

## SIGNAL AND IMAGE PROCESSING WITH FRACLAB

JACQUES LÉVY VÉHEL

*Projet Fractales, INRIA Rocquencourt, 78153 Le Chesnay Cedex, France*  
and

*IRCCyN, 1 rue de la Née, 44321 Nantes, France*  
e-mail: [jacques.levy-vehel@inria.fr](mailto:jacques.levy-vehel@inria.fr)

PIERRICK LEGRAND

*IRCCyN, 1 rue de la Née, 44321 Nantes, France*  
e-mail: [pierrick.legrand@irccyn.ec-nantes.fr](mailto:pierrick.legrand@irccyn.ec-nantes.fr)

We describe a software toolbox, called *FracLab*, for signal and image processing based on fractal and multifractal methods. *FracLab* allows both to compute a large number of (multi)fractal parameters on sampled data (dimensions, regularity exponents, multifractal spectra, ...) and to perform several basic tasks in signal processing (modelling, segmentation, denoising, synthesis, oversampling ...). We illustrate the use of *FracLab* on a numerical example, and give some new results for estimating fractal parameters in the course of the presentation.

### 1 Introduction and Motivations

Fractal and multifractal tools have found a large number of applications in recent years. They are increasingly used in areas including astronomy, medical image/signal processing, telecommunications, finance, speech processing and many more.

With the spread of fractal analysis in such diverse fields, it seems important that researchers and practitioners willing to make use of fractal tools dispose of a stable set of methods for computing e.g. fractional dimensions, correlation exponents or multifractal spectra. Such methods should both be thoroughly tested and up-to-date, so that they may serve as a common background to compare approaches and results. One aim of *FracLab* is to fulfill such a need, by providing an *open and free software toolbox*, composed of routines that can be tested, enhanced, and may serve as benchmark in various situations. *FracLab* currently contains roughly 800 hundred routines, contributed by various groups (check the *FracLab* distribution for a list of contributors). Further contributions as well as comments are welcome. A few dozens of research groups are known to use *FracLab* at this time. *FracLab* may be downloaded at <http://fractales.inria.fr> and [www.irccyn.ec-nantes.fr/hebergement/FracLab/](http://www.irccyn.ec-nantes.fr/hebergement/FracLab/).

A second aim of *FracLab* is related to an important and recent evolution in the use of fractal analysis: It has been realized that it is often beneficial to apply fractal tools to arbitrary (i.e. "non-fractal") signals. The best known example is probably fractal image compression based on IFS theory, as popularized in <sup>1</sup>: IFS-based compression allows to process any kind of images, without an assumption of "fractality". This is also the point of view adopted in *FracLab*: *FracLab* performs *fractal processing* of signals, rather than processing of *fractal signals*.

This approach should not be too surprising: Just as, e.g., gradient-based algo-

rithms are often successfully applied for image segmentation even when there are no mathematical or physical reasons for the original signal to possess an ordinary derivative, a fractal analysis may yield new insights for “non-fractal” data. More generally, *FracLab* proposes to use fractal analysis in exactly the same way as other mathematical tools are used in everyday signal processing: Under certain assumptions, one may always estimate a gradient from discrete data (for instance *via* a model, or by first regularizing it), or compute its Fourier transform (for instance by extending it in a proper way outside the observation domain). In the same way, *FracLab* computes fractional dimensions or multifractal spectra (these quantities are always defined) by making adequate assumptions. These will in fact be of the same nature as in classical signal processing: We shall assume that the underlying continuous signal belongs to a parametric class, or we shall “regularize” it in some sense (in scale rather than in space). Of course, a fractal analysis will not in general give useful indications when the signal is mainly regular or smooth. It will be of interest only if there is enough singularity in the data, and if the singularity structure bears important information. There are many cases where the irregular part of the observed data contains important information that cannot be processed if only the smooth part is kept. It can even be the case that most or all of the relevant information is carried in the singular structure of the observation. Let us give an example. Radar images are difficult to process because of the presence of a specific noise, the so-called *speckle*. However, speckle is not pure noise, but rather a genuine part of the signal, caused by the interferometric nature of radar images. In this respect, it contains information which is essential about the imaged region. Although removing the speckle can be useful for purposes of e.g. segmentation, analyzing it is a necessary task for other applications, for instance classification, simply because the smoothed signal does not contain the necessary information. From a broader point of view, one may even argue that, although many image processing techniques aim at getting rid of irregularities in the data, the segmentation of simple, non noisy optical images should more logically be based on singularity analysis: One is indeed mostly interested in singularities, since edges, for instance, are basically discontinuities in the grey levels. In that respect, most classical approaches, based on smoothing, do not appear as natural as is usually assumed.

