Modeling of multiscale time series using fractal splines

Olga Novikova[1], Volodymyr Shutko[2]

[1]Electronics Department, National Aviation University, UKRAINE, Kiev, Kosmonavta Komarova ave. 1, E-mail: novikovaolga88@gmail.com

[2]Electronics Department, National Aviation University, UKRAINE, Kiev, Kosmonavta Komarova ave. 1, E-mail: vnshutko@mail.ru

***Abstract* – A method of generating fractal data for simulation and natural shapes reconstruction is presented. The algorithm of fractal interpolation using splines is listed. Fractal splines allow building a complex curve manipulating a small amount of parameters and keeping calculations efficiently. An example of meteorological time series renewal is shown.**

Key words – spline, fractal spline, fractal interpolation, time series, multiscale analysis.

I. Introduction

The modern approach in studying complex natural and social systems states that they function over several levels, what is manifested in the fluctuations on a wide range of time scales. For example, at the currency market there are investors dealing with hour, day, week, month contracts; temperature time series maintain the property of inherent periodicity of the data within years, months and even hours of the day. The detailed overview of fractal time series, their properties and models can be found in [1].

To examine such systems multiscale analysis is used. It is based on the data representation with varying degrees of detail. This allows studying the global features of the data on a large-scale representation and detailing the local features on a smaller scale. In order to observe fractal and multifractal behavior several tools have been developed. The most popular one is wavelet-transformation [2]. We offer fractal splines for this purpose.

II. Fractal splines modeling

Fractal spline is a continuous polynomial function that is piecewise-defined and self-similar. Cubic fractal spline is also a smooth function. It is fully characterized by knots location and proportion between them, order and scale. To smoothly join the fragments on the nested scales, the initial spline should be periodic.

Self-similarity is achieved in a recursive way of constructing. On the zero scale classical linear or cubic spline is built. Spline of the *i*-th scale is a shifted and scaled copy of the zero-scale spline:

$$S_i(x) = m_i S_0(x-t)\ , \qquad (1)$$

where m_i is a scaling factor. Scaling factor reflects the proportion between fragment and whole spline lengths. In general one fragment is replaced by one scaled spline copy, but multiple replacements are also possible. To do this, divide scaling factor on number of times.

Fractal spline of the *k*-th scale is the sum of splines of scales from zero to *k*:

$$^{kf}S(x) = \sum_{i=0}^{k} \alpha_i S_i(x)\ , \qquad (2)$$

where $\alpha_i \in [0;1]$ is a weight coefficient. In particular, the case $\alpha_i = 0$ means the exception of fractal spline of the *i*-th scale from the decomposition, and the case of $\alpha_i = 1$ stays for full inclusion in the decomposition.

Points of fractal spline of some scale *k* form fractal time series. This occurs through applying the algorithm of fractal interpolation to some set of points defining the breaks in generated time series.

Suppose there is a set of points $\Delta = \{X, Y\}$, where $X = \{x_i\}_{i=1}^N$, $Y = \{y_i\}_{i=1}^N$ and $x_i > x_{i-1}$ for all $i = 2,3,...,N$. Interpolation function ^{kf}S is called fractal spline, if the following conditions are satisfied:

1) $^{kf}S(x)$ is a linear or cubic polynomial for all $x \in [x_{i-1}, x_i]$;
2) $^{kf}S(x_i) = y_i$ for all $i = 1,...,N$;
3) $^{kf}S(x) = k^{-\alpha} S_0(kx)$, where α is a self-similarity coefficient, $k > 0$ – scale.

For fractal spline of the third order a further condition is additionally imposed:

4) $^{kf}S'(x)$, $^{kf}S''(x)$ are continuous on X.

The algorithm of fractal spline interpolation includes the next steps:

1. Specify interpolation points and mode of basis spline function (fractal generator).
2. Define the maximum acceptable nested scale under fixed step between data.
3. For given interpolation points and chosen step build a cubic (linear) interpolation spline. It's a zero scale spline.
4. Divide first fragment of spline into sub-fragments according to the proportion of the knots of fractal generator.
5. Calculate spline with the same parameters over obtained sub-fragments.
6. Repeat steps 4-5 for all fragments of the zero scale spline.
7. Add values of the zero scale spline and the first scale spline. If necessary, weigh the last one by some coefficient *α*.
8. Repeat steps 4-7 for all scales till maximum.

The number of nested scales is limited by the length of the shortest fragment of the zero scale spline (if knots are distributed irregularly) and the amount of its knots. Interpolation points constitute a general trend of fractal data, and if this set is large, it is inefficient to use them as knots of fractal generator also. In this case dimension of planning matrix increases, meanwhile the quantity of nested scales drops. Better way is to determine fractal generator parameters separately, for example divide interpolation set on parts or define independent knots for spline being replaced on fragments. Though the same work is made twice (calculation of the zero scale spline and spline-replacer), but the advantage is more precise control over curve's shape and faster calculations.

Shortly, it’s more efficient to built consistently 5-knots-fractal spline twice than 10-knots-fractal spline once.

