A Stochastic Iterative Amplitude Adjusted Fourier Transform Algorithm with improved accuracy

Victor Venema[1], Felix Ament[1], and Clemens Simmer[1]

[1]{ Meteorologisches Institut, Universität Bonn, Germany}

Correspondence to: Victor Venema (Victor.Venema@uni-bonn.de)

Abstract

A stochastic version of the Iterative Amplitude Adjusted Fourier Transform (IAAFT) algorithm is presented. This algorithm is able to generate so-called surrogate time series, which have the amplitude distribution and the power spectrum of measured time series or fields. The key difference between the new algorithm and the original IAAFT method is the treatment of the amplitude adjustment: it is not performed for all values in each iterative step, but only for a fraction of the values. This new algorithm achieves a better accuracy, i.e. the power spectra of the measurement and its surrogate are more similar. We demonstrate the improvement by applying the IAAFT algorithm and the new one to 13 different test signals ranging from rain time series and 3-dimensional clouds to fractal time series and theoretical input. The improved accuracy can be important for generating high-quality geophysical time series and fields. The traditional application of the IAAFT algorithm is statistical nonlinearity testing. Reassuringly, we found that in most cases the accuracy of the original IAAFT algorithm is sufficient for this application.

1 Introduction

The Iterative Amplitude Adjusted Fourier Transform (IAAFT) algorithm was developed by Schreiber and Schmitz (1996, 2000) to generate surrogate time series for statistical nonlinearity testing (Theiler et al., 1992; Theiler and Prichard, 1996; Kugiumtzis, 1999). Surrogates are time series which share certain statistical properties with the original time series. In case of the IAAFT algorithm, the surrogates share their distribution and power spectrum with the measurement. To stress that the surrogate is a permutation of the original, i.e. that the values of the original are reproduced exactly, the term amplitude distribution is preferred over probability density function, but in this article we will also simply use the term distribution.

Besides nonlinearity testing, the IAAFT algorithm is applied to generate realistic geophysical fields. It is, for example, not possible to measure a full 3-dimensional cloud field, but one can simulate a surrogate cloud field based on estimates of the distribution and power spectrum from a limited measurement (Venema et al., 2006). Surrogate fields can also be used as idealised boundary conditions for dynamical models. For applications where the distribution is equally important as the structure, IAAFT surrogates could be useful instead of (multi-)fractal time series or fields. Furthermore, the algorithm is a practical method to generate time series with interesting statistical properties for testing, e.g. analysis and error-detection, algorithms. We expect to see many more geophysical applications when the technique becomes better known in that community.

In the engineering community the IAAFT algorithm was recently discovered independently to simulate pressure fields from strong winds and offshore waves (Masters and Gurley, 2003). Masters and Gurley also compared the algorithm to older ones used in the engineering community and found the IAAFT algorithm to be more accurate. An example of such an older algorithm is the one proposed by Popescu et al. (1998) to generate fields of soil properties such as the elastic modulus and the mass density. Lewis and Austin (2002) used a similar algorithm to create fractal clouds with a lognormal distribution.

The structure of a cloud field is important for its radiative properties, e.g. for the reflection of sun light by clouds (Scheirer and Macke, 2001; Pincus et al., 2005). For this purpose it is important that surrogate fields capture this structure very accurately. Figure 1 shows the reflectance bias of surrogate cloud fields created with the IAAFT algorithm for sparse cumulus fields generated with a Large Eddy Simulation (LES) model. LES models are able to simulate atmospheric flow at spatial resolutions that are sufficient to resolve turbulent eddies. The IAAFT cumulus surrogates display some noise (wisps of cloud) in the cloud free sections, which biases their reflectance. The surrogate cumulus clouds generated with our more accurate Stochastic IAAFT (SIAAFT) algorithm do not have a bias; see Venema et al. (2006). This illustrates that the accuracy of the power spectrum can be important for geophysical applications. On the other hand, the accuracy is not always a limiting factor. In case of less demanding stratocumulus clouds a much lower accuracy of the power spectrum was high enough to avoid biases in the reflectance of their surrogates.

Another method to generate surrogates that are more accurate than the IAAFT surrogates is constrained randomization using global search algorithms such as simulated annealing (Schreiber, 1998). Unfortunately, Schreiber is unsure if his algorithm can be applied to statistical problems such as nonlinearity testing. The computation of large geophysical fields using simulated annealing will be very computationally expensive.

In Section 3, we present the SIAAFT algorithm which generates surrogates whose power spectra match the original power spectra more accurately than the surrogates generated with the original IAAFT algorithm. Furthermore, we demonstrate the accuracy of the SIAAFT algorithm by generating surrogates from a range of different inputs (Section 4). Before explaining the step from IAAFT to SIAAFT, the original IAAFT algorithm will be shortly reviewed (Section 2).

2 IAAFT Algorithm

The IAAFT algorithm is explained mathematically below for a time series and illustrated by its application to a cloud liquid water path (LWP) measurement in Figure 2. LWP is the vertically integrated amount of liquid water in a cloud. On our webpage (http://www.meteo.uni-bonn.de/venema/themes/surrogates/iaaft), the algorithms are further clarified using pseudo code; working Matlab versions for time series and fields can be downloaded.

The measured time series (Fig. 2, upper Panel) is denoted by the vector $\{m_n\}$, with the index n = {0,1,...,N-1}, and N the number of values of the original. The algorithm starts with a random shuffle of the data points (Panel 2). Then an iterative process is started as illustrated in the flow diagram (left side of Fig. 2). As explained later on, in each iteration, (i), the Fourier spectrum is adjusted first (Panel 3) and then the amplitudes (Panel 4). The time series after the spectral adjustment is called $\{x_{1,i}\}$; the 1 denotes that it is the time series after the first adjustment, the i denotes the current number of iterations. Similarly the time series after the amplitude adjustment is called $\{x_{2,i}\}$.