The second aim of *FracLab* is then to help disseminate the use of fractal tools in the processing of irregular but arbitrary signals and images. This will allow to discover new situations where fractal analysis yields an interesting alternative to more classical signal processing tools.

The paper is organized as follows. Section 2 briefly reviews some of the main functionalities of *FracLab*. A new method for robust estimation of fractal parameters is detailed in section 3. The last section illustrates the use of *FracLab* on a simple example, namely the denoising of a synthetic aperture radar image.

## 2 Main functionalities of *FracLab*

As explained above *FracLab* may be approached from two points of view:

- Synthesis of fractal signals and computation of various fractal parameters.

- Signal and Image processing.

This separation is artificial in a sense, since the tools associated with these two streams overlap greatly, but it is conceptually helpful. Most functionalities can be accessed either from a fractal analysis or from a signal processing point of view.

In order to make FracLab user-friendly, a graphic interface is provided (see figure 1).

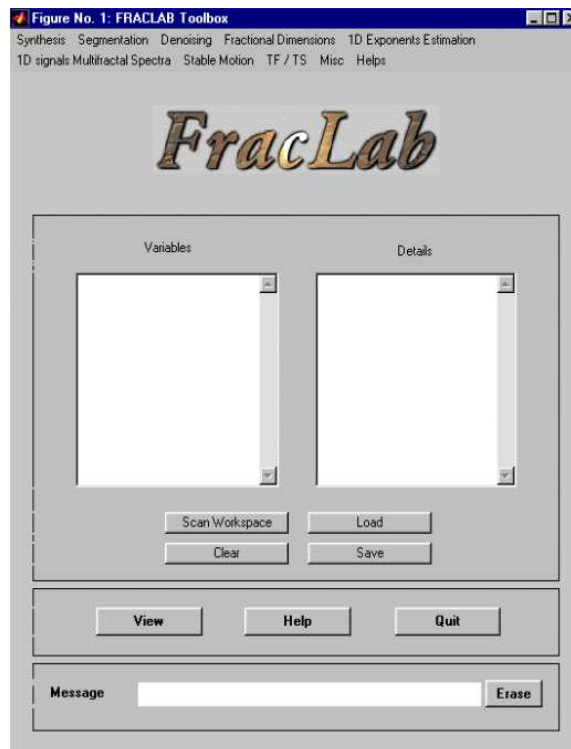


Figure 1. Graphical interface to FracLab

We briefly review the main functionalities in each mode.

### 2.1 Synthesis of fractal signals

Two types of signals can be generated : measures (i.e. arrays of non negative data that add to one) or functions. Measures are interesting in particular when one needs to take into account the resolution in an explicit way. For both measures and signals, either deterministic or stochastic data may be generated. By and large, *FracLab* allows to synthesize a substantial subset of all classical fractal models described in the literature : 1D and 2D fractional Brownian motions, multifractional Brownian motions, (generalized) Weierstrass functions, stable Lévy

motions, wavelet based 1/f process<sup>12</sup>, lacunary wavelet series<sup>7</sup>, 1D and 2D random multiplicative measures<sup>3</sup>, generalized Riesz products, ...

## 2.2 Fractal and Multifractal Analysis

The most basic parameters that can be computed are of course fractional dimensions. In the current implementation of *FracLab*, the box<sup>6</sup> and regularization<sup>11</sup> dimension are available. In many applications in signal processing, one is more interested in local characterizations of the data. Hölder exponents<sup>5</sup> are then more pertinent. A specific set of tools allows to estimate both pointwise and local exponents using various methods. In addition, a long range<sup>2</sup> or correlation exponent may be computed, as well as 2-microlocal exponents and frontiers<sup>10</sup>. Large deviation and Legendre Multifractal Spectra<sup>4</sup> may be computed through various estimations procedures. Finally, *FracLab* allows to test the Lévy-stability of a given process and to estimate the associated relevant parameters.

## 2.3 Signal and Image Processing

*FracLab* allows to perform segmentation of both 1D signals and images. In the former case, a modelling based on a generalization of IFS, called weakly self affine functions, is used. Images are segmented into edges or regions of given regularity through multifractal analysis. It is also possible to regularize and denoise 1D or 2D data using various methods based on Hölder regularity analysis or multifractal analysis. Finally, one may interpolate or oversample 1D and 2D data in a such a way that the evolution of various fractal features are controlled in the process.

