

# AMATH 582 Homework Two: Shredding Analysis

Daniel W. Crews

February 10, 2021

## Abstract

Spectral analysis consists of far more than determination of the Fourier coefficients of a time series. This numerical experiment explores the spatial distribution of a signal's frequency spectrum (or spectrogram) through the practical examples of classic shredding moments in music history (here, Guns and Roses and Pink Floyd). To compute the spectrograms a short-time (masked) Fourier transform is used, and in particular this report explores its filtering using the Wigner distribution function of the time series. Here the Wigner function is computed as an  $\mathcal{O}(N^2)$  operation in terms of the Fourier coefficients.

## 1 Introduction and Overview

The purpose of this work is to visualize some of the most famous shredding in rock and roll history (Pink Floyd and Guns and Roses (GNR)) by construction of a spectrogram. Section 2 outlines the theory of the combined Gabor-Wigner distribution used for the spectrogram, followed by a section detailing implementation of the algorithm. Results are presented for the opening guitar riff in GNR, bass line in Floyd, and a stab at the Floyd guitar solo. The attached appendices describing elements of Wigner distributions are hoped not to count towards the overall page limit and included merely for completeness. The report itself is kept concise.

## 2 Theoretical Background

### 2.1 Time-frequency analysis

The Fourier decomposition of a periodic signal  $f(t+L) = f(t)$  exchanges the coordinate  $t$  for the distribution  $\hat{f}(\omega)$  in frequency  $\omega$ , which when continuous yet periodic in  $t$  is discrete in  $\omega$ ,

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{i\omega_n t}, \quad \hat{f}(\omega) = \sum_{n=-\infty}^{\infty} c_n \delta(\omega - \omega_n), \quad \omega_n = \frac{2\pi}{L} n. \quad (1)$$

Each spectral coefficient is found through integration with the unmasked Fourier integral kernel,

$$c_n = \frac{1}{L} \int_0^L f(t) e^{-i\omega_n t} dt. \quad (2)$$

The total frequency distribution  $\hat{f}(\omega)$  counts the orthogonal components of  $f(t)$  but is completely delocalized (in coordinate  $t$ ). On the other hand the original distribution  $f(t)$  does not convey local frequencies. However, the spatial distribution of signal frequencies is of broad interest in science, demonstrated by the following abstract example. Consider energy distributed across a range of frequencies with mean zero,

$$\psi(t) = \sum_{n=1}^N \cos(2\pi n t + \varphi(n)) \quad (3)$$

with  $\varphi$  a phase variable. If  $\varphi(n)$  is randomly distributed in  $[0, 2\pi)$  then  $\psi$  is white noise, yet if  $\varphi \equiv 0$  then the signal energy is increasingly singularly concentrated in coordinate  $t$  (i.e.  $\lim_{N \rightarrow \infty} \psi(t) \rightarrow \delta(t)$ ). Both functions have the same amplitude distribution  $|\hat{f}(\omega)|$  (where phase information is discarded).

The goal of time-frequency analysis is to establish a function which can capture the spatial distribution of signal energy and describe its frequency contents as well. The two classic approaches are the short-time method (Gabor transform) and the instantaneous autocorrelation transform (or Wigner function).

## 2.2 Masked Fourier transform (Gabor method)

The short-time Fourier transform, also called the Gabor transform, masks the Fourier kernel with a window,

$$\mathcal{G}_n(t) = \frac{1}{L} \int_0^L f(\tau) g(t, \tau) e^{-i\omega_n \tau} d\tau \quad (4)$$

where  $g(t, \tau)$  is an integrable window function which should be translationally covariant and even about  $t = \tau$  [1]. Common choices of window are the Gaussian and the bump function. The bump function is used for this study because it has compact support and can slide across a periodic domain without overlapping itself,

$$g(t - \tau) = \exp(1) \exp\left(\frac{-a^2}{a^2 - (t - \tau)^2}\right), \quad t - \tau \in (-a, a), \quad g \equiv 0, \quad t - \tau \notin (-a, a). \quad (5)$$

The discrete Gabor transform is approximated by simple quadrature just like the discrete Fourier transform, so it is implemented by multiplying data by the bump function on its points and applying an FFT.

## 2.3 Instantaneous autocorrelation Fourier transform (Wigner method)

A distribution's autocorrelation is essentially a moment with itself given some delay  $t$ ,

$$\Phi(t) = \int_{-\infty}^{\infty} f^*(t - \tau) f(\tau) d\tau. \quad (6)$$

The Wigner distribution is the Fourier transformation of the *instantaneous* autocorrelation,

$$W(t, \omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} f^*\left(t - \frac{\tau}{2}\right) f\left(t + \frac{\tau}{2}\right) e^{-i\omega \tau} d\tau, \quad (7)$$

so arranged for Hermitian symmetry. The Wigner function is the distribution of the intensity of a signal in its *phase space*. Although  $W(t, \omega)$  may have negative values, it is the energy distribution because it is the unique distribution whose zeroth moments recover the signal intensity in both space and frequency,

$$\int_{-\infty}^{\infty} W(t, \omega) d\omega = |f(t)|^2, \quad \int_{-\infty}^{\infty} W(t, \omega) dt = |\hat{f}(\omega)|^2. \quad (8)$$

For a periodic function with Fourier series  $f(t) = \sum_n c_n e^{i\omega_n t}$ , the Wigner function is analytically

$$W\left(t, \frac{\pi}{L}\ell\right) = \sum_{n=-\infty}^{\infty} (c_n e^{i\omega_n t})^* (c_{\ell-n} e^{i\omega_{\ell-n} t}), \quad \ell \in \mathbb{Z} \quad (9)$$

or a discrete convolution of the Fourier series with itself (see Appendix A for derivation and discussion of the Wigner function of an analytic signal (used in this report) and Appendix B for examples). While  $W(t, \omega)$  picks up signals with great accuracy, its interference patterns predict seemingly erroneous signal intensities.

