

Energy spectral density heatmaps with meaningful units using LTFAT

Pierre St-Laurent (pst-laurent@vims.edu)

June 28, 2021

1 Introduction

I wrote down my notes on how to obtain an Energy Spectral Density heatmap (ESD; *e.g.*, *Emery and Thomson*, 2001, their §5.6.1) with time on the horizontal axis, frequency on the vertical axis, and a colorscale with units of $\text{amplitude}^2 \text{ frequency}^{-1}$. The latter is not always easy to obtain; many methods will produce a colorscale with relative units (*e.g.*, decibels) that are meaningful within a specific ESD, but not comparable from one ESD to another.

The ESD would satisfy Parseval’s theorem that relates the energy of the timeseries in the time domain to the energy of the ESD after integrating over frequencies. Energy is here equated to the squared amplitude of the timeseries in analogy with a simple harmonic oscillator. For a discrete timeseries y_n with $n = 1, \dots, N$ and sampling period Δt , Parseval’s theorem would be (*Emery and Thomson*, 2001, their Eq.5.6.13):

$$\Delta t \sum_{n=1}^N |y_n|^2 = \frac{1}{N \Delta t} \sum_{k=0}^{N-1} |Y_k|^2, \quad (1)$$

where Y_k is the discrete Fourier transform of y_n and has units of $\text{amplitude} \times \text{time}$:

$$Y_k = \Delta t \sum_{n=1}^N y_n \exp(-i 2\pi f_k n \Delta t), \quad (2)$$

where $f_k = k (N \Delta t)^{-1}$ and $k = 0, \dots, N$. With the Parseval theorem above, the energy of a signal $y(t) = \sin(2\pi f t)$ would be proportional to its length N . It is convenient to rewrite the theorem as:

$$\frac{1}{N} \sum_{n=1}^N |y_n|^2 = \frac{1}{N^2 \Delta t^2} \sum_{k=0}^{N-1} |Y_k|^2, \quad (3)$$

where the left-hand side is now weighted by N^{-1} and, in cases such as $y(t) = \sin(2\pi f t)$, will yield the same energy regardless of the timeseries’ length.

2 ESD heatmaps using LTFAT

The Large Time/Frequency Analysis Toolbox (LTFAT, *Průša et al.*, 2014; *Søndergaard et al.*, 2012, available at <http://ltfat.github.io/>) is “a Matlab/Octave toolbox for working with time-frequency analysis and synthesis”. It is one out of multiple toolboxes with similar purposes but has the benefits of being reasonably well-documented and easy to obtain.

LTFAT’s function `sgram` uses a short-time Fourier transform to compute the time-varying transform of a discrete timeseries y_n . For simplicity, we focus on a simple periodic timeseries (*e.g.*, $y(t) = \sin(2\pi f t)$) where the Fourier transform Y_k is the same at all times. One of the complication of using `sgram` is that the function, by default, returns transform coefficients as a two-dimensional array (*i.e.*, as a function of time and frequency) with the array’s dimensions being close to 800×600 , regardless of the sampling frequency or length of the original timeseries y_n . The rationale behind this default behavior is that the output of `sgram` can be readily visualized on a computer monitor. However, in our case, some scaling becomes necessary so that the coefficients satisfy Parseval’s theorem (as formulated in Eq. 3) and be used for a ESD with meaningful units. The rest of this document shows my notes on how to achieve this in Matlab/Octave.

```

pkg load ltfat

% Create a time-axis with N points and a sampling period delt.
% The Nyquist frequency is 1/(2 delt) and the sampling frequency is 1/delt.

npts = 300; % Length of timeseries.
delt = 0.4; % Sampling period (s).
taxi = (0 : npts - 1) * delt; % Time-axis (s).

% Create a fictional timeseries y__n representing, say, water level in meters.
% The amplitude is 2.5m and the frequency is 0.25s-1.

y__n = 2.5 * sin( 2. * pi * 0.25 * taxi ); % (meters).

% Compute the coefficients from the short-term Fourier transform.
% The first argument is the timeseries, the second the sampling frequency.
% The third argument indicates that we want raw coefficients on a linear scale.
% By default, sgram plots their magnitude and returns them in a matrix (coef).

coef = sgram( y__n, 1. / delt, 'lin' );

% Scale the coefficients from sgram to obtain units of amplitude x time.

coef = coef * sqrt( 1. / (size( coef, 1 ) - 1) ) * npts * delt;

% Create time and frequency axes that match the dimensions of array coef as
% returned by function sgram. delf is the frequency stepsize corresponding
% to array coef.

t_sg = size( coef, 2 );
t_sg = (0 : t_sg - 1) / t_sg * npts * delt; % s.
f_sg = linspace( 0., 0.5 / delt, size( coef, 1 ) ); % s-1.
delf = f_sg(2) - f_sg(1); % s-1.

% Compute the energy spectral density, in units of amplitude2 / frequency.

esde = abs( coef ).^2 / (npts * delt)^2 / delf;

figure;
colormap( inferno );
imagesc( t_sg, f_sg, esde );
axis xy;
xlabel( 'Time (s)' );
ylabel( 'Frequency (Hz)' );
title('Energy spectral density');
hand = colorbar;
title( hand, 'Ampl.2 / Hz' );

% Index to the midpoint of the time-axis in the array returned by sgram:
midp = round( 0.5 * size( coef, 2 ) );

disp( 'Energy from signal in time domain (ampl2) = ' );
disp( num2str( sum( abs(y__n).^2 ) / npts ) );
disp( ' ' );
disp( 'Energy from signal in frequency domain (ampl2) = ' );
disp( num2str( sum( esde(:, midp) ) * delf ) );