## 3 Estimation of Fractal Exponents

We explain in this section a special feature of *FracLab* which is useful in the estimation of many fractal parameters. As is well-known, the definition of most fractal quantities involve upper or lower limits. For instance, the box dimension of a set  $E$  is defined as the limit, when it exists, of the ratio  $\frac{\log(N(\varepsilon))}{-\log(\varepsilon)}$  when  $\varepsilon$  tends to 0, where  $N(\varepsilon)$  is the number of "boxes" of size  $\varepsilon$  needed to cover  $E$ . When the limit does not exist, one uses upper and lower box dimensions, defined as the upper and lower limits of the previous ratio. These upper and lower dimensions always exist, but they do not coincide in general. In dealing with arbitrary signals, one often encounter the case where the box dimension is not defined, and one has to resort to the use of the upper and/or lower dimensions (as a matter of fact, the box dimension also does not exist for many fractal signals). Because the philosophy of *FracLab* is to be able to process any irregular signal, it is important to deal with this situation. In practice, in almost all applications, fractal parameters such as the box dimension are estimated through a linear regression: The estimated dimension is the slope of the least squares linear regression of the vector  $(\log(N(\varepsilon_1)), \dots, \log(N(\varepsilon_k)))$  versus the vector  $(\log(\varepsilon_1), \dots, \log(\varepsilon_k))$ , where  $(\varepsilon_1, \dots, \varepsilon_k)$  is a given decreasing sequence. Such a linear regression makes sense only if the box dimension exists as a limit. If the upper and lower limits are different, then no well-defined slope may be found in

the regression. However, it is still possible to estimate the upper and lower dimensions through a modified regression scheme, that we proceed to explain now. The use of these *liminf* and *limsup* regression methods is crucial, as it allows to attribute well-defined fractal quantities to arbitrary signals. This is of particular importance for local parameters such as the Hölder exponents. Indeed, even for fractal signals, the Hölder exponents are generally obtained as genuine lower limits (i.e. the limit does not exist).

Let  $(l_j)_{j \geq 1}$  be an arbitrary sequence of real numbers, and denote  $u_j = \frac{l_j}{j}$ . Let  $a = \liminf_{j \rightarrow \infty} u_j$ . In our frame, think for instance of  $l_j$  as the logarithm of the number of boxes in the computation of the box dimension. Define, for all  $n \geq 1$ :

$$E_n^0 = \{1, \dots, n\}$$

$$L_n^0 = \{l_1, \dots, l_n\}$$

Let  $(a_n^0, b_n^0)$  be the parameters of the least square regression of  $L_n^0$  with respect to  $E_n^0$ , i.e. the real numbers that minimize  $\sum_{j=1}^n (l_j - aj - b)^2$  over all couples  $(a, b)$ . We write :

$$(a_n^0, b_n^0) = \text{Reg}(E_n^0, L_n^0)$$

Let now:

$$E_n^1 = \{j \in E_n^0, l_j \leq a_n^0 j + b_n^0\}$$

$$L_n^1 = \{l_j, j \in E_n^1\}$$

$$(a_n^1, b_n^1) = \text{Reg}(E_n^1, L_n^1)$$

Define recursively:

$$E_n^i = \{j \in E_n^{i-1}, l_j \leq a_n^{i-1} j + b_n^{i-1}\}$$

$$L_n^i = \{l_j, j \in E_n^i\}$$

$$(a_n^i, b_n^i) = \text{Reg}(E_n^i, L_n^i)$$

for all  $i = 2, \dots, N_n$ , where  $N_n$  is defined as the first index such that  $\#E_n^{N_n+1} < 2$ .

The geometrical interpretation of the sequence  $(a_n^i, b_n^i)$  is simple: In the first step, we keep in  $(E_n^1, L_n^1)$  those points that are “below” the regression line of  $L_n^0$  with respect to  $E_n^0$ . We then compute the regression line of  $L_n^1$  with respect to  $E_n^1$  to obtain  $(a_n^1, b_n^1)$ , and iterate the process until at most one point remains below the regression line.

**Proposition 1.** *Let  $(l_j)_{j \geq 1}$  be an arbitrary sequence of real numbers, and define  $u_j$ ,  $a$ ,  $a_n^i$  and  $N_n$  as above. Then:*

$$\lim_{n \rightarrow \infty} a_n^{N_n} = a$$

The proof of this proposition may be found in<sup>9</sup>.