## 2.4 Filtered Wigner function (or combined Wigner-Gabor transform)

In the context of signal processing these interferences (some of them artifacts of the signal's assumed periodicity), make the unfiltered Wigner function difficult to use. The Gabor transform helps to pick out where the signal is actually non-zero (as in Fig. 10). An interesting compromise, equivalent to filtering the Wigner distribution, is to simply multiply the two distributions together,

$$\psi_{GW}(t, \omega) = W(t, \omega) \mathcal{G}(t, \omega). \quad (10)$$

This combined approach is pursued in the present study.

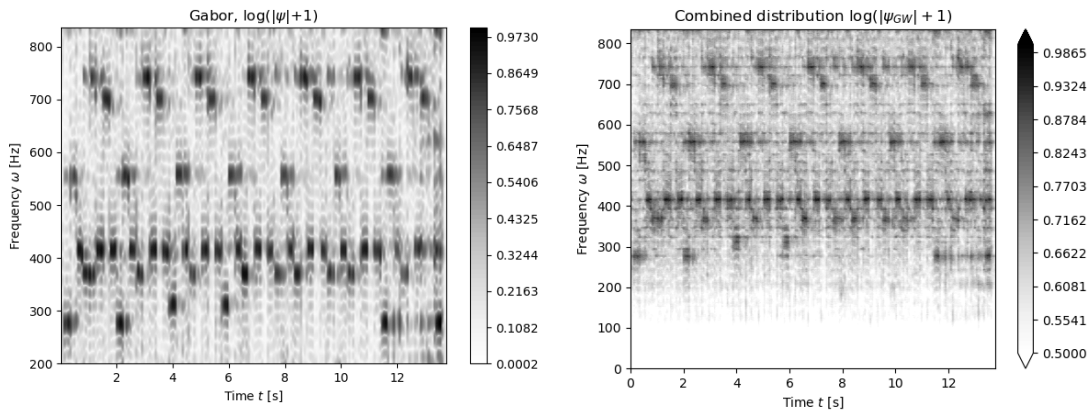


Figure 1: Spectrogram of the first fourteen seconds of Guns and Roses’ “Sweet Child of Mine,” computed by Gabor (left) and Gabor-Wigner (right) methods with 0.02 s long Gabor window. The Gabor-Wigner approach resolves spectral lines better (predicting more localization in frequency) and hints at additional structure (chords) in the music through spectral lines at harmonic modulation (mean) frequencies. The original low D note (chord root) is carried up to G by the end of the intro section and repeats at 12 seconds.

### 3 Algorithm Implementation and Development

Development of the analysis algorithm proceeded as follows. The provided musical waveforms are considered to be periodic and analyzed by discrete Fourier transform. After picking out an appropriate frequency range (*e.g.* the bass line in Floyd), the waveform is both Wigner and Gabor transformed, where only the analytic signal (or real FFT) is used in order to reduce low-frequency interference in the Wigner function (*c.f.* Section A.1). The two phase space distributions are then multiplied together and the (logarithm of) the resulting spectrogram plotted as filled contours. Implementation details are in Appendices C and D.

Having obtained the spectrogram, the musical score is then reconstructed by first listening to the recording to figure out the time signature (both are 4-4). Then the frequencies are compared to a music scale chart to figure out which notes they correspond to. Finally, the score is written down. Note that the Gabor transform is sufficient for the purposes of this assignment, however the combined Wigner-Gabor spectrogram is of interest by revealing harmonic combinations (additional spectral lines) in the music. Github repository for this work is available [here](#).

## 4 Computational Results

The results are divided into two parts, first analyzing the Guns and Roses clip and then the Pink Floyd one.

### 4.1 Sweet Child of Mine guitar riff

The Guns and Roses sample was analyzed as described in Section 3 by computing its combined Wigner-Gabor spectrogram. The result for both the original Gabor and the combined Wigner-Gabor transform are shown in Fig. 1 for contrast. The time signature is counted to be at 4-4, and the guitar pattern for the first measure is identified as (all eighth notes) D (294 Hz), D (587 Hz), A (440 Hz), G (392 Hz), G (784 Hz), A (440 Hz), F# (740 Hz), and A (440 Hz) by consulting an online guitar fretboard chart. This pattern is clearly visible in both the Gabor and Gabor-Wigner spectrograms, with more pure tones predicted by the Wigner method. The identified music score is shown in Fig. 2 (without key signature, apparently D).

