# FRACTAL DIMENSIONS

Originally the following fractal dimension (FD) ideas were formulated and implemented by Ieva Kizlaitienė. She has prepared Master thesis „Fractal Modeling of Speech Signals“[[1]](#footnote-1) with 5 originally proposed fractal dimension algorithms.

All these FD algorithms were implemented in MATLAB (Version 2020, namely). This document introduces FD algorithm ideas, analytical expressions, and describes the use of the implemented software. The document uses the original names of the FD algorithms, suggested by the author Ieva Kizlaitienė.

The FD algorithm description consists of 2 parts:

* Methodological part. This part presents the FD idea and analytical expression for calculation. The material is taken from the original source1.
* Implementation part. This part presents the software implementation for the FD calculation, the input and output arguments lists, shares some remarks of performance of the implementation.

## AMPLITUDE FRACTAL DIMENSION

### Methodological part

Amplitude (named after the signal value, i.e. amplitude) algorithm intends to evaluate the ratio between actual data value changes and maximal possible value change. We make an assumption, that the highest value of changes would be obtained if in each step data value changes from positive maximal to minimal negative or vice versa. In this case the fractal dimension value should converge to 2. Considering this, the Amplitude FD can be computed:

here is accumulated data value change

is the length of data sequence , is the *i*-th data value.

is maximal possible accumulated data change

### Implementation part

The algorithm was implemented by MATLAB function “AmplitudeFD”

fd = AmplitudeFD(data, mind, maxd).

Output:

* fd - the calculated fractal dimension value.

Input:

* data - input data. If 'data' is a column vector, the function returns the calculated fractal dimension value. If 'data' is a matrix, the columns are treated as column vectors and the function returns fractal dimension value for each column.
* mind - minimum possible data value. If no value provided, the actual minimum data value is used.
* maxd - maximum possible data value. If no value provided, the actual maximum data value is used.

In case of constant data (i.e. the data contains one unchanging value), the algorithm results in division by zero, the MATLAB function returns NaN.

## DISTANCE FRACTAL DIMENSION

### Methodological part

This algorithm is similar to the Amplitude FD algorithm including the following differences:

* Data argument value are taken into account (as in Katz algorithm[[2]](#footnote-2)).
* The data sequence is decimated (as in Higuchi algorithm[[3]](#footnote-3)).

FD value is calculated:

here is accumulated data value change

with argument step (which can be assumed to be equal to 1 in case of measureless argument), is the *i*-th value of data sequence .

is defined

with newly defined data sequence

Data decimation (into subsets) should allow to capture and exploit peaks values in subsets.

### Implementation part

The algorithm was implemented by MATLAB function “DistanceFD”

fd = DistanceFD(data, n).

Output:

* fd - the calculated fractal dimension value.

Input:

* data - input data. If 'data' is a column vector, the function returns the calculated fractal dimension value. If 'data' is a matrix, the columns are % treated as column vectors and the function returns fractal dimension value for each column.
* n - the length of the segment. If no value is defined, the 'n' value is equated to the length of the analyzed data sequence.

The implemented source contains calls of specific MATLAB functions: buffer, circshift.

## SIGN FRACTAL DIMENSION

### Methodological part

Sign algorithm idea is to evaluate the meandering and the waviness of the analyzed data variation. For this purpose, a straightforward algorithm is proposed: to compare the number of local maximum and minimum points to the maximum hypothetical number of local extremum points. The FD value is calculated:

here is the number of local extremum (minimum and maximum) points, is the number of data points.

### Implementation part

The algorithm was implemented by MATLAB function “SignFD”

fd = SignFD(data).

Output:

* fd - the calculated fractal dimension value.

Input:

* data - input data. If 'data' is a column vector, the function returns the calculated fractal dimension value. If 'data' is a matrix, the columns are treated as column vectors and the function returns fractal dimension value for each column.

## LINEAR REGRESSION INTERSECTION FRACTAL DIMENSION

### Methodological part

The idea of linear regression (LR) intersection dimension is to evaluate data irregularity by computing the number of intersection points with fitting linear regression. For this purpose, the data sequence is presented as a curve with linear approximation added. The basis of this idea is the expectation that more complex curve will have more intersection points with line and vice versa.

Let be a fitting line of a curve and let , where satisfies condition . Then

Where is the number of intersections, is the length of analyzed data sequence.

### Implementation part

The algorithm was implemented by MATLAB function “LintersectFD”

fd = LintersectFD(data).

Output:

* fd - the calculated fractal dimension value.

Input:

* data – input data. If 'data' is a column vector, the function returns the calculated fractal dimension value. If 'data' is a matrix, the columns are treated as column vectors and the function returns fractal dimension value for each column.

## Polynomial intersection fractal dimension

### Methodological part

As linear regression model might be insufficient to represent all irregularities (for example, if data representing curve has preeminent peaks), a polynomial regression was applied and similarly as in LR fractal dimension number of intersection points was computed.

Let be a fitting polynomial regression of a data representing curve and let , where satisfies condition . Then intersection fractal dimension could be calculated in the same way as in linear regression case. In order to have more comprehensive results, logarithms were applied:

### Implementation part

The algorithm was implemented by MATLAB function “PintersectFD”

fd = PintersectFD(data).

Output:

* fd - the calculated fractal dimension value.

Input:

* data - input data. If 'data' is a column vector, the function returns the calculated fractal dimension value. If 'data' is a matrix, the columns are treated as column vectors and the function returns fractal dimension value for each column.
* order - maximal polynomial order for approximation. If not specified, the 18th order approximation is used (as defined by the author).

1. I. Kizlaitienė. „Fractal Modeling of Speech Signals“, Master thesis, Vilnius University, 2021. Available at: <https://epublications.vu.lt/object/elaba:81590289/> [↑](#footnote-ref-1)
2. M. J. Katz. „Fractals and the Analysis of Waveforms“, Computers In Biology and Medicine, 1988, Vol. 18, No. 3, pp. 145-156. [↑](#footnote-ref-2)
3. T. Higuchi. „Approach to an Irregular Time Series on the Basis of the Fractal Theory“, Physica D: Nonlinear Phenomena, 1988, Vol. 3, No. 2, pp. 277-283. [↑](#footnote-ref-3)