Let us put this theoretical result to work on a numerical example. We consider the following sequence:  $y_j = (2 + 2|\cos(j)|)j + 5 + 20 * \epsilon_j$ , where  $(\epsilon_j)$  is a sequence of i.i.d. gaussian random variables with mean 0 and variance 1. We deal with a sequence of  $n = 100$  points.

Figure 2 shows the deterministic signal  $x_j = (2 + 2|\cos(j)|)j + 5$  (in black) and its noisy version  $y_j = (2 + 2|\cos(j)|)j + 5 + 20 * \epsilon_j$  (in blue). Figure 3 shows the noisy signal (in blue), the associated regression line computed in the standard fashion (in blue), the line  $y = 4 * x$  (in red), and the limsup regression line obtained following the algorithm described in proposition 1 (in black). The standard regression line has slope 3.21, as the limsup one has slope 3.87, which is not to far from the theoretical superior limit 4.

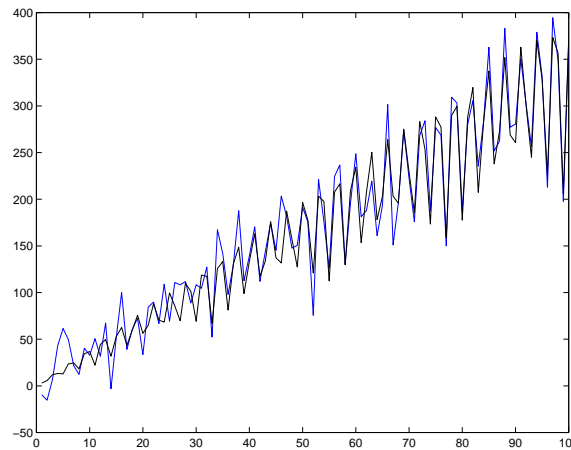


Figure 2. Original signal  $y_j = (2 + 2|\cos(j)|)j + 5$  (black) and its noisy version (blue)

Remark that this limsup/liminf device has in fact to applications:

- it allows to estimate in a robust way fractal quantities on arbitrary signals, by computing for instance the upper and lower box dimensions,
- it allows to check the "fractality" of signal: If the upper and lower limits coincide, then there is a (say) well defined box dimension or correlation exponent.

#### 4 Example of a processing with FracLab: Synthetic Aperture Radar image denoising

Synthetic Aperture Radar (SAR) images are generally difficult to read and to analyze because they contain a large amount of a specific noise, called speckle. Dozens of methods have been proposed to enhance their quality. Some use precise knowledge about, e.g., the statistics of the noise, while other are rather generic. The

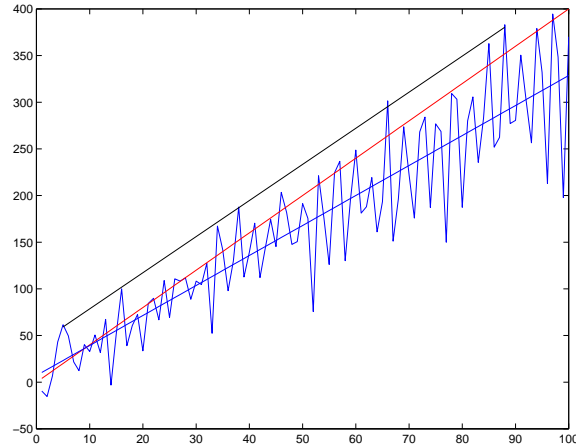


Figure 3. Noisy signal  $y_j = (2 + 2|\cos(j)|)j + 5 + 20 * \epsilon_j$  (blue), standard regression line (blue), line  $y = 4 * x$  (red), and lim sup regression line (black)

fractal denoising method is based on the following simple observation: Consider an image  $I$ , and its noisy version  $J$ . Pick a particular location  $(x, y)$  at random in the image. Then, chances are that the local regularity of  $I$  around  $(x, y)$  will be larger than the one of  $J$ . Of course, this statement is rather imprecise if we do not define how we measure regularity. In practice, this is done using some kind of Hölder exponent. Without going into details, we will admit the intuitive fact that adding noise decreases the local regularity at all points. Denoising can then be performed by increasing in a controlled way the Hölder exponent. This is exactly how the fractal methods implemented in *FracLab* work. More details may be found in<sup>8</sup>.