### 4.2 Comfortably Numb guitar solo bass line

In a similar manner the bass line under the extremely sick Pink Floyd guitar solo was identified in Fig. 3 through Gabor and Gabor-Wigner spectrograms, consisting of the notes B (124 Hz), A (110 Hz), G (98 Hz),

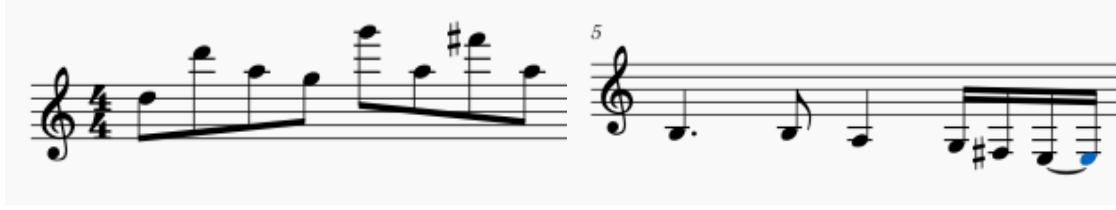


Figure 2: Identified scores for Sweet Child of Mine guitar riff (left) and Pink Floyd bass line (also 4-4, right) made in the free software “MuseScore 3”. The bass line appears to repeat B before descending.

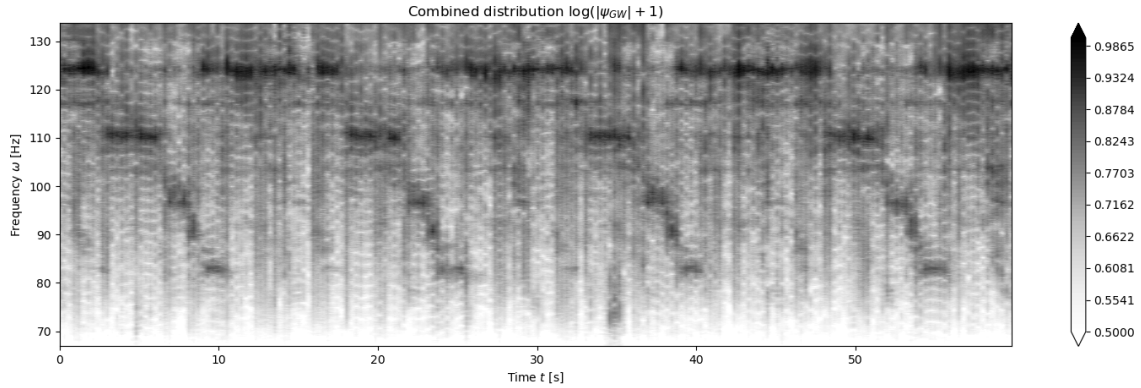


Figure 3: Spectrogram of the bass line during the guitar solo in Pink Floyd’s “Comfortably Numb,” computed by Gabor-Wigner transform with 0.023 s long Gabor window. The bass line maintains a slow tempo throughout the mind-melting guitar solo and appears to consist of (on its long timescale) a dotted quarter and eighth (B), quarter note (A), and descending three eighth notes (G, F#, E).

F# (90 Hz), and E (80 Hz) as detailed Fig. 2.

### 4.3 Attempt at guitar solo in Comfortably Numb

An attempt was made at part of the guitar solo in Pink Floyd. The Wigner filter was not used as it was too computationally expensive to extend so high into frequency for such a long clip on an  $\mathcal{O}(N^2)$  operation, so the Gabor transform with FFT was used. The first five seconds of the Floyd guitar solo are examined in Fig. 4. A lot of distortion is present which made the spectral signatures rather diffuse. A general structure is seen as a descending sequence of notes beginning on the first string, apparently in key of D.

## 5 Summary and Conclusions

This numerical experiment combined the short-time Fourier transform (Gabor) with the Wigner distribution to enhance each other’s spectral signatures. The combination of the two can be thought of as a filtered version of either, compromising between the strengths of each method. By this method the musical score was identified of two classic music clips. Python was used for implementations in this exercise.

## References

- [1] J.N. Kutz. *Data-driven Modeling and Scientific Computation*. Oxford University Press, 1st edition, 2013.
- [2] L.D. Paarmann. Analysis of the wigner-ville transform of periodic signals. *IEEE*, 1994.

## Appendix A Wigner distributions of periodic functions

A remarkable property of the Wigner distribution is that it is straightforward (though computationally expensive) to calculate once a Fourier analysis of the function  $f(t)$  has been determined. To see this, consider a periodic function expressed by its Fourier series,

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{i\omega_n t}, \quad \omega_n = \frac{2\pi}{L} n \quad (11)$$

where  $f(t+L) = f(t)$ . Note that the function  $f(t)$  is continuous in  $t$  but discrete in its spectrum. The Wigner function inherits this property. Substituting into Eqn. 7, one has

$$W(t, \omega) = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} (c_n e^{i\omega_n t})^* (c_m e^{i\omega_m t}) \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-i(\omega - (\omega_n + \omega_m)/2)\tau} d\tau \quad (12)$$

which then simplifies from the orthogonality of the Fourier basis into the expression

$$W(t, \omega) = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} (c_n e^{i\omega_n t})^* (c_m e^{i\omega_m t}) \delta\left(\omega - \frac{1}{2}(\omega_n + \omega_m)\right). \quad (13)$$

