear cross-section shown in item 3
5. Further enlargement of part of the cross-section in item 4 with Organ of Corti



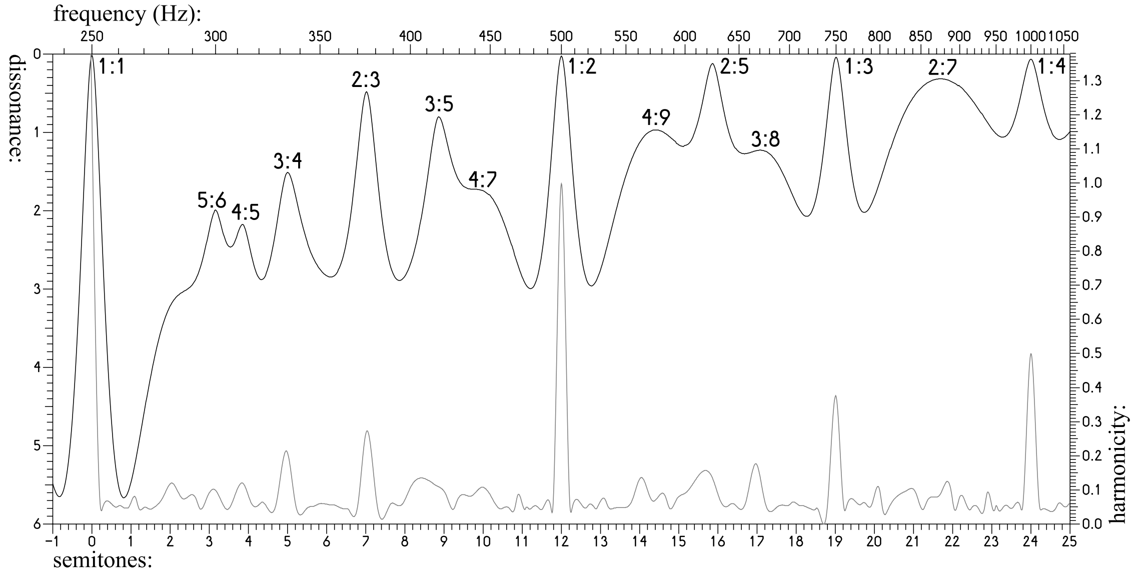
b) Plomp and Levelt’s experimental results on consonance and dissonance shown against subjective pitch in Bark (their measurements – here spline-connected – in light grey, Plomp and Levelts’s stylised dissonance curve in bold light grey, with algebraic approximations by Parncutt and Sethares nearby in variable grey)



c) Calculation after Plomp and Levelt of the dissonance of a tritone with six equally loud partials

