



AppliedMath

Article

Financial Time Series Modelling Using Fractal Interpolation Functions

Polychronis Manousopoulos, Vasileios Drakopoulos and Efstathios Polyzos



<https://doi.org/10.3390/appliedmath3030027>

Article

Financial Time Series Modelling Using Fractal Interpolation Functions

Polychronis Manousopoulos ^{1,*}, Vasileios Drakopoulos ² and Efstathios Polyzos ³¹ Bank of Greece, 21 Eleftherios Venizelos Avenue, 10250 Athens, Greece² Department of Computer Science and Biomedical Informatics, University of Thessaly, 35131 Lamia, Greece; vdrakop@uth.gr³ College of Business, Zayed University, Abu Dhabi P.O. Box 144534, United Arab Emirates; efstathios.polyzos@zu.ac.ae

* Correspondence: pmanousopoulos@bankofgreece.gr

Abstract: Time series of financial data are both frequent and important in everyday practice. Numerous applications are based, for example, on time series of asset prices or market indices. In this article, the application of fractal interpolation functions in modelling financial time series is examined. Our motivation stems from the fact that financial time series often present fluctuations or abrupt changes which the fractal interpolants can inherently model. The results indicate that the use of fractal interpolation in financial applications is promising.

Keywords: financial time series; time series modelling; fractal interpolation; iterated function system



Citation: Manousopoulos, P.; Drakopoulos, V.; Polyzos, E. Financial Time Series Modelling Using Fractal Interpolation Functions. *AppliedMath* **2023**, *3*, 510–524. <https://doi.org/10.3390/appliedmath3030027>

Academic Editor: Alexander Ayriyan

Received: 12 May 2023

Revised: 23 June 2023

Accepted: 27 June 2023

Published: 29 June 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Time series of financial data such as asset prices or market indices occur frequently in everyday practice. The modelling, analysis and prediction of financial time series is an important tool in decision making for institutions or investors; see, for example [1], and [2]. It is not uncommon that financial time series are not smooth but exhibit significant fluctuations, because of reflecting a financial crisis or market turbulence. This provides an adequate motive of examining methods that inherently allow fluctuations when modelling such time series.

Fractal interpolation, as defined in [3], provides a framework for interpolating irregular, non-smooth sets of data, which possibly exhibit self-similarity and details at different scales. While traditional interpolation techniques are based on smooth functions such as polynomials and generate smooth interpolants, fractal interpolation is founded on the theory of iterated function systems (IFSs, see [4]) and is able to model irregular sets of data such as projections of physical objects, e.g., coastlines, or experimental data of non-integral dimension. See [5,6], for example, for such successful applications of fractal interpolation.

Financial time series are typically modelled using autoregressive integrated moving average (ARIMA) or generalised autoregressive conditional heteroskedasticity (GARCH) models; see e.g., [7–12]. These models take into account the autoregressive properties of such data and are thus commonly used in the existing literature to model returns/growth rates (ARIMA) or volatility (GARCH). In the last few years, there is an emerging trend using machine learning or deep learning approaches, such as Support Vector Machines (SVM), Long Short-Term Memory (LSTM) or other neural networks ([13–16]). Such approaches can provide greater predicting accuracy but are often prone to overfitting, thus resulting in data-specific findings ([17]). Furthermore, machine and deep learning approaches are also criticized for lack of interpretability ([18]), as these models are only able to supply numerical forecasts without providing interpretable results (such as coefficients or standard errors) that would improve our theoretical understanding of the underlying mechanism at

work; a new trend in Artificial Intelligence, termed eXplainable AI (XAI), aims to address this issue ([19,20]).

In this article, we examine the application of fractal interpolation to the modelling of financial time series. As previously mentioned, fractal interpolation functions are a suitable candidate for this application. The use of fractal techniques in the analysis of financial data is seen occasionally in the literature, see e.g., [21–30]. However, in most cases IFS-based fractal interpolation is not used; in the previous references the sole exception is [25], where a simple example of comparing self-affine fractal interpolation to linear interpolation of the values of a company's assets is presented. Here we perform in-sample and out-of-sample tests on a variety of sets of data (bitcoin prices, S&P 500, U.S. GDP) which possess different characteristics, such as time span and smoothness. The results indicate that fractal interpolation functions are indeed successful in this area, even when a sparse selection of interpolation points is made.

The article is structured as follows. Section 2 presents the elementary concept of both iterated functions system and the fractal interpolation function. Section 3 presents the results of modelling financial time series with fractal interpolation functions. Finally, Section 4 summarises our conclusions and indicates areas of future work.

2. Fractal Interpolation

Fractal interpolation, as defined in [3], is based on the theory of *iterated function systems*. Here we present the basic concepts; for more details see [4].