2.1 Spectral adjustment

Based on the original time series the power spectrum is estimated as

$$M_k^2 = \left| \sum_{n=0}^{N-1} m_n e^{i2\pi kn/N} \right|^2, k = 0,..., N-1. \tag{1}$$

The spectral adjustment starts by calculating the Fourier transform ($S_k^{'}$) of the time series after the last amplitude adjustment $\{x_{2,i-1}\}$ (or in the first iteration the Fourier transform of the initial white noise time series). The magnitudes of these Fourier coefficients are replaced by those of the original time series (M_k). The phases remain unaltered in this step. If we define $\varphi_k = S_k^{'} / \left| S_k^{'} \right|$, then the complex Fourier coefficients of $\{x_{1,i}\}$ are given by

$$S_k = \left| M_k \right| \varphi_k . \tag{2}$$

From S_k, $\{x_{1,i}\}$ is calculated using an inverse Fourier transform. Consequently, the time series in Panels 1 and 3 share the same power spectrum; the difference between their structures is due to differences in their distributions.

2.2 Amplitude adjustment

In the second step, the amplitudes are adjusted based on their ranking. A new time series is created with the values of $\{m_n\}$, but with the order given the ranking of the $\{x_{1,i}\}$-values. For example, the highest value of the iterated time series is substituted by highest value of the original time series. To perform this operation for all values, a sorted list is created of the values of the measured time series $\{ m_n^{'} \}$, where the prime denotes here that it is a sorted vector. Let the function rank() return the ascending rank number, i.e. return 1 for the highest number, 2 for the second highest, etc., then the amplitude adjusted time series is given by:

$$x_{2,i} = m_{rank(x_{1,i})}^{'} . \tag{3}$$

Since this amplitude adjustment will alter the power spectrum, both the amplitude and the spectral adjustments are repeated until a convergence criterion is reached (Panel 5).

3 A stochastic IAAFT algorithm

Schreiber and Schmitz (2000) suggested that one can start the IAAFT algorithm either with a random shuffle of the measurements (white noise) or with a surrogate created with the non-iterative and less accurate Amplitude Adjusted Fourier Transform (AAFT) algorithm (Theiler et al., 1992). We observed that the surrogates which were initialized with the AAFT surrogate started with a better fitting power spectrum, but finally obtained a lower accuracy, than the surrogates that were initialized with white noise; compare the dashed lines in Figure 3. Apparently a high initial quality can cause the algorithm to get stuck in a less accurate local minimum, because the algorithm favours solutions that are close to the first guess. Consequently, we looked for ways to slow down the convergence speed.

The SIAAFT algorithm differs from the IAAFT algorithm in three points. First of all we changed the way the amplitudes are adjusted. Because of this we also altered the convergence criterion and introduced two stages. These two stages differ in the fraction of amplitudes that is adjusted. See the flow diagram in Figure 4 for an overview.

Just as the standard version of the IAAFT algorithm, the SIAAFT algorithm starts with white noise. In the first stage only a fraction of the amplitudes is substituted instead of all of them. We have tested three different methods to adjust the amplitudes. The main method, which is used for all figures in this paper, is the ‘partially stochastic’ adjustment. This method calculates 5 vectors with equidistant indices, {{1, 6, 11, …}{2, 7, 12, …}{3, 8, 13, …}{4, 9, 14, …}{5, 10, 15, …}}, and each iteration selects randomly one of these 5 vectors. In this new amplitude adjustment only the amplitudes pertaining to the ranks in the indices are exchanged. The ‘deterministic’ adjustment uses the same equidistant indices. However it does not select one randomly, but uses a fixed pattern for the indices; the first index is given by: 1, 2, 3, 4, 5, 1, 2, …. The ‘fully stochastic’ version simulates drawing numbers without replacement. The differences in accuracy of the surrogates produced by these three methods are discussed in section 4.2.

After the first stage, the algorithm generates a surrogate where both amplitude distribution and spectrum fit closely, but none exactly. As we would like to have a perfect fit of the amplitude distribution, a second stage is applied where all amplitudes are adjusted, just as in the standard IAAFT algorithm. This second stage is initialised with the surrogate from the first stage.

The convergence criterion is based on the accuracy measure (Δ), which is the root mean square (RMS) difference of the Fourier Spectra of the original M_k and the surrogate $S_k^{'}$:

$$\Delta = \frac{1}{\sigma}\sqrt{\frac{1}{N}\sum_k \left(\left|M_k\right| - \left|S_k^{'}\right|\right)^2} \tag{4}$$

The measure is normalized by the standard deviation (σ) of the original time series for ease of comparison between various originals. Due to the stochastic nature of the algorithm, the last iterated time series is not always the best; see the noisy drawn line in Fig. 3. Thus, for the final result the *best-converged* surrogate is stored in a separate variable. The final accuracy depends on how many iterations one is willing to wait for the next surrogate with a better fitting Fourier spectrum. We use the exceedance of this number of iterations, which we call the iteration threshold, as termination criterion for both stages. The noisy line in Figure 3 depicts the accuracy of a SIAAFT surrogate as a function of the number of iterations. For this calculation we utilized an iteration threshold of 1000 iterations. The small peak near the maximum number of iterations (see the arrow in Figure 3) is the reduction in accuracy that is typical for the beginning of the second stage; the second stage is very short compared to the first stage.

The algorithm is not very sensitive to the fraction of values substituted in the amplitude adjustment. Figure 5 shows that the quality as function of the fraction of adjusted values has a broad minimum. Any value between a few percent and about 40 % will perform well. We use 20 % in the rest of this paper. The calculation time scatters strongly, but there is a tendency for fractions below 40 % to be faster. Except that adjusting all values (as the IAAFT algorithm does) is faster.