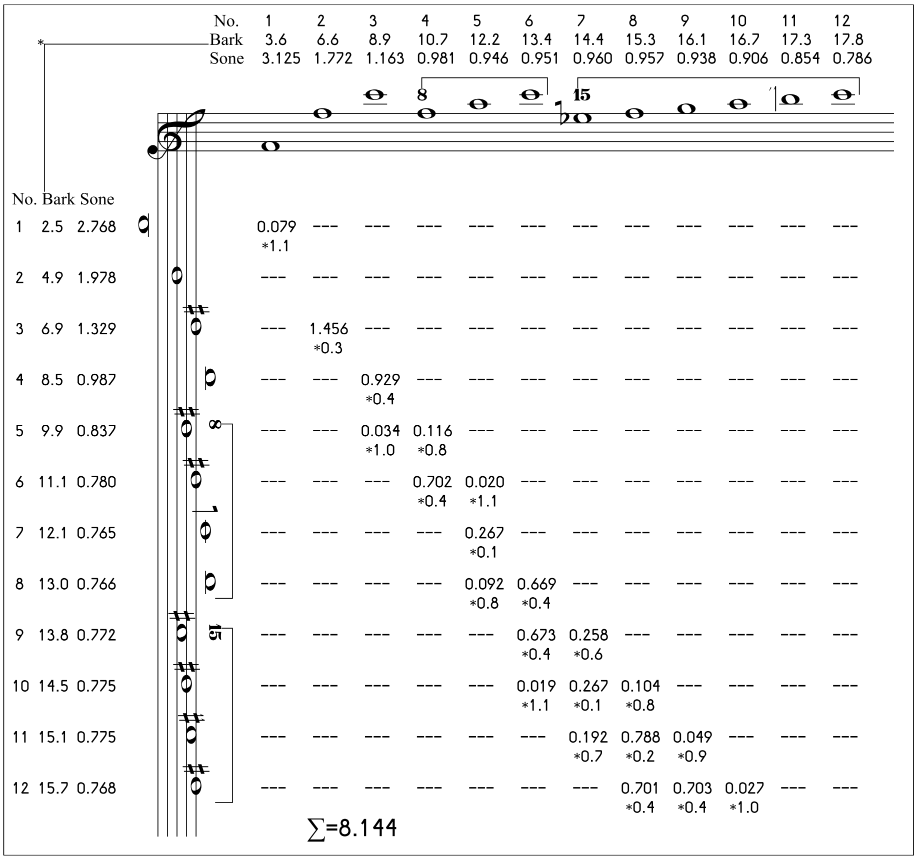


d) Calculation after Plomp and Levelt of the dissonance of various pairs of complex tones with six equally loud partials each, in a range of two octaves and compared with a harmonicity curve first described in Chapter 19 (grey)

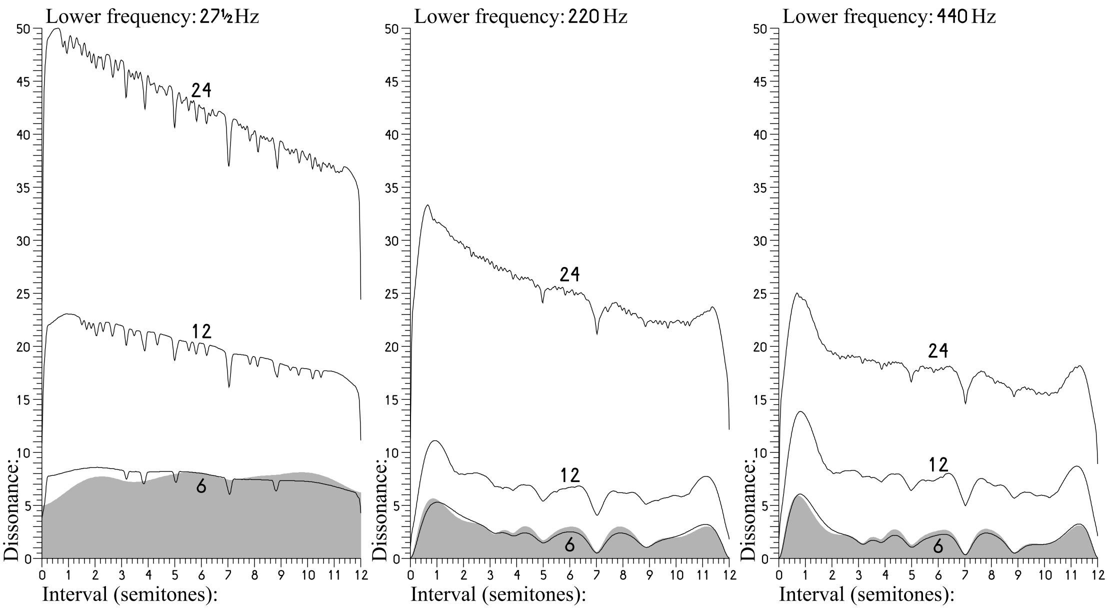


...Γ45...

e) Calculation according to Plomp and Levelt of the total dissonance of a sawtooth-tritone at 55 dB(SPL) with the additional involvement of subjective loudness – Bark intervals are marked with a ‘*’; the corresponding P&L dissonances (calculated after Parncutt) yield – multiplied by √(Sone$_1$ x Sone$_2$) – the final values, which add up to Σ.



f) Three dissonance calculations of pairs of sawtooth-tones with partials totalling 6, 12 und 24 also involving their loudness, ranging over one octave each, beginning with the fixed lower frequency shown (the calculation after Plomp and Levelt/ Parncutt with six equally loud partials is shown as grey areas)