Let simulate fractal time series, which contains 10 given points: (0,0), (10, 1), (20, 0.25), (30, 0.5), (40, -0.5), (50, 0.75), (60, 0), (70, 0.25), (80, 0.8), (90, 0). They are marked by circles in Fig. 1. Fractal function generator has 5 points non-uniformly distributed (indicated by dashed line). Consistently interpolate these points by fractal spline of the first, second and third scale (Fig. 1).

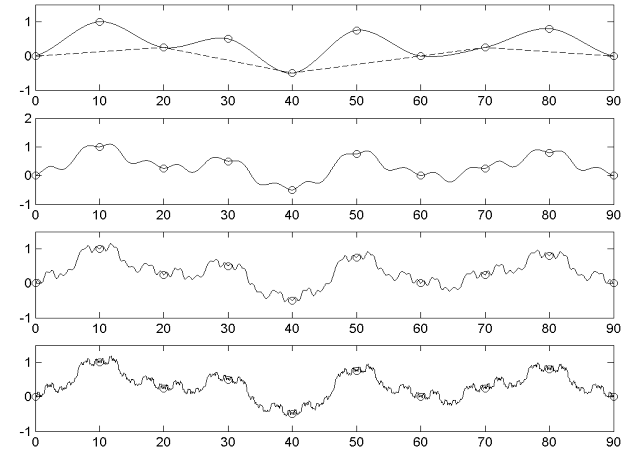

Fig. 1. Fractal spine interpolation of scales 0-3

If increase some region of fractal spline of the third scale, one can see that it recreates the fractal splines of previous scales and multiply repeats the shape of basis fractal function.

Thus, a sophisticated self-similar curve can be produced operating a small number of parameters.

Fractal splines application for curve approximation and signal generation can be found in [3].

III. Modeling of natural time series

Let's try to recreate the shape of natural multiscale time series. Meteorological observations are a good example of such a data, because they hold self-similarity and internal cycling. It can be observed some relations between temperature and season, month, week and even hours of the day.

Consider time series of average temperature in Kiev during 01.01.2012-31.12-2012 (marked as gray line in Fig. 2).

Natural curves are irregular and asymmetric. To recreate such a curve, preliminary analysis is needed. In this case we divide curve on two parts by point 34 (in the graph it corresponds to temperature drop in the beginning of February). Right part is bell-shaped and can be well described by Takagi-like fractal. To build it, we select 3 knots X = [34,179,366] (marked by circles) that form triangular fractal generator. To ensure the condition of periodicity values of the first and last knots were replaced by their mean value.

On the right part fractal spline of the sixth scale was built, on the left out of the lack of data – fractal spline of the fourth scale. Both fractal splines share the same basis spline. To bring the points of fractal splines and initial data closer, we have multiplied its fragments on individual vertical scaling factors.

Because time series has acute fluctuations not requires smoothing, so basis spline of the first order (linear) was chosen.

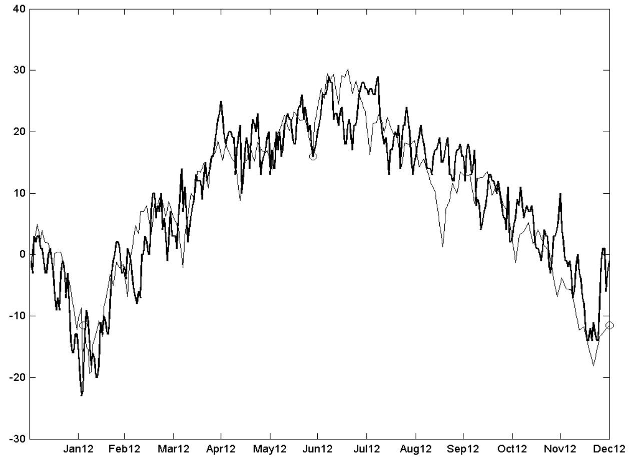

Fig. 2. Recreating the time series of daily temperature

Figure 2 shows fractal spline in black line. It follows well the shape of process and re-creates the inner loops. Division of time series on two parts allows to replicate even the anomalous temperature drop. The lack of accuracy in passing the exact values can be explained due to random fluctuations in time series.

Time series generated by fractal splines can be used for simulating test data for different fractal models, constructing realistic natural curves in computer graphics and generating fractal functions for technical applications.

Conclusion

Fractal spline can simulate multiscale time series of given form or follow the shape of the natural fractal time series. Very complex curves are uniquely determined by a small number of parameters and inherent spline simplicity of calculation keeps. Also it is possible to interpolate non-smooth and nowhere differentiated functions.

References

[1] J. W. Kantelhardt, “Fractal and multifractal time series,” Mathematics of Complexity and Dynamical Systems, pp.463-487, 2011.

[2] P. Abry, P. Goncalves, J. Levy Vehel, Scaling Fractals and Wavelets, Wiley, 2009.

[3] O.B. Novikova, "Aproksymatsiia fraktalnykh kryvykh za dopomohoiu splainiv” [“Approximation of fractal curves using splines"], ASU ta prylady avtomatyky – ICS and automation devices, vol. 158, pp.63-68, 2012.

[4] P. Manousopoulos, V. Drakopoulos, T. Theoharis, “Curve fitting by fractal interpolation,” Transactions on Computational Science, vol. 1, pp.85-103, 2008.