We tried a number of other ways to improve the accuracy of the SIAAFT algorithm. We found that the accuracy does not improve by making the spectral adjustment stochastic in a similar way as it was implemented for the amplitude adjustment. Analogous to the cooling scheme in simulated annealing (Vidal, 1993), we tried to gradually increase the fraction of amplitude adjustments in a small steps from 20 to a 100 percent. This did not lead to improvements. Calculating an ensemble of surrogates and selecting the best converged ones, is an important way to improve the accuracy. This is especially true for the IAAFT algorithm as its accuracy has a broad distribution; see Figure 6.

4 Results

To investigate the improvements by the SIAAFT algorithm, we calculated surrogates for 13 cases ranging from rain and clouds to theoretical statistics. These cases are described in section 4.1. From each of these original time series we produced 25 surrogates (except where indicated otherwise), and calculated the mean accuracy. All three methods for the amplitude adjustment of the SIAAFT algorithm (explained in Section 3) are assessed. In addition, we utilise two iteration thresholds: 100 and 10.000. To make the calculations well comparable, the results marked IAAFT were calculated with the SIAAFT algorithm, substituting all values in the amplitude adjustment. The results are summarized in Table 1 and described in section 4.2.

4.1 Measurements and statistics

- Fractal time series: As the first test case – a fractal time series – we used theoretical statistical input: an exponential amplitude distribution and a power law power spectrum with an exponent of -5/3, i.e. $S_k^2 \propto k^{-5/3}$. This exponent is typical for the inertial subrange of turbulence and stratiform clouds.
- Discrete Gaussian: With the same fractal power spectral shape of the first test function, we used a “discrete Gaussian” amplitude distribution. This distribution was computed starting with a Gaussian distribution with standard deviation 3 and rounding its values to integers.
- Positive Gaussian: The third case again had the same power spectral shape as the previous cases. Its PDF was constructed by starting with a Gaussian PDF, from which the negative instances were set to zero.
- Model cumulus clouds: 52 Cumulus cases were generated with a Large Eddy Simulation model. These simulations represent the diurnal cycle of cumulus over land, from no clouds in the morning, to a maximum of 16 % mean solar reflectance in the afternoon, to little, but thicker clouds in the evening (Brown et al., 2002). The clouds have a resolution of 100 m in the horizontal and 112 m in the vertical. The number of grid boxes is 66 by 66 horizontally. The model grid has 122 height levels, but the levels with clear air above and below the cloud have been removed. To maintain the vertical structure of the clouds, the amplitude adjustments are performed separately for every height level following Venema et al. (2006).
- Daily rain sums: As input for these surrogates we used daily rain sums for 5 different stations in Germany (Bremen, Hamburg-Fuhlsbuettel, Hohenpeissenberg, Karlsruhe and Potsdam) with a length of approximately 90 years (DWD, Deutscher Wetterdienst, Offenbach, Germany, http://www.dwd.de/en/FundE/Klima/KLIS). For every station 8 surrogates were calculated.
- River discharge: These surrogates were generated based on 3 stations (Cologne, Rees, and Lobith) with daily discharge measurements of the river Rhine (approximately 90 years; GRDS, Global Runoff Data Centre of the federal institute of Hydrology, Koblenz, Germany, http://grdc.bafg.de). For every station 8 surrogates were calculated.
- EEG (electroencephalogram): From the Department of Epileptology of the University of Bonn we used two EEGs (Andrzejak et al., 2001). The first is an EEG from within the

epileptic zone during an epileptic attack; the second one was recorded during a seizure free interval.

- Henon map: A chaotic time series was calculated using the Henon map (Kantz and Schreiber, 1999).
- Step-function: Especially the behaviour of the algorithms in case of bimodal distributions is interesting. One of the simplest cases of this type is a binary time series consisting of only ones and zeros. The first case considered, is a step-function signal with 512 ones, followed by 512 zeros.
- Random binary: As an example of a more random binary function we used a signal with 1024 numbers, with all values set to zero, except for the indices 10 to 25, 100 to 225, and 300 to 625, which were set to unity.
- Random sine: In a further test we created a time series by substituting the three blocks (with ones) of the previous test time series with the positive parts of sine functions, i.e. sin(x), with x=[0, π].

4.2 Accuracy

The largest improvement (see Table 1) of SIAAFT over IAAFT is found for the long fractal time series. The accuracy of the fully stochastic SIAAFT surrogates with an iteration threshold of 10.000 is almost an order of magnitude better than the accuracy of the IAAFT surrogates.

For the model cumulus clouds, the IAAFT algorithm obtained an accuracy of 5.8 10^{-4}. Of the 52 cloud fields 39 converged fully in one SIAAFT run, which is the main cause of the factor 5 improvement in accuracy. In these fully converged cases the surrogate was identical to the original LES clouds except for horizontal translations and/or reflections of the field. For completeness: also for the 13 LES clouds that did not converge in the first run, a fully converged surrogate could be generated by generating multiple SIAAFT-surrogates. Thus, in these 13 cases the SIAAFT algorithm got stuck in a local minimum.

In case of the step-function, the IAAFT algorithm achieves full convergence in 6 of 25 trials; the SIAAFT algorithm converged completely in all 25 trials. Because of computational error full convergence is defined as $\Delta < 10^{-10}$. The fully converged surrogates were a time shifted version of the original time series. The random binary case is more difficult; in this case the IAAFT algorithm never converged fully. The accuracy was around 1 %, which is the highest value of all cases considered. The fully stochastic SIAAFT algorithm converged better by a factor of two, and it converged fully in 7 of 25 trials. For the random sine case, the SIAAFT algorithm did perform a little, but not statistically significantly, better than the IAAFT algorithm. None of the surrogates converged fully.

It is interesting to note that in most cases the SIAAFT algorithm with deterministic amplitude adjustments performs worse than the two versions with stochastic adjustments. However, the deterministic version does perform better than the IAAFT algorithm. Thus, the improved performance of the SIAAFT algorithm is only partially due to its stochastic nature.