As an illustration, we display an original image SAR image on figure 4, along with its fractal denoising on figure 5. As one can see, the original image appears very noisy, and does not seem to hold any useful information. However, this is not quite true, as this scene does contain a river flowing from the top of the image and assuming roughly an inverted "Y" shape, as is apparent from figure 5. Denoising is used here as a pre-processing step that will enhance the image so that it will be possible to detect automatically the river. Such a procedure is used by the IRD, a French agency, which, in this particular application, is interested in monitoring water resources in this region of Africa.

The numerical experiment we have described in this section is part of the demonstration file of *FracLab*.

## References

1. M. BARNSELY, *Fractals Everywhere*, Academic Press, New-York, **1988**.
2. J. BERAN, *Statistics for Long-Memory Processes*, Chapman and Hall, New York, **1994**.
3. J. BARRAL, *Continuity of the multifractal spectrum of a random statistically*

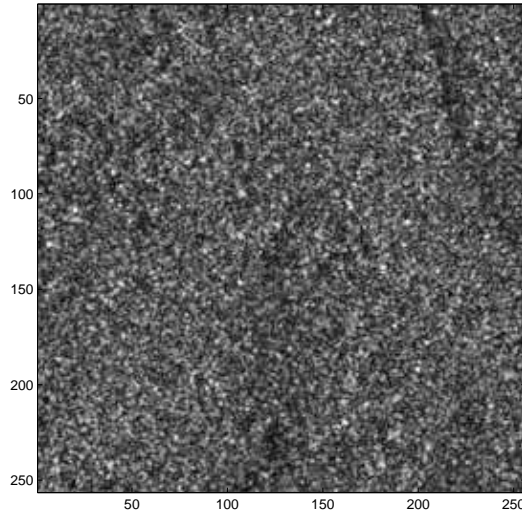


Figure 4. Original SAR image (courtesy IRD).

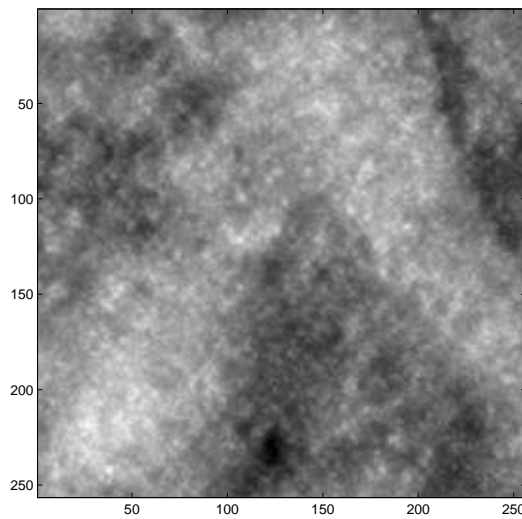


Figure 5. Image denoised through Hölder regularity analysis using *FracLab* .

- self-similar measures, *J. Theor. Probab.*, 13, pp 1027–1060, **2000**
4. G. BROWN, G. MICHON, AND J. PEYRIÈRE, On the multifractal analysis of measures, *J. Statist. Phys.*, 66 (3-4), pp 775-790, **1992**.
  5. K. DAUDI, J. LÉVY VÉHEL, Y. MEYER, Construction of Continuous Functions with Prescribed Local Regularity, *Constructive Approximation*. Vol. 14. Num. 3, pp.349-386, **1998**.



6. K.J. FALCONER, *Fractal Geometry: Mathematical Foundations and Applications*, John Wiley, New York, **1990**.
7. S. JAFFARD, On lacunary wavelet series, *Annals of Applied Probability*, **2000**.
8. J. LÉVY VÉHEL, *Signal enhancement based on Hölder regularity analysis*, IMA Volumes in Mathematics, **2002**.
9. J. LÉVY VÉHEL, P. LEGRAND, *Signal Processing with FracLab*, Inria Research Report, **2003**.
10. J. LÉVY VÉHEL, S. SEURET, The two-microlocal formalism, PSPUM, to appear, **2003**.
11. F. ROUEFF, J. LEVY VÉHEL, A regularization approach to fractional dimension estimation, *Fractals conference*, Malta, **1998**.
12. G.W. WORNELL, Wavelet-Based Representation for the  $1/f$  Family of Fractal Processes, *Proc. IEEE*, 81 (10), pp 1428-1450, **1993**.