T46 – Orthographic Representation of Vowels and Consonants according to the International Phonetic Association (IPA)

a) Selected vowels, indicating their physiological production

	Tongue position			
Openness of jaw	front	central	rear	Tongue height
close	i y	ɨ ʉ	ɯ u	high
near-close	ɪ ʏ		ʊ	semi-high
close-mid	e ø	ɘ ɵ	ɤ o	higher middle
mid		ə		middle
open-mid	ɛ œ	ɜ ɞ	ʌ ɔ	lower middle
near-open	æ	ɐ		semi-low
open	a ɶ		ɑ ɒ	low

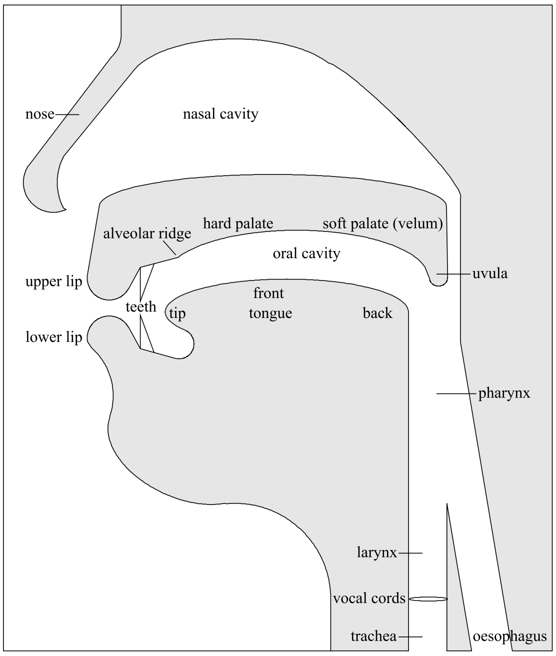
b) Selected consonants, indicating their physiological production

	Place of articulation									
Manner of articulation	Bilabial	Labiodental	Dental	Alveolar	Postalveolar	Retroflex	Palatal	Velar	Uvular	Glottal
Plosive	p b			t d		ʈ ɖ	c ɟ	k g	q ɢ	ʔ
Nasal	m	ɱ		n		ɳ	ɲ	ŋ	ɴ	
Vibrant	ʙ		r						ʀ	
Flap			ɾ			ɽ				
Fricative	ɸ β	f v	θ ð	s z	ʃ ʒ	ʂ ʐ	ç ʝ	x ɣ	χ ʁ	h ɦ
Lateral-Fricative				ɬ ɮ						
Approximant	w*	ʋ		ɹ		ɻ	j	ɰ		
Lateral-Approximant				l		ɭ	ʎ	ʟ		

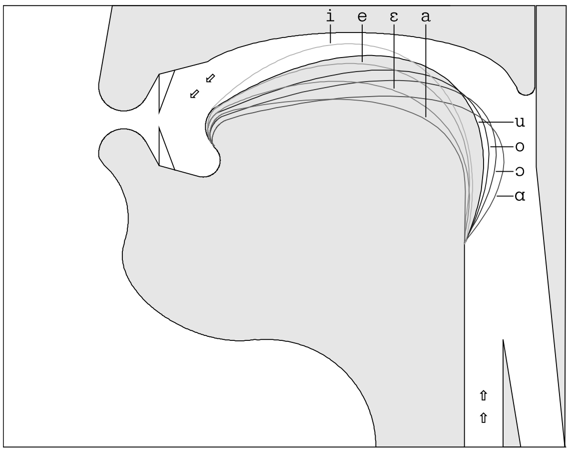
*usually referred to as a *labiovelar* approximant