The fully stochastic version is more accurate for the long fractal time series than the other two SIAAFT versions. In most cases, however, the partially stochastic version of the SIAAFT algorithm is just as accurate as the fully stochastic version. Since the fully stochastic SIAAFT version is about 20 % slower in these cases, the partially stochastic version is our first choice, but it is advised to test the fully stochastic version for any new application.

4.3 Nonlinearity testing

An important application of surrogate modelling is nonlinearity testing. We compared the performance of the surrogates from the two algorithms numerically by calculating the root-mean-square nonlinear prediction error on time series with known dynamics. We used a standard nonlinear prediction algorithm (Small, 2005) without distinguishing between training and testing dataset.

For example, we generated an ensemble of linear time series by adding three sine functions with random wavelengths that were a multiple of the total length of the time series to get a periodic function without jumps. These time series were filtered by different static nonlinear functions: we discretised the values, or raised them to the third power or computed their absolute values. From these time series and their surrogates the nonlinear prediction error was calculated utilising six embedding dimensions and a delay time equal to the correlation length. In all cases both the IAAFT surrogates as well as the SIAAFT surrogates correctly identified the time series as linear. In addition, a number of standard nonlinear dynamical systems were combined with nonlinear static measurement functions. In none of these cases, we found a real advantage of the SIAAFT algorithm over the IAAFT algorithm for nonlinearity testing.

Nonlinearity tests can generate false answers in case the nonlinear prediction estimate is calculated with an insufficient embedding dimension (thus, care has to be taken for typically high-dimensional geophysical systems) and for time series with bimodal distributions. In the latter case, the scientist is warned by a (near) significant difference in the correlation of two consecutive points (Schreiber and Schmitz, 2000) between the original signals and the surrogates.

In some of the cases, the SIAAFT algorithm found the global minimum more often than the IAAFT algorithm. This prompts us to make a critical comment on nonlinearity testing with surrogates. One could argue that a fully converged surrogate has converged too far, that the algorithm could have produced a nonlinear surrogate and that it is not possible to use such a surrogate for nonlinearity testing. However, as soon as the Fourier phases are tuned, as the IAAFT and SIAAFT algorithms do, the time series is nonlinear and one cannot distinguish rigorously between a static and a dynamical origin of this nonlinearity. Thus, also without full convergence the surrogates are not fully guaranteed to represent a linear dynamical system filtered through a static nonlinear measurement function.

5 Summary

This paper presented a new algorithm to generate surrogate time series, which is a further development of the Iterative Amplitude Adjusted Fourier Transform (IAAFT) algorithm. This Stochastic IAAFT (SIAAFT) algorithm was tested on a number of test cases, e.g. various fractal time series, and EEG, runoff, cloud and rain measurements. In all test cases the SIAAFT algorithm achieved a higher accuracy than the IAAFT algorithm, i.e. the power spectrum of the surrogates was closer to that of the original time series. The largest improvement in accuracy, almost a factor 10, was found for a fractal time series. The algorithm also converged to a global minimum more often. That improvements in the accuracy are possible will hopefully inspire other researchers in finding even more accurate and efficient algorithms.

The main change in the algorithm is that the SIAAFT algorithm performs the amplitude adjustment only for a fraction of the values. This fraction can be selected between a few percent and 40 percent. We tested three different methods for the amplitude adjustment, which differ in the way the amplitudes to be adjusted are selected. The method called ‘partial random’ was shown to be a good compromise between accuracy and efficiency.

Reassuringly, we found that in most cases the accuracy of the IAAFT algorithm is sufficient for nonlinearity testing. However, for bimodal distributions, the accuracy of both algorithms can be insufficient.

Acknowledgements

We want to thank Sebastián Gimeno García for the Monte Carlo reflectance calculations and Klaus Lehnertz from Department of Epileptology of the University of Bonn for introducing us to surrogate time series and for the EEG time series. Furthermore, we are grateful to the Deutscher Wetterdienst for the rain data and the Global Runoff Data Centre for the river discharge data. This research was carried out in the framework of the 4D-clouds project, which is sponsored by the German ministry of research, BMBF, in the AFO2000 program on atmospheric research.

References

Andrzejak R.G., Lehnertz K., Rieke C., Mormann F., David P., Elger C.E., Indications of nonlinear deterministic and finite dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state, Phys. Rev. E., 64, 061907 (2001).

Brown, A.R., R.T. Cederwall, A. Chlond, P.G. Duynkerke, J.C. Golaz, M. Khairoutdinov, D.C. Lewellen, A.P. Lock, M.K. MacVean, C.H. Moeng, R.A.J. Neggers, A.P. Siebesma, and B. Stevens, Large-eddy simulation of the diurnal cycle of shallow cumulus convection over land, Q. J. R. Meteorol. Soc., 128, 582, pp. 1075-1093 (2002).

Kantz, H., Th. Schreiber, Nonlinear time series analysis, Cambridge Univ. Press, 0-521-55144-7 (1999).

Kugiumtzis, D., Test your surrogate data before you test for nonlinearity, Phys. Rev. E, 60, 3, pp. 2808-2816 (1999).

Lewis, G.M., Ph.H. Austin, An iterative method for generating scaling log-normal simulations, in Proc. of the 11th conf. on Atmospheric Radiation, 3-7 June, Odgen UT, USA (2002).

Masters, F., and K.R. Gurley, Non-Gaussian simulation: cumulative distribution function map-based spectral correction, J Eng. Mech., pp. 1418.1428, doi:10.1061/(ASCE)0733-9399(2003)129:12(1418) (2003).

Pincus, R., C. Hannay, and K.F. Evans, The accuracy of determining three-dimensional radiative transfer effects in cumulus clouds using ground-based profiling instruments, J. Atmos. Sc., 62, pp. 2284-2293 (2005).

Popescu, R. G. Deodatis, J.H. Prevost, Simulation of homogeneous non-Gaussian stochastic vector fields, Prob. Engng. Mech., 13, pp. 1-13 (1998).