```

3 Figures

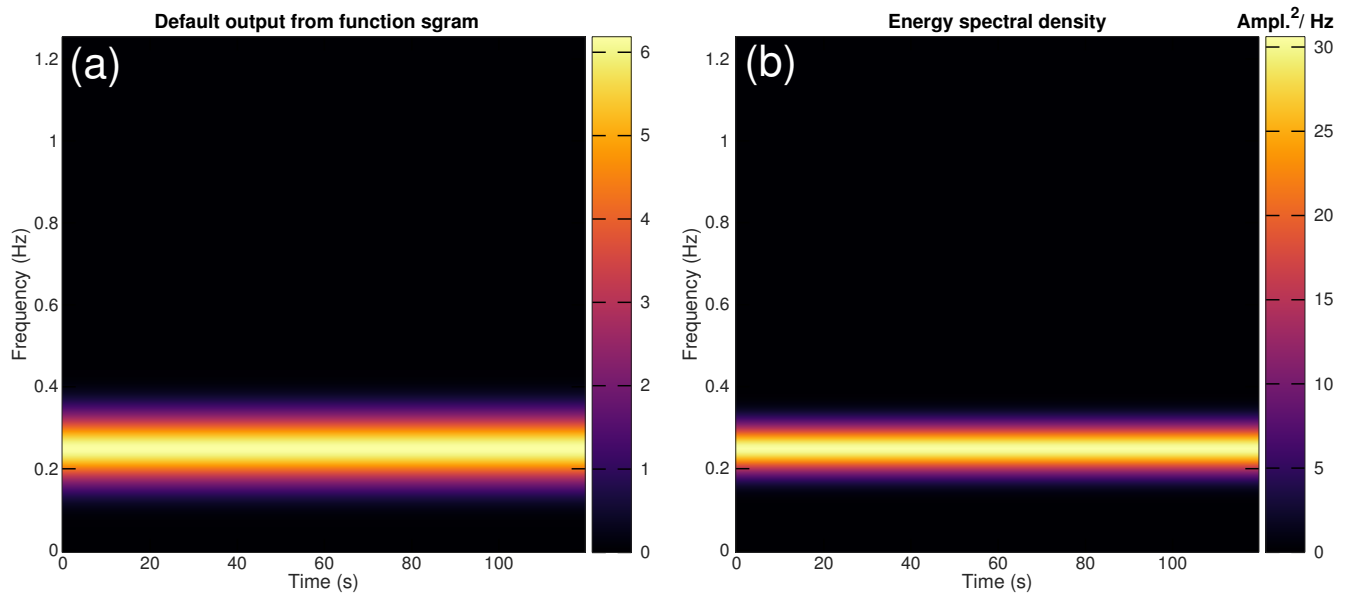


Figure 1: **(a)** Default output from LTFAT’s function `sgram` for the fictional timeseries y_n . The vertical axis (frequency) runs from zero to the Nyquist frequency. The horizontal axis covers the whole duration of the timeseries. The colorscale is the magnitude of the Fourier transform coefficients (`coef`) as returned by `sgram`. The goal is to scale these coefficients in such a way that their value satisfies Eq. 3. **(b)** Energy spectral density heatmap obtained after scaling the coefficients and computing $(N \Delta t)^{-2} |Y_k|^2 / \delta f$, where δf is the frequency interval in array Y_k .

References

- Emery, W. J., and R. E. Thomson (2001), *Data analysis methods in physical oceanography*, 2nd ed., 638 pp., Elsevier.
- Průša, Z., P. L. Søndergaard, N. Holighaus, C. Wiesmeyer, and P. Balazs (2014), The Large Time-Frequency Analysis Toolbox 2.0, in *Sound, Music, and Motion*, LNCS, pp. 419–442, Springer International Publishing, doi:10.1007/978-3-319-12976-1_25.
- Søndergaard, P. L., B. Torrèsani, and P. Balazs (2012), The linear time frequency analysis toolbox, *International Journal of Wavelets, Multiresolution Analysis and Information Processing*, 10(4), doi:10.1142/S0219691312500324.