Γ47 – The Vocal Tract

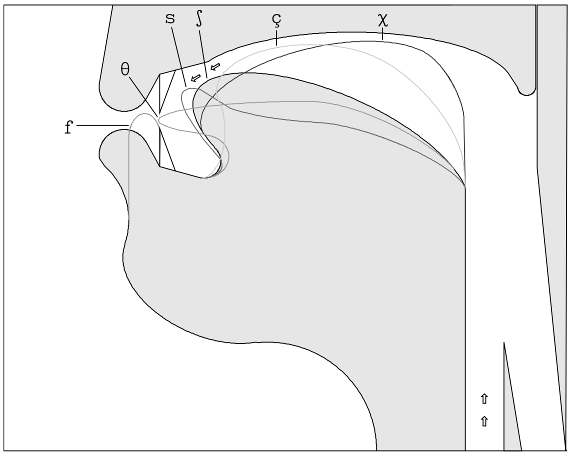
a) anatomically schematised



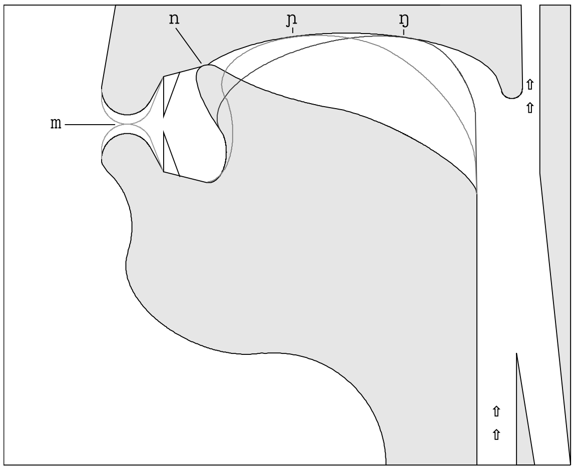
b) the tongue’s role in the production ... of frontal [ieɛæ] and rear [uoɔɒ] vowels



c) ...of fricatives [fθsʃçχ]

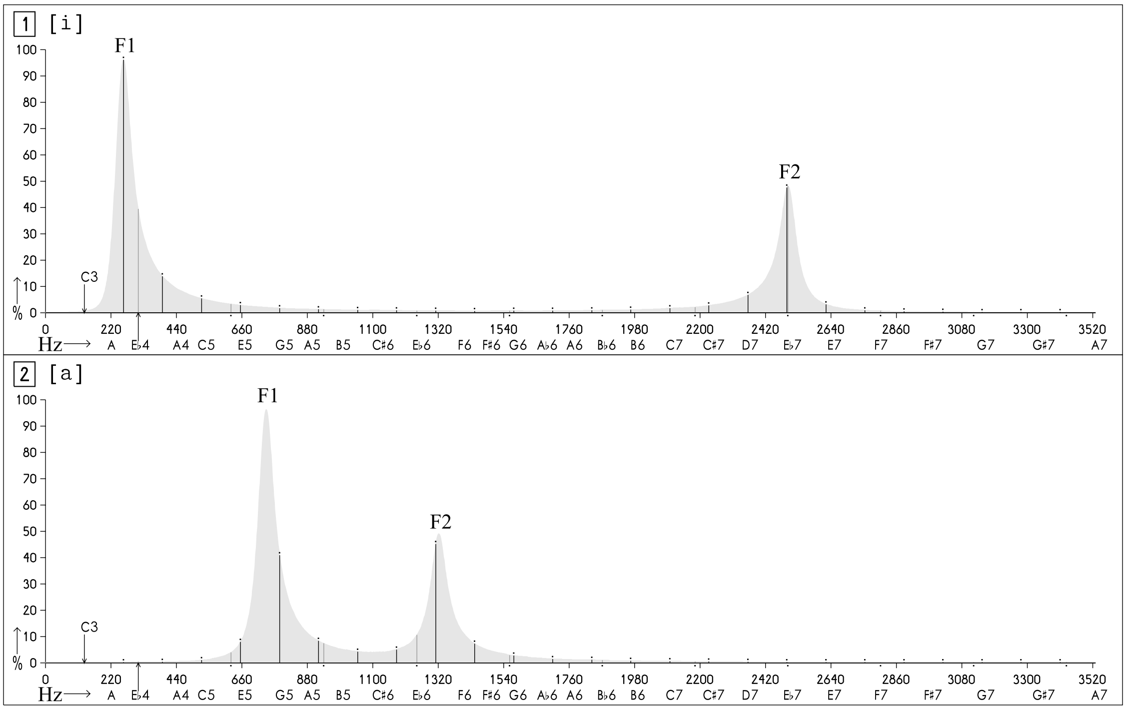


d) ...of nasals [mnɲŋ]