Scheirer, R., and A. Macke, On the accuracy of the independent column approximation in calculating the downward fluxes in the UVA, UVB and PAR spectral ranges, J. Geophys. Res., 106, pp. 14,301-14,312 (2001).

Schreiber, Th., Constrained randomization of time series data, Phys. Rev. Lett., 80, pp. 2105 (1998).

Schreiber, T., and A. Schmitz, Improved surrogate data for nonlinearity tests, Phys. Rev. Lett, 77, pp. 635-638 (1996).

Schreiber, T., and A. Schmitz, Surrogate time series, Physica D, 142, 3-4, pp. 346 382 (2000).

Small, M., “Applied Nonlinear Time Series Analysis: Applications in Physics, Physiology and Finance”, 981-256-117-X, World Scientific (2005).

Theiler, J., S. Eubank, A. Longtin, B. Galdrikian, and J. D. Farmer, Testing for nonlinearity in time series: the method of surrogate data, Physica D, 58, pp. 77-94 (1992).

Theiler, J., and D. Prichard, Constrained-realization Monte-Carlo method for hypothesis testing, Physica D, 94, 4, pp. 221-235 (1996).

Venema, V., S. Meyer, S. Gimeno García, A. Kniffka, C. Simmer, S. Crewell, U. Löhnert, Th. Trautmann, and A. Macke, Surrogate cloud fields generated with the Iterative Amplitude Adapted Fourier Transform algorithm, Tellus 58A, no 1, pp. 104-120 (2006).

Vidal R.V.V. (ed.), “Applied simulated annealing”, ISBN: 3-540-56229-X, Springer, Berlin, Germany (1993).

Table 1. The final accuracy of the IAAFT and the SIAAFT algorithm for various test functions. N is the number of values of the time series. Δ(SIAAFT-10^2) or Δ(SIAAFT-10^4) denotes the accuracy of the SIAAFT algorithm, which stops if it does not find a better surrogate within 100 or 10.000 iterations, respectively. The values marked IAAFT-10^4 in this table are calculated with the SIAAFT algorithm, but all values are adjusted as in the IAAFT algorithm. In square brackets is the number of fully converged surrogates. For the difference between deterministic, partially stochastic and fully stochastic see section 4.

Test function	N	Δ(IAAFT-10^4)	Δ(SIAAFT-10^4)	Δ(SIAAFT-10^2)	Δ(SIAAFT-10^4)	Δ(SIAAFT-10^4)
			Deterministic	Part. stochastic	Part. stochastic	Fully stochastic
Fractal time series	8192	1.5 10^{-5}	4.3 10^{-6}	7.8 10^{-6}	2.3 10^{-6}	1.4 10^{-6}
Fractal time series	1024	2.7 10^{-4}	1.7 10^{-4}	1.3 10^{-4}	5.1 10^{-5}	5.9 10^{-5}
Discrete Gaussian	1024	3.1 10^{-3}	1.5 10^{-3}	1.6 10^{-3}	1.3 10^{-3}	1.3 10^{-3}
Positive Gaussian	1024	5.2 10^{-4}	1.7 10^{-4}	3.0 10^{-4}	1.4 10^{-4}	1.5 10^{-4}
Cumulus clouds		5.8 10^{-4}			1.2 10^{-4} [39]	
Daily rain [1]	32768	3.9 10^{-4}	1.4 10^{-4}	2.6 10^{-4}	1.4 10^{-4}	1.4 10^{-4}
Discharge [2]	32768	1.0 10^{-5}	3.1 10^{-6}	4.2 10^{-6}	3.4 10^{-6}	3.0 10^{-6}
EEG epileptic zone	256	2.1 10^{-3}	1.1 10^{-3}	1.1 10^{-3}	7.3 10^{-4}	7.3 10^{-4}
EEG seizure	4096	7.7 10^{-5}	3.5 10^{-5}	3.8 10^{-5}	2.9 10^{-5}	2.8 10^{-5}
Henon map	1024	1.5 10^{-4}	6.0 10^{-5}	6.4 10^{-5}	4.2 10^{-5}	3.6 10^{-5}
step-function	1024	4.2 10^{-3} [6]	$< 10^{-10}$ [25]	$< 10^{-10}$ [25]	$< 10^{-107}$ [25]	$< 10^{-10}$ [25]
random binary	1024	1.0 10^{-2}	5.1 10^{-3} [3]	7.0 10^{-3}	5.9 10^{-3} [4]	5.1 10^{-3} [7]
Random sine	1024	1.6 10^{-3}	9.9 10^{-4}	1.5 10^{-3}	1.3 10^{-3}	1.1 10^{-3}

[1] Based on 40 surrogates, 8 repetitions from 5 stations.

[2] Based on 24 surrogates, 8 repetitions from 3 stations.