The summation is on the infinite lattice  $(n, m) \in \mathbb{Z}^2$ , yet the delta function restricts the frequency spectrum to be discrete. The comb picks out all frequencies  $\omega$  integer multiples of *half* the fundamental frequency,

$$\omega = \frac{\pi}{L} \ell, \quad \ell \in \mathbb{Z}. \quad (14)$$

From the frequency-matching condition and summation on diagonals of the lattice one has

$$W\left(t, \frac{\pi}{L} \ell\right) = \sum_{n=-\infty}^{\infty} (c_n e^{i\omega_n t})^* (c_{\ell-n} e^{i\omega_{\ell-n} t}), \quad \ell \in \mathbb{Z} \quad (15)$$

as shown in Fig. 5. Note that if the function  $f(t)$  is real, then the Wigner distribution  $W(t, \omega)$  is also real. This calculation reveals the Wigner function as particularly simple in the Fourier basis, being just a discrete

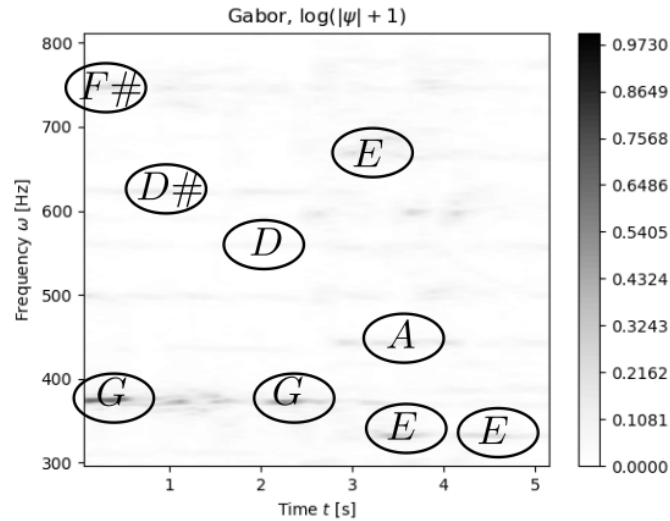


Figure 4: Spectrogram of the first five seconds of the guitar solo in Pink Floyd’s “Comfortably Numb,” computed by Gabor transform with 0.1 s long Gabor window. Identified notes are circled and labelled.

convolution of the Fourier series with itself for each  $\ell$ . For example, the zero-mode is the summation

$$W(t, 0) = \sum_{n=-\infty}^{\infty} |c_n|^2 e^{-i2\omega_n t} \quad (16)$$

which has zero time-average (contributing nothing to the spectrum  $\hat{f}(\omega)$ ) but necessary to describe the *local* (in coordinate  $t$ ) deviation from mean-zero in a function's spatial energy distribution. As an energy it oscillates with twice the frequency of its corresponding mode. In order to avoid aliasing interference the distribution should be truncated at the Nyquist frequency of the series,

$$W\left(t, \frac{\pi}{L}\ell\right) = \sum_{n=-k}^k (c_n e^{i\omega_n t})^* (c_{\ell-n} e^{i\omega_{\ell-n} t}), \quad \ell \in \mathbb{Z}. \quad (17)$$

### A.1 Wigner distribution of an analytic signal

The *analytic signal*  $f_s(t)$  joins the Hilbert transform of a function  $f(t)$  to itself,

$$f_s(t) = f(t) + i\mathcal{H}[f](t). \quad (18)$$

If a signal is real-valued then the spectrum of its analytic signal is the one-sided series

$$f_s(t) = \sum_{n=0}^{\infty} 2c_n e^{i\omega_n t}, \quad \omega_n = \frac{2\pi}{L}. \quad (19)$$

In signal processing this is called the “real Fourier series”. If desired, one can eliminate the interference effects at the low-frequencies present in the Wigner distribution of Eqn. 17. It should be noted that this process will discard the beat effects, and the marginal distribution will be altered to instead return the intensity spectrum of the analytic signal  $|f_s(t)|^2$ . This can be useful in data analysis however, and the formula for the Wigner distribution simplifies.

Repeating the calculation of Section A, the distribution of  $f_s(t)$  is given by [2]

$$W\left(t, \frac{\pi}{L}\ell\right) = 2 \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} (c_n e^{i\omega_n t})^* (c_m e^{i\omega_m t}) \delta\left(\frac{\pi}{L}(\ell - (n + m))\right), \quad \ell \in \mathbb{Z}^+. \quad (20)$$

The summation is now in the first quadrant only of the lattice in Fig. 5, and the summation along the diagonal  $m = \ell - n$  truncates. For a given mode  $\ell$ , interference is possible only between frequencies up to and including  $\ell$ , giving

$$W\left(t, \frac{\pi}{L}\ell\right) = 2 \sum_{n=0}^{\ell} (c_n e^{i\omega_n t})^* (c_{\ell-n} e^{i\omega_{\ell-n} t}), \quad \ell \in \mathbb{Z}^+. \quad (21)$$

This reduces interferences but discards important information regarding *frequency differences*, instead describing only frequency-sum modulations. For example, beat frequencies  $f_b = |f_2 - f_1|$  are described by the Wigner distribution of an analytic signal more like the corresponding Gabor transform.

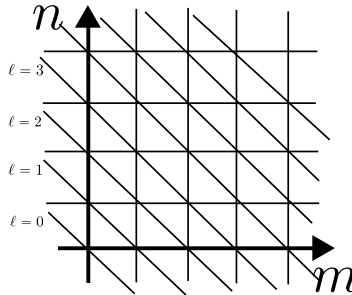


Figure 5: Summation on the lattice  $(n, m)$  resolves into a single summation on a diagonal satisfying  $m = \ell - n$  for each frequency component  $\ell$  of the Wigner distribution.

## Appendix B Examples of Wigner distributions

The following are simple examples of Wigner functions presented in order to build intuition. An objective of the following discussion is to point out that the interference patterns present in the distribution are physically meaningful, but are perhaps undesirable from a signal processing perspective, in particular if the signal is not truly periodic (e.g. a music recording).

### B.1 Single sine function

The distribution of a basic sine function  $f(t) = \sin(t) = -\frac{1}{2}ie^{it} + \frac{1}{2}ie^{-it}$  is given by

$$W\left(t, \frac{\pi}{L}\ell\right) = \frac{1}{4} \begin{cases} 1 & \ell = 2 \\ -2 \cos(2t) & \ell = 0 \\ 1 & \ell = -2 \end{cases} \quad (22)$$

so that summing in  $\ell$ , one finds the energy distribution  $|f(t)|^2 = \frac{1}{2} - \frac{1}{2} \cos(2t) = \sin^2(t)$ . The Wigner distribution is shown below in Fig. 6. Evidently the interference pattern at  $\ell = 0$  is due to combination of the positive and negative frequency components. In the signal processing context it is remarked that the zero-mode is “unexpected”, but this example demonstrates that it is necessary in order to describe the non-zero mean in energy of the sine wave. The zero-mode  $\frac{1}{2} \cos(2t) = 0$  when  $f(t) = |f(t)|^2$  and no more energy is needed than that present in  $\ell = \pm 2$ .

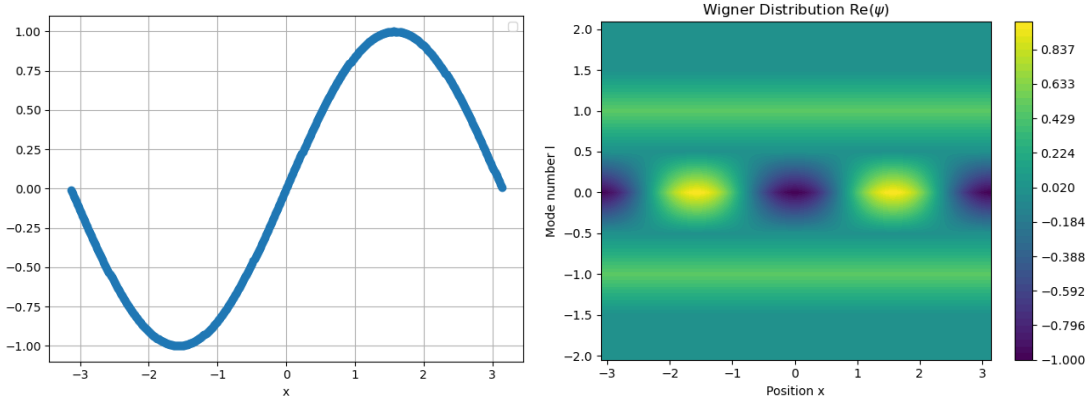


Figure 6: Wigner function of the sine wave (right), note interference between modes  $n = \pm 1$ .

### B.2 Classic beat wave

Now consider a function with more spatially modulated structure than the sine wave,

$$f(t) = \cos(4t) + \cos(5t) = 2 \cos\left(\frac{1}{2}t\right) \cos\left(\frac{9}{2}t\right). \quad (23)$$

This waveform has a classic beat structure as shown in Fig. 7, with a low-frequency of  $\omega_B = 1/2$  and a modulation frequency of  $\omega_M = 9/2$ . The spectrum of the signal (its orthogonal components) is of course only present at  $\omega = 4$  and  $5$ , but the spatial structure of the waveform has a *low-frequency variation* which is present in the mode  $\omega = 0.5$  and a *harmonic modulation frequency* at  $\omega = 4.5$ . A generic spectrogram computed by a windowed Fourier transform will typically not see the energy present in the harmonic combinations resulting in spatial structure of the waveform (appearing instead as a bump).

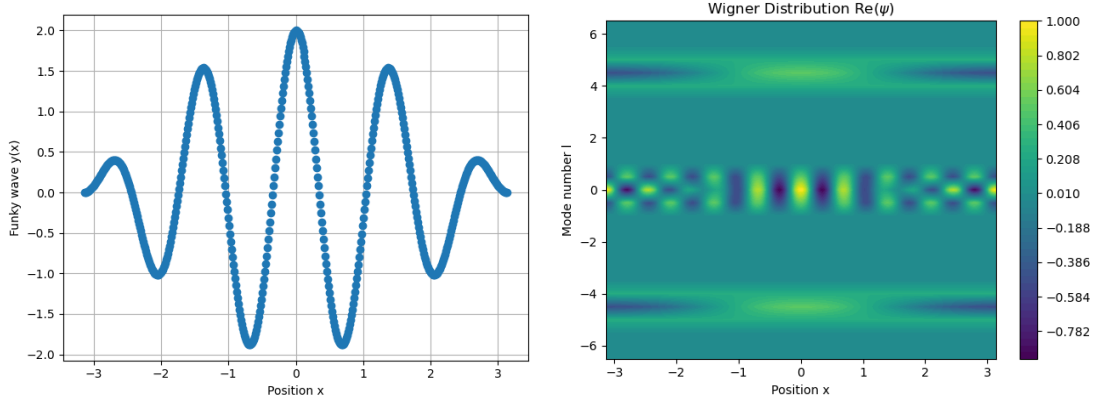


Figure 7: Spatial modulation in the superposition of  $\cos(4t)$  and  $\cos(5t)$  (left) and Wigner distribution (right). Low-frequency energy is present in modes  $\ell = 0, \pm 1$  and high-frequency modulation in  $\omega = 9/2$  which cancel out when considering the total frequency spectrum.

### B.3 Elliptic cosine

Another good example of spatial structure is the elliptic cosine function with Fourier series

$$\text{cn}(4K(m)x|m) = \frac{\pi}{K(m)\sqrt{m}} \sum_{n=0}^{\infty} \frac{\cos(2\pi(1+2n)x)}{\cosh(\frac{\pi}{2}(1+2n)\frac{K'(m)}{K(m)})}. \quad (24)$$

where  $K(m)$  is a complete elliptic integral. For parameter  $m$  close to one the wave energy becomes very localized, shown in Fig. 8. Where the function is peaked there is a zero-mode (DC) component to the energy! However in its totality there is no mean value to the function. In order to cancel out this zero-mode a *virtual frequency distribution* is present in coordinate  $x$  where the signal energy is entirely zero. This phenomenon can be understood as a “spatial beat” in contrast to the usual frequency beat. In the context of the energetic distribution of a truly periodic signal this interference phenomenon has physical significance.

### B.4 D Major arpeggio on the guitar

Yet for analysis of a waveform *assumed* to be periodic the spatial beat effect is detrimental. An example of such is in analysis of a musical waveform. Below the Wigner function (of the analytic signal, c.f. Section A.1) is calculated of the four notes of a D Major chord played in succession. The four notes of the D chord

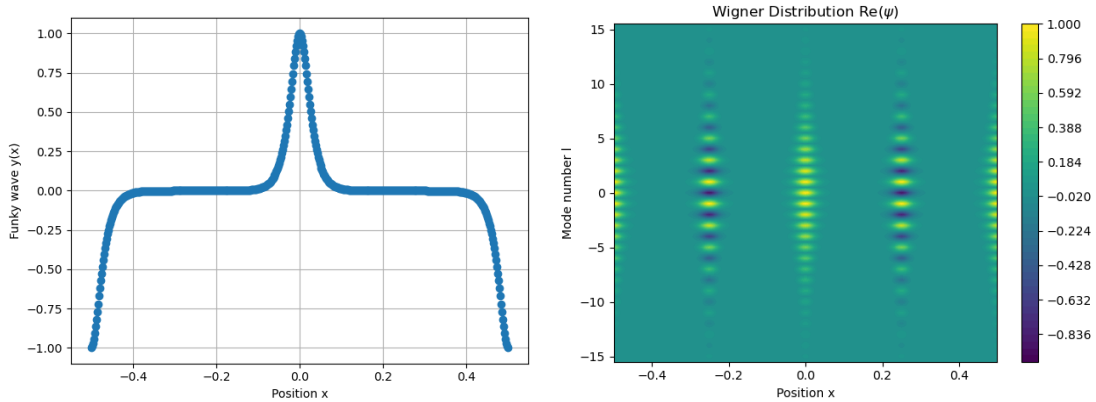


Figure 8: Elliptic cosine and its Wigner distribution for parameter  $m = 1 - 10^{-10}$ .



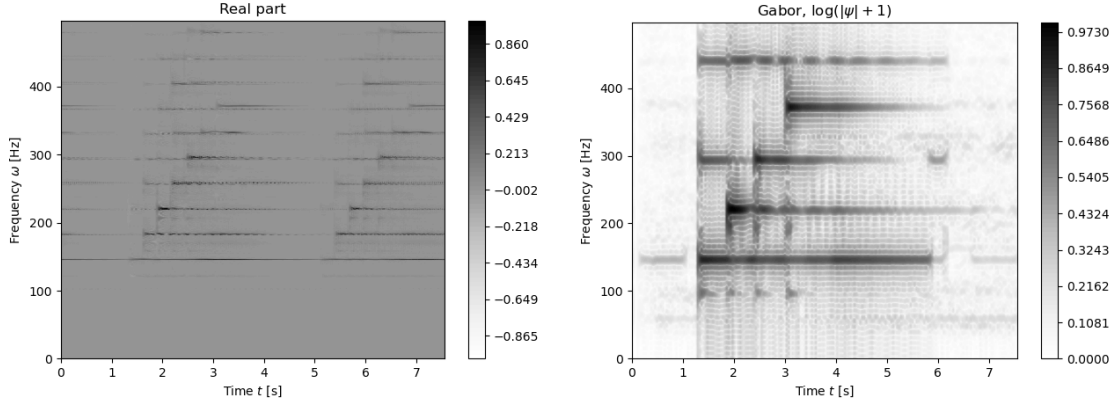


Figure 9: Wigner distribution (left) and typical spectrogram (right) of four notes played on the guitar, each calculated from the analytic signal of the waveform. The Wigner function has an interference pattern where the signal is silent (0-1.1 s and the end). It also describes interferences between each pair of notes, in particular at the chord center frequency of 254 Hz, (with a role in the chord’s psychoacoustical perception).

are 144 Hz (D), 216 Hz (A), 288 Hz (D), and 364 Hz (F#). A windowed Fourier transform is also computed and shown for comparison, which has only four spectral lines (plus their integer harmonics).

The Wigner function results in more localized signals, but also encounters *time beats* when the recording is silent (a purely artificial effect). It also predicts *interferences* between any pair of notes. One of the pleasant aspects of the D Major chord is its internal harmony, as shown in Table 1. The frequency of modulation between any pair of notes reinforce one another and fill out the succeeding intervals. The Wigner spectrogram picks up this interference pattern and indicates frequencies and relationships which may be *perceived* but not present in the orthogonal decomposition. A compromise distribution is shown in Fig. 10.

Chord Notes [Hz]	F#, 364		D, 288		A, 216		D, 144
First Neighbor Mean [Hz]		326		252		180	
Second Neighbor Mean [Hz]			290		216		
Center Frequency [Hz]				254			

Table 1: Frequencies of modulation in the D Major chord on the guitar.

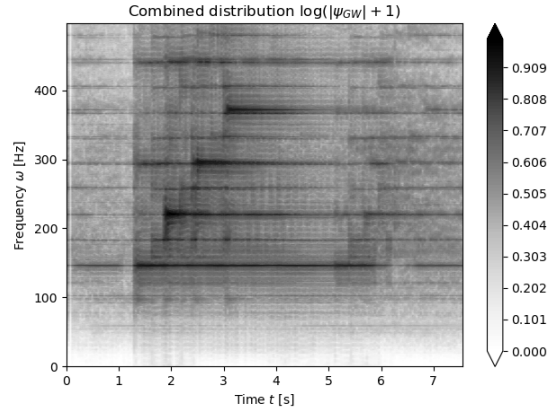


Figure 10: The combined spectrogram multiplies the windowed Fourier transform with the Wigner distribution. The resulting spectrogram separates the true signal from its “time beat” phantom part and includes spectral lines at the frequencies of modulation between the component notes of the arpeggiated D chord.

## Appendix C Python Functions

The following list compiles important Python functions used in implementation:

- `soundfile.read` is used to read in a `.wav` file as a list along with sample rate
- `numpy.fft.rfft` computes only the positive frequency components of the Fourier transform, used to construct the analytic signal of the time series.
- `numba.cuda.jit` is used to GPU-accelerate the brute-force calculation of the Wigner function.
- `numpy.roll` is used to roll the Gabor window of a specified number of points across the time series domain.

## Appendix D Python Implementation

Wigner function file `fourier.py`:

```
import numpy as np
import cmath
import numba.cuda as cuda

TPB = 8

##### Function to compute wigner function
#### Wigner function
def WT_as_f(x, k, c, L):
    # fund. frequency
    k1 = 2.0*np.pi/L

    # Set up kernel
    blockSize = (TPB, TPB)
    numBlocksX = (x.shape[0] + blockSize[0] - 1)//blockSize[0]
    numBlocksK = (k.shape[0] + blockSize[1] - 1)//blockSize[1]
    numBlocks = (numBlocksX, numBlocksK)

    # output on device
    dW = cuda.device_array((x.shape[0], k.shape[0]), np.complex128)
    # input
    dC = cuda.to_device(np.ascontiguousarray(c))
    dx = cuda.to_device(x)
    dk = cuda.to_device(k)

    # call kernel
    wignerKernel[numBlocks, blockSize](dC, dx, dk, dW, k1)

    return 2.0*dW.copy_to_host()

#### Wigner function kernel on GPU
@cuda.jit
def wignerKernel(dC, dx, dk, dW, k1):
    i, k = cuda.grid(2)

    # stay in bounds
    if (i >= dW.shape[0] or k >= dW.shape[1]):
        return
```

```

# zero elements
dW[i,k] = 0

# Compute transform
for n in range(k+1):
    f1 = dC[n]*cmath.exp(1.0j*k1*n*dx[i])
    f2 = dC[k-n]*cmath.exp(1.0j*k1*(k-n)*dx[i])
    dW[i,k] += f1.conjugate()*f2

```

Main implementation file `hw2.py`:

```

import numpy as np
import soundfile as sf

import matplotlib.pyplot as plt
import matplotlib as mpl

# My functions
import fourier as f

# Matplotlib Chunk size (for huge number of points)
mpl.rcParams['agg.path.chunksize'] = 10000

#### Read file (samplerate in Hz)
data, samplerate = sf.read('music_files/Floyd.wav')#sf.read('music_files/GNR.wav')
# Create time axis (suppose samples are given in midpoints of intervals of length dt)
dt = 1.0/samplerate
Nt = data.shape[0]
t = np.linspace(0, dt*(Nt+1), num=Nt+1)
# Midpoints
tbar = np.array([(t[i+1] + t[i])/2.0 for i in range(t.shape[0]-1)])
# Jacobian
J = dt/2.0
# Length
L = t[-1]-t[0]

#### Wavenumbers
k1 = 2.0*np.pi/L
lim0 = 0#15000
lim1 = 80000
wvs0 = k1*np.arange(lim0, lim1)

```

```

#### Compute fourier coefficients
# Use real for analytic signal trick
c = np.fft.rfft(data)[lim0:lim1]
wvs = np.fft.rfftfreq(data.shape[0], dt)[lim0:lim1]*2.0*np.pi#*(data.shape[0]*dt)
# Actual Fourier coefficients
c = c/(data.shape[0])

res_wig = lim1
wvs_wig = (np.pi/L)*np.arange(res_wig)

#### Check out spectrum
plt.figure()
plt.plot(wvs/(2.0*np.pi), np.absolute(c))
plt.show()

#### Compute Wigner transform
# Downsample time
t_final = 25.0#tbar[-1]
t_ds = np.linspace(0, t_final, num=200)
# Wigner dist.
wig = f.WT_as_f(t_ds, wvs_wig, c, L).transpose()

#### Compare to Gabor transform
gab_lim0 = 15000#15000#int(lim0//2)
gab_lim1 = 80000#int(lim1//2)
pts = 5000 # how many points in the window
window_size = pts*(tbar[1]-tbar[0]) # scaled window_size
#freqs = np.fft.fftfreq(xbar.shape[0], d=xbar[1]-xbar[0]) # frequencies in fft IN HERTZ
gabor_sg = np.zeros((t_ds.shape[0], gab_lim1-gab_lim0)) + 0j # gabor spectrogram
#fig, ax = plt.subplots()
size = int(tbar.shape[0]//2)
ratio = int((t_ds[1]-t_ds[0])/(tbar[1]-tbar[0]))
for i in range(t_ds.shape[0]):
    # Build window
    r = (np.arange(2*pts)-pts) + size
    window = np.zeros_like(tbar)
    window[r] = np.exp(1.0)*np.exp(-1.0/(1.0 - tbar[r]**2.0/window_size**2.0+1.5e-10))
    # Shift window
    window = np.roll(window, ratio*i - size)
    #window[rp] = np.exp(-1.0/(1.0 - shift_p[rp]**2.0/window_size**2.0)+1.0e-16)
    weighted_y = np.multiply(window, data)
    gab = np.fft.rfft(weighted_y)[gab_lim0:gab_lim1]/(data.shape[0])
    #flt = np.exp(-0.5*(wvs[:gab_lim]/(2.0*np.pi) - 125.0)**2.0/(25.0**2.0))
    flt = 1.0
    gabor_sg[i,:] = np.multiply(gab, flt)
#Plot to check
#ax.plot(tbar, weighted_y, 'o')
#ax.plot(freqs, np.absolute(gabor_sg[i,:]), 'o--')
#ax.set_ylim([np.amin(data), np.amax(data)])
#plt.pause(0.001)
#ax.clear()

# Meshgrid

```

```

T, W = np.meshgrid(t_ds, wvs_wig/(2.0*np.pi))
TG, WG = np.meshgrid(t_ds, wvs[gab_lim0:gab_lim1]/(2.0*np.pi))

#### Spectrograms
ga = np.log(np.absolute(gabor_sg).transpose() + 1.0)
#ga = np.absolute(gabor_sg).transpose()
Mg = np.amax(ga)
cbg = np.linspace(np.amin(ga)/Mg, np.amax(ga)/Mg, num=75)

#### Gabor
plt.figure()
plt.contourf(TG, WG, ga/Mg, cbg, cmap='Greys')
plt.xlabel(r'Time $t$ [s]')
plt.ylabel(r'Frequency $\omega$ [Hz]')
plt.colorbar()
plt.title(r'Gabor, $\log(|\psi| + 1)$')

plt.show()

#### Wigner
# Absolute
wa = np.absolute(wig)
Ma = np.amax(wa)
# Real
wr = np.real(wig)
#Mr = np.amax(wr)
cbr = np.linspace(np.amin(wr)/Ma, np.amax(wr)/Ma, num=75)
# Imag
wi = np.imag(wig)
cbi = np.linspace(np.amin(wi)/Ma, np.amax(wi)/Ma, num=75)

plt.figure()
plt.contourf(T, W, wr/Ma, cbr, cmap='Greys')
plt.xlabel(r'Time $t$ [s]')
plt.ylabel(r'Frequency $\omega$ [Hz]')
plt.colorbar()
plt.title('Real part')

#### Gabor-Wigner
gw = np.multiply(np.absolute(wig[:,2,:]), np.absolute(gabor_sg.transpose()))

gwa = np.log(np.absolute(gw) + 1.0)
#gwa = np.absolute(gw)
Mgw = np.amax(gwa)
cbgw = np.linspace(0.05, 1.0, num=75)

plt.figure()
plt.contourf(TG, WG, gwa/Mgw, cbgw, cmap='Greys', extend='both')
plt.xlabel(r'Time $t$ [s]')
plt.ylabel(r'Frequency $\omega$ [Hz]')
plt.title('Combined distribution $\log(|\psi_{\{GW\}}| + 1)$')
plt.colorbar()

plt.show()

```