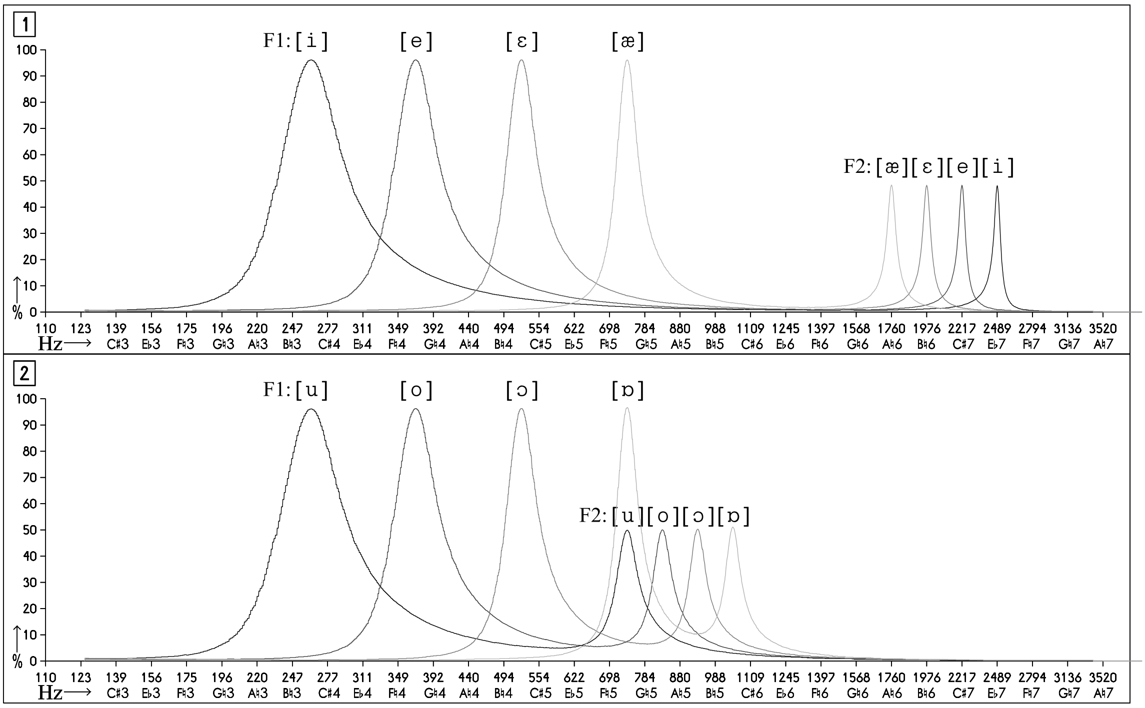


Γ48 – Formants of speech

a) Spectra of [i] and [a] on fundamentals C3 and E♭4, with formants F1 and F2

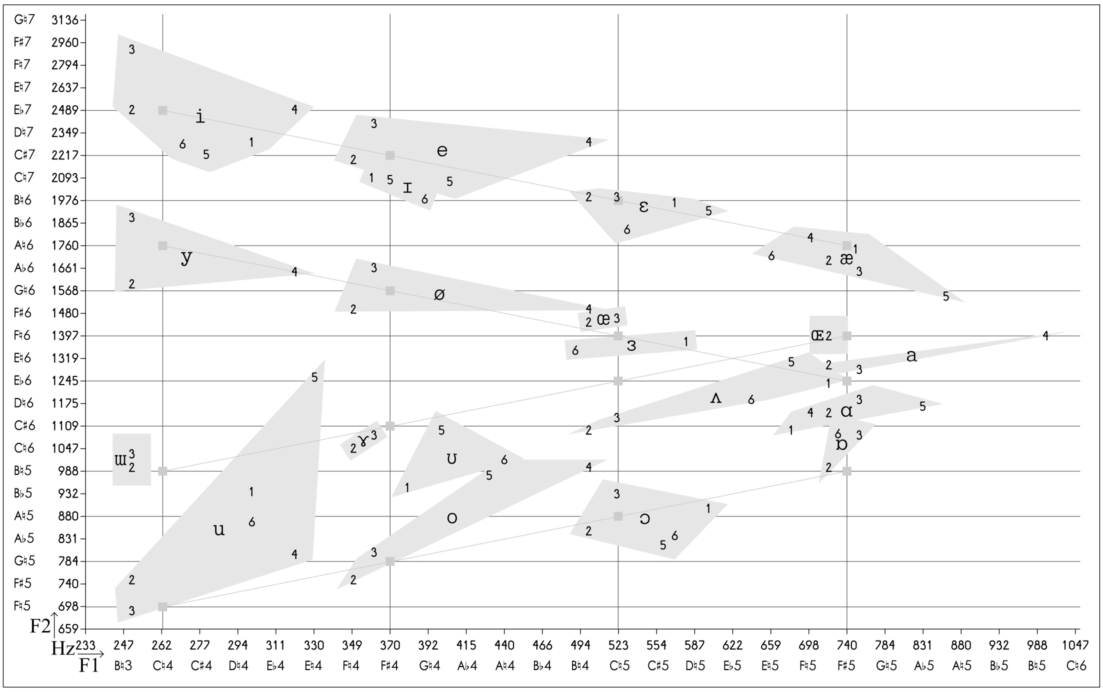


b) Formant comparison: 1) the frontal