2.1. Iterated Function Systems

Let (X, ρ) be a complete metric space, for instance $(\mathbb{R}^n, \|\cdot\|)$. A function $f : X \rightarrow X$ is called a *Lipschitz function*, if there exists $k \in \mathbb{R}$ such that $\rho(f(x), f(y)) \leq k\rho(x, y)$, for all $x, y \in X$; by definition $k \geq 0$. If $k < 1$, then f is called a *contraction* with *contractivity factor* k . Let $\mathcal{H}(X)$ denote the set of nonempty, compact subsets of X . The metric space $(\mathcal{H}(X), h)$, where h is an appropriate metric, e.g., the Hausdorff metric, is often called the “space of fractals”. Note that not every member of $\mathcal{H}(X)$ is necessarily a fractal.

An *iterated function system* (or *IFS* for short) consists of a complete metric space (X, ρ) together with a finite set of continuous mappings $w_n : X \rightarrow X, n = 1, 2, \dots, N$. An IFS is denoted as $\{X; w_n, n = 1, 2, \dots, N\}$. If all mappings w_n are contractions with contractivity factors $s_n, n = 1, 2, \dots, N$, then the IFS is called *hyperbolic* with contractivity factor $s = \max_{n=1,2,\dots,N} s_n$. Let $W : \mathcal{H}(X) \rightarrow \mathcal{H}(X)$ be the mapping defined as $W(B) = \bigcup_{n=1}^N w_n(B)$, where $B \in \mathcal{H}(X)$ and $w_n(B) = \{w_n(b), b \in B\}$. Each hyperbolic IFS has a unique *attractor* which is the set $\mathcal{A} \in \mathcal{H}(X)$, with $\mathcal{A} = W(\mathcal{A}) = \lim_{n \rightarrow \infty} W^n(B)$ for every $B \in \mathcal{H}(X)$, where W^n denotes the n -fold composition $W \circ W \circ \dots \circ W$. Therefore the attractor \mathcal{A} is the unique fixed point of W and every set $B \in \mathcal{H}(X)$ converges to it under successive applications of W . The term attractor stems from this property, which also provides the basis for the computational construction of an IFS attractor. Specifically, the *deterministic iteration algorithm* or the *random iteration algorithm* (see e.g., [4]) can be used for constructing an IFS attractor.

2.2. Self-Affine Fractal Interpolation Functions

Let Δ_1 be a partition of the real compact interval $I = [a, b]$, i.e., $\Delta_1 = \{u_0, u_1, \dots, u_M\}$ where $a = u_0 < u_1 < \dots < u_M = b$. Let the set of data points be represented as $P = \{(u_m, v_m) \in I \times \mathbb{R}, m = 0, 1, \dots, M\}$. Let Δ_2 be another partition of $I = [a, b]$, i.e., $\Delta_2 = \{x_0, x_1, \dots, x_N\}$ where $a = x_0 < x_1 < \dots < x_N = b$, such that Δ_1 is a refinement of Δ_2 . Let the set of interpolation points be represented as $Q = \{(x_i, y_i) \in I \times \mathbb{R}, i = 0, 1, \dots, N \leq M\}$, while it is a subset of the data points, i.e., $Q \subseteq P$. The subintervals of Δ_2 $[x_i, x_{i+1}]$, $i = 0, 1, \dots, N - 1$, are called interpolation intervals; the abscissas x_i of the interpolation points may be equidistant or not. The set of data points within the n th interpolation interval $I_n = [x_{n-1}, x_n], n = 1, 2, \dots, N$, is denoted as $P_n = \{(u_m, v_m) : x_{n-1} \leq u_m \leq x_n\}$; obviously, it is $P = \bigcup_{n=1}^N P_n$.

An *affine transformation* is defined as the composition of a linear transformation and a translation. Let $\{\mathbb{R}^2; w_n, n = 1, 2, \dots, N\}$ be an iterated function system with affine transformations

$$w_n \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_n & 0 \\ c_n & s_n \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d_n \\ e_n \end{bmatrix}$$

satisfying the constraints

$$w_n \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} x_{n-1} \\ y_{n-1} \end{bmatrix} \text{ and } w_n \begin{bmatrix} x_N \\ y_N \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \end{bmatrix}$$

for every $n = 1, 2, \dots, N$. In other words, I is mapped by each affine transformation w_n to its corresponding interpolation interval. Solving the above equations leads to

$$a_n = \frac{x_n - x_{n-1}}{x_N - x_0}, d_n = \frac{x_N x_{n-1} - x_0 x_n}{x_N - x_0}$$

$$c_n = \frac{y_n - y_{n-1}}{x_N - x_0} - s_n \frac{y_N - y_0}{x_N - x_0}, e_n = \frac{x_N y_{n-1} - x_0 y_n}{x_N - x_0} - s_n \frac{x_N y_0 - x_0 y_N}{x_N - x_0}$$

i.e., the real parameters a_n, c_n, d_n, e_n are uniquely determined by the interpolation points, while the real parameters s_n are *free parameters* of the transformations such that $|s_n| < 1, n = 1, 2, \dots, N$. This constraint is necessary to guarantee that the IFS is hyperbolic with respect to an appropriate metric. The transformations w_n are *shear* transformations, i.e., they map line segments parallel to the y -axis to line segments also parallel to the y -axis contracted by a factor $|s_n|$. For this reason, the parameters s_n are called *vertical scaling factors* or *contractivity factors* of the transformations w_n .