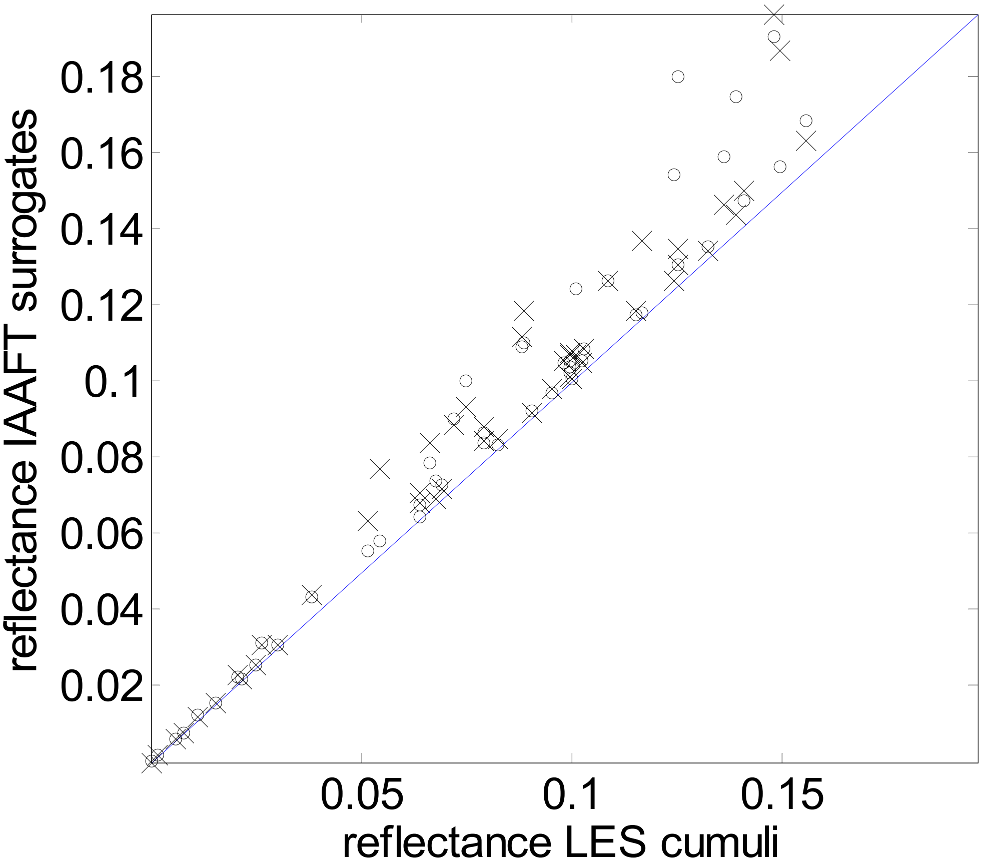

Figure 1. The reflectance of sparse cumulus clouds compared with the reflectance (R) of their IAAFT surrogates. The surrogate cumulus clouds have a bias toward a too high reflectance. The accuracy was improved by repeating the calculation and selecting the best converged surrogate. For every cumulus cloud, we have generated two surrogate clouds. The most accurate cloud (average Δ=3.4 10^{-3}; see Eq. 4) is denoted by a cross, the least accurate (average Δ=4.7 10^{-3}) one by a circle. Even in this optimised set of clouds, the accurate surrogates have a little less bias (ΔR=8.1 10^{-3}), than the less accurate ones (ΔR=8.9 10^{-3}).

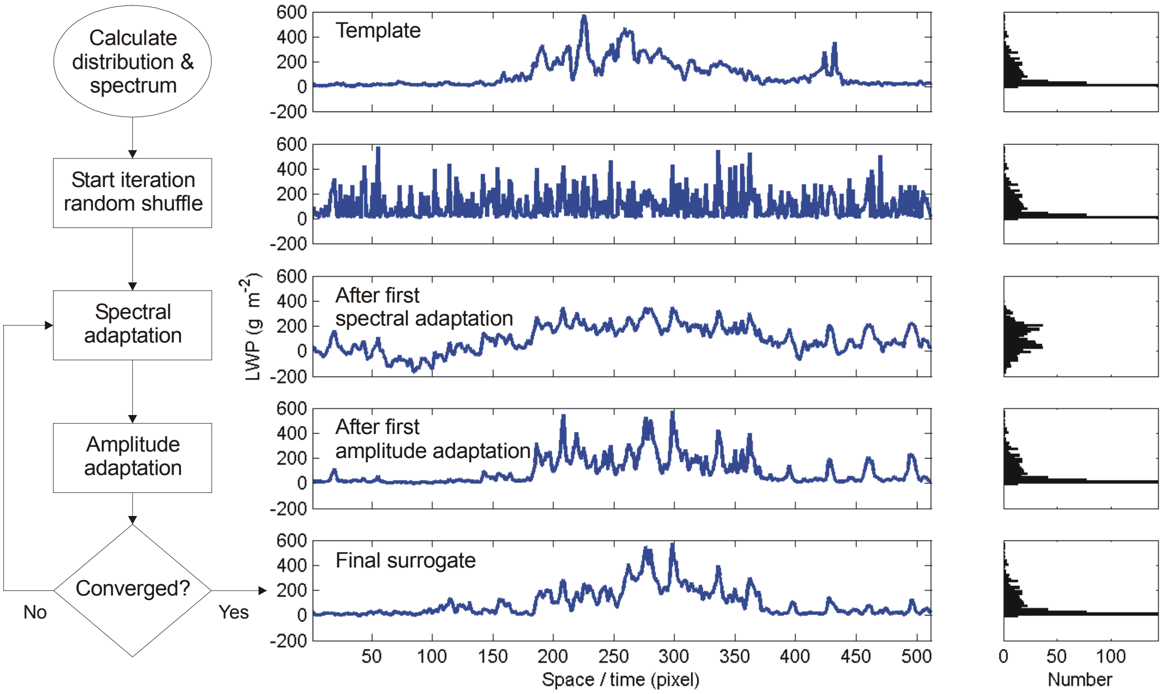

Figure 2. The Iterative Amplitude Adjusted Fourier Transform algorithm illustrated with a LWP measurement. The left column is the flow diagram of the algorithm, the middle column shows example LWP time series, and the right column are the histograms of the LWP time series. For the explanation of the algorithm, see Section 2.