vowels [ieɛæ], 2) the rear vowels [uoɔɒ]

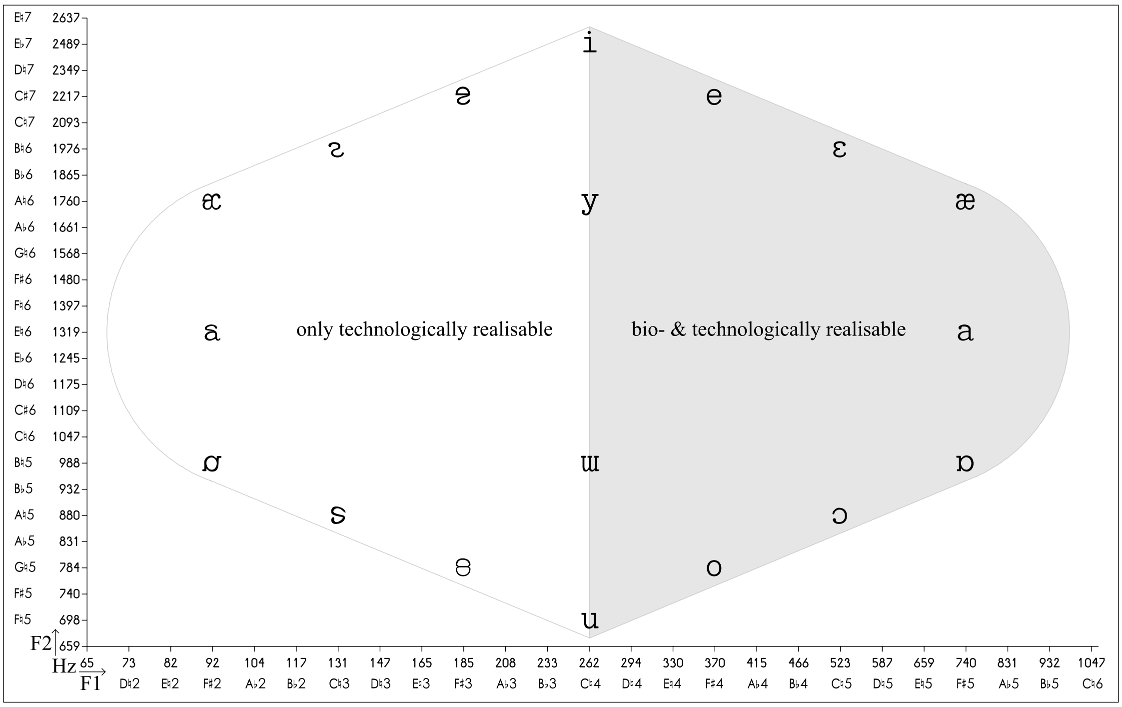


c) Two-dimensional formant chart of vowels indicating six sources (s. below) as well as the four trajects [ieɛæ], [yøœɑ], [ɯɤʌɶ] and [uoɔɒ]
Sources:

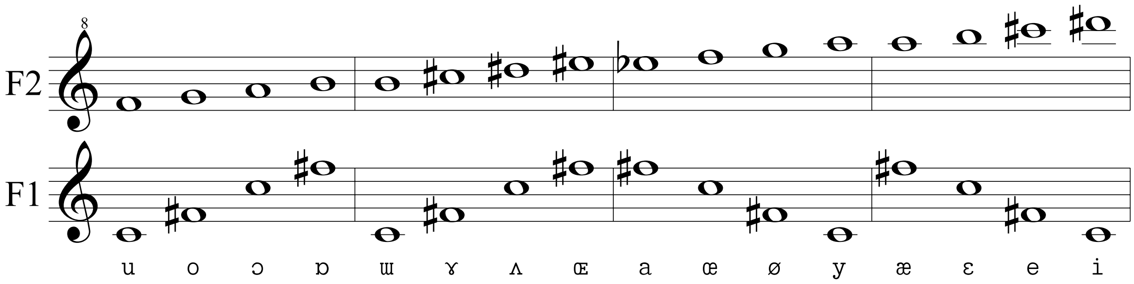
1) after J. C. Wells, quoted in D. B. Fry: The Physics of Speech (1979), p.79
2) from the Institute of Phonetics, Cologne University: Reports/No. 1 (1973), p.11
3) from Delattre et al: *Voyelles synthetiques à deux formants et voyelles cardinales* (1951), quoted inW. Hess, Bonn University: *Grundlagen der Phonetik/Deskriptive Phonetik* (2002)
4) from Wikipedia: <*http://en.wikipedia.org/wiki/Formant*>
5) from Kevin Russell, University of Manitoba: *General Phonetics* (2003)
6) from Gordon Peterson & Harold Barney: *Control methods used in a study of the vowels* (1952), JASA No.24, pp.175-184, quoted in various internet sources



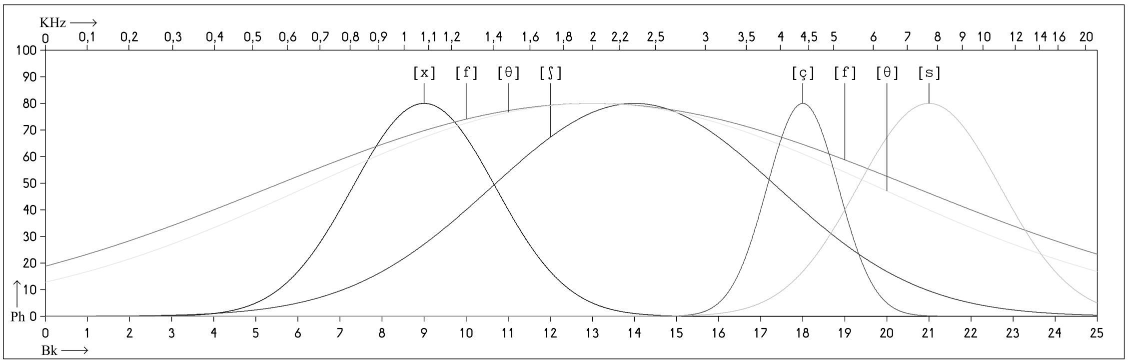
d) Two-dimensional formant chart of the vowels [ieɛæaɒɔouɯy] with their (abiological) F1 mirror images.



e) score notated formants F1 and F2 of the 16 vowels [uoɔɒ ɯɤʌɶ aœøy æɛei]



f) spectral envelopes of the fricatives [xfθʃçs] shown as Phon (y) vs. Bark (x)



g) sonagramme of a seamless transition through the fricatives [xfθʃçs]