The attractor of the afore-mentioned IFS, i.e., the unique set $G = \bigcup_{n=1}^N w_n(G)$, is the graph of a continuous function $f : [x_0, x_N] \rightarrow \mathbb{R}$ that passes through all interpolation points $(x_i, y_i), i = 0, 1, \dots, N$. This function is called *fractal interpolation function*, or *FIF* for short, corresponding to these points. It is a *self-affine* function since each affine transformation w_n maps the entire graph of the function to its section within the corresponding interpolation interval.

An example of a fractal interpolation function is presented in Figure 1, where a set of ten interpolation points and vertical scaling factors $s_n = 0.25, n = 1, 2, \dots, 10$ are used. Note that from a simple set of few points a complicated interpolant is generated.

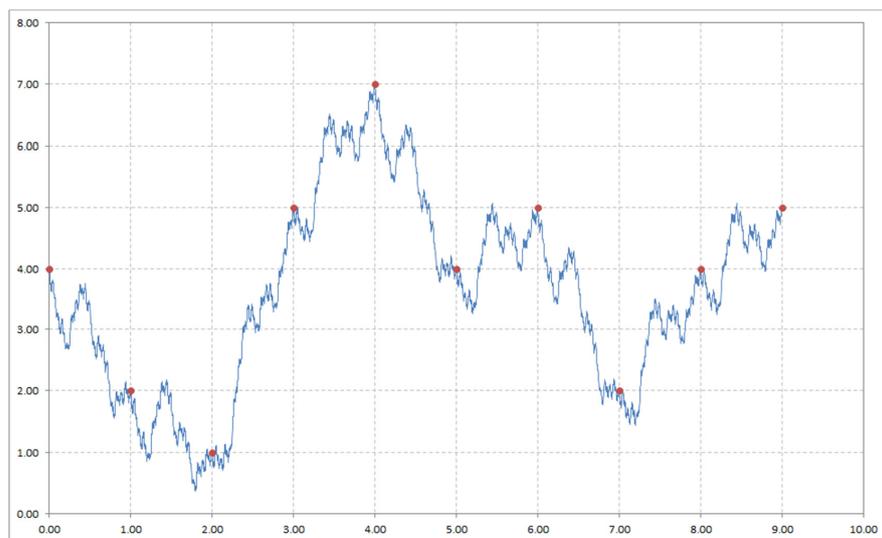


Figure 1. A fractal interpolation function constructed from the set of interpolation points $Q = \{(0, 4), (1, 2), (2, 1), (3, 5), (4, 7), (5, 4), (6, 5), (7, 2), (8, 4), (9, 5)\}$ (marked red) using the vertical scaling factors $s_n = 0.25, n = 1, 2, \dots, 10$.

The remaining data points $P \setminus Q$ are only approximated by the fractal interpolation function, which does not necessarily pass through them. To optimise the fit of the data points for a given set of interpolation points, the vertical scaling factors, the only free parameters, must be appropriately determined. Various methods have been proposed in the literature for calculating them. Usually, the vertical scaling factors are calculated by minimising an error measure, such as the squared error between the ordinates of the original and the reconstructed points $\sum_{m=0}^M (v_m - G[u_m])^2$, where $G[u_m]$ is the attractor ordinate at abscissa u_m , or the Hausdorff metric $h(P, G)$. For example, in [31] an analytic and a geometric method are proposed for minimising this squared error. In [32,33] the use of bounding volumes of data points subsets is proposed, to optimise the fit between original and transformed bounding volumes instead of individual points. Other methods use different approaches instead of minimising an error measure; in [34] the target is to preserve the fractal dimension of the data points; in [35] the target is to detect self-affinity and the implied vertical scaling factors in the continuous wavelet transform of the data.

An additional example of a fractal interpolation function is depicted in Figure 2. A set of 37 data points is interpolated by a fractal interpolation function using every 3rd point as interpolation points, i.e., 13 points in total; the vertical scaling factors are calculated by the geometric algorithm of [31]. Despite of the use of approximately only 1/3 of the data points and the simple, equidistant selection of the interpolation points, the remaining data points are approximated rather well.