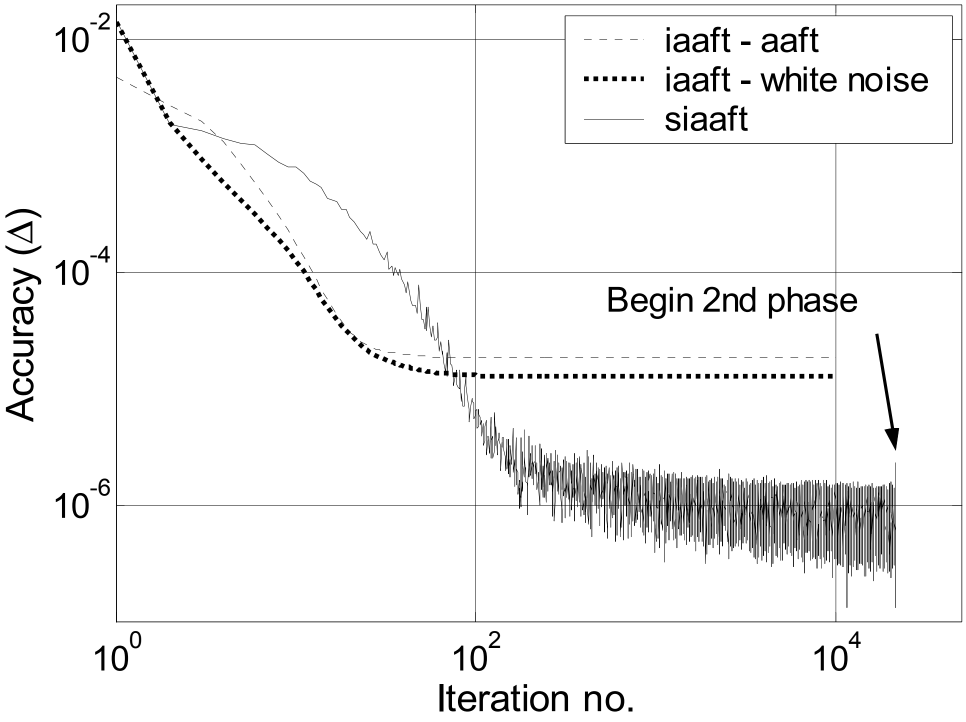

Figure 3. The convergence of the algorithms as a function of the number of iterations. As test statistics we used a fractal time series (N = 8192) with a power law power spectrum with an exponent of -5/3 and an exponential amplitude distribution. The three line types represent the IAAFT algorithm starting with white noise (thick dashes), the IAAFT algorithm starting with an AAFT surrogate (thin dashes) and the SIAAFT algorithm (noisy full line).

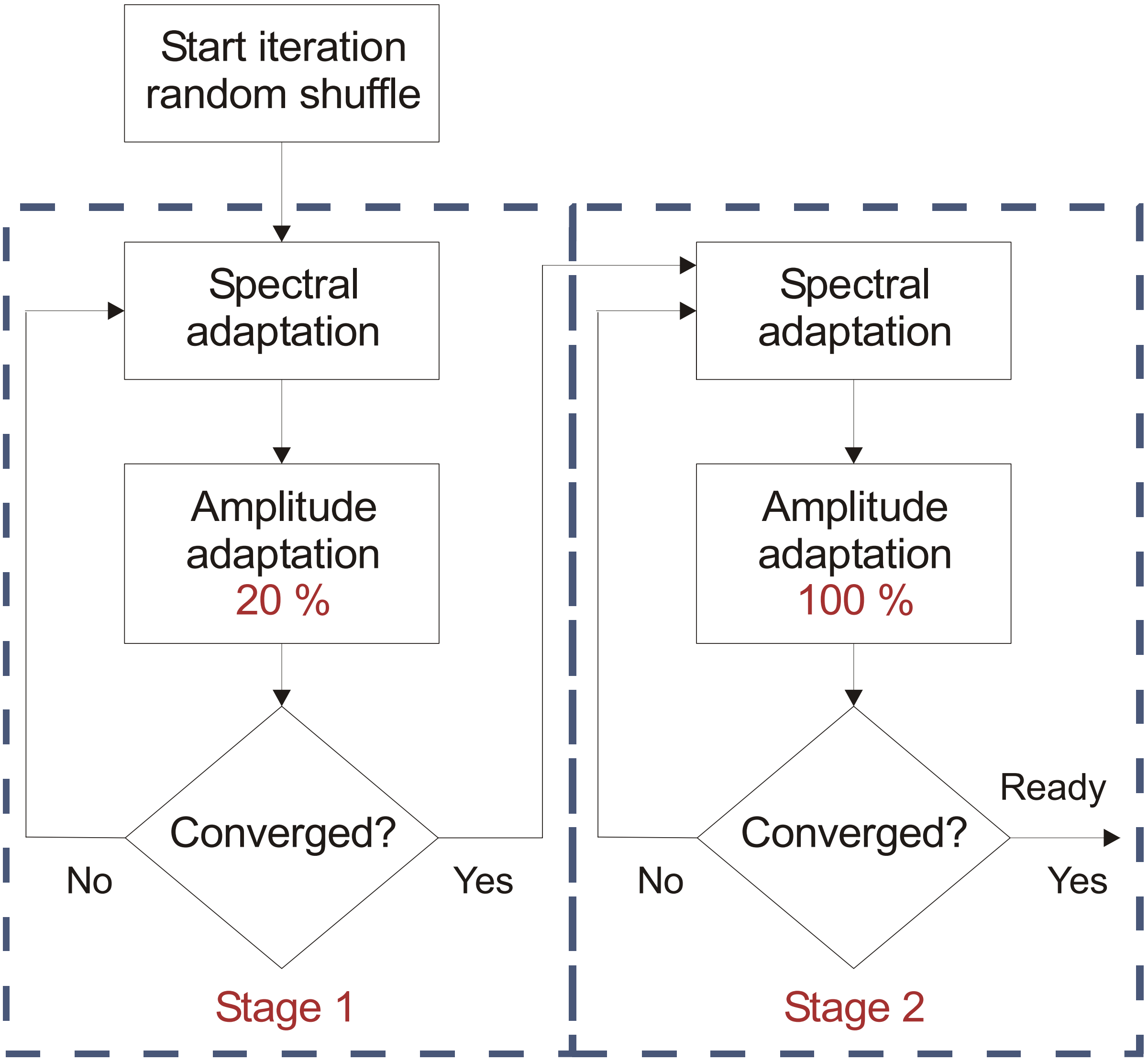

Figure 4. The flow diagram of the Stochastic Iterative Amplitude Adjusted Fourier Transform algorithm. The two stages are similar to the original IAAFT algorithm and differ in their treatment of the amplitude adjustment.

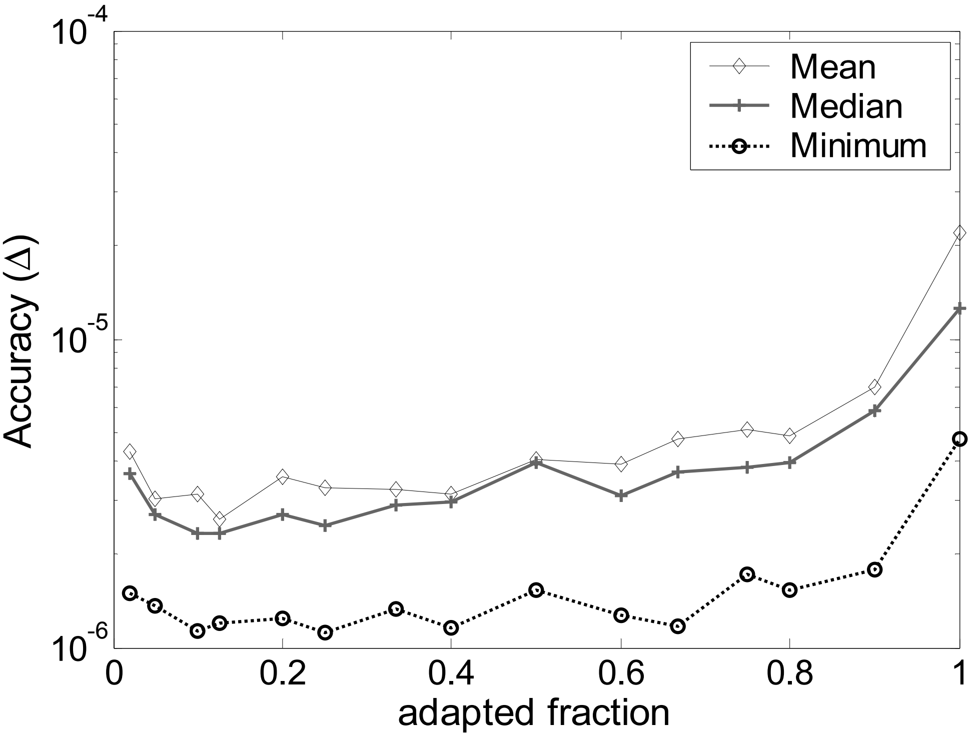

Figure 5. The accuracy of the SIAAFT algorithm as a function of the fraction of values that is adjusted in the amplitude adjustment. This was calculated with the test function described in Figure 3. The SIAAFT algorithm with an adjusted fraction of one is similar to the IAAFT algorithm. The SIAAFT algorithm used an iteration threshold of 10^3. An ensemble of one hundred surrogates was utilised to calculate the statistics.

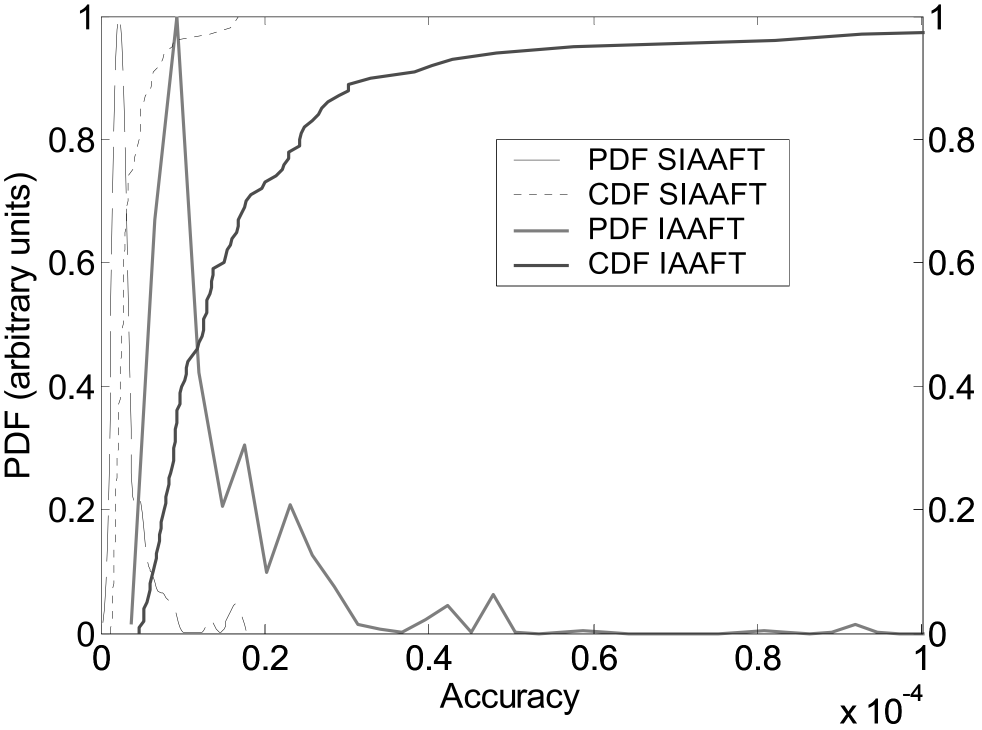

Figure 6. The Probability Density Functions (PDF) and the Cumulative Distribution Functions (CDF) of the accuracy (Δ) of the IAAFT and the SIAAFT algorithm. The SIAAFT algorithm used an iteration threshold of 1000 and the test function described in Figure 3. An ensemble of one hundred surrogates was generated to calculate these statistics. Three outliers of the IAAFT algorithm, were not plotted, to get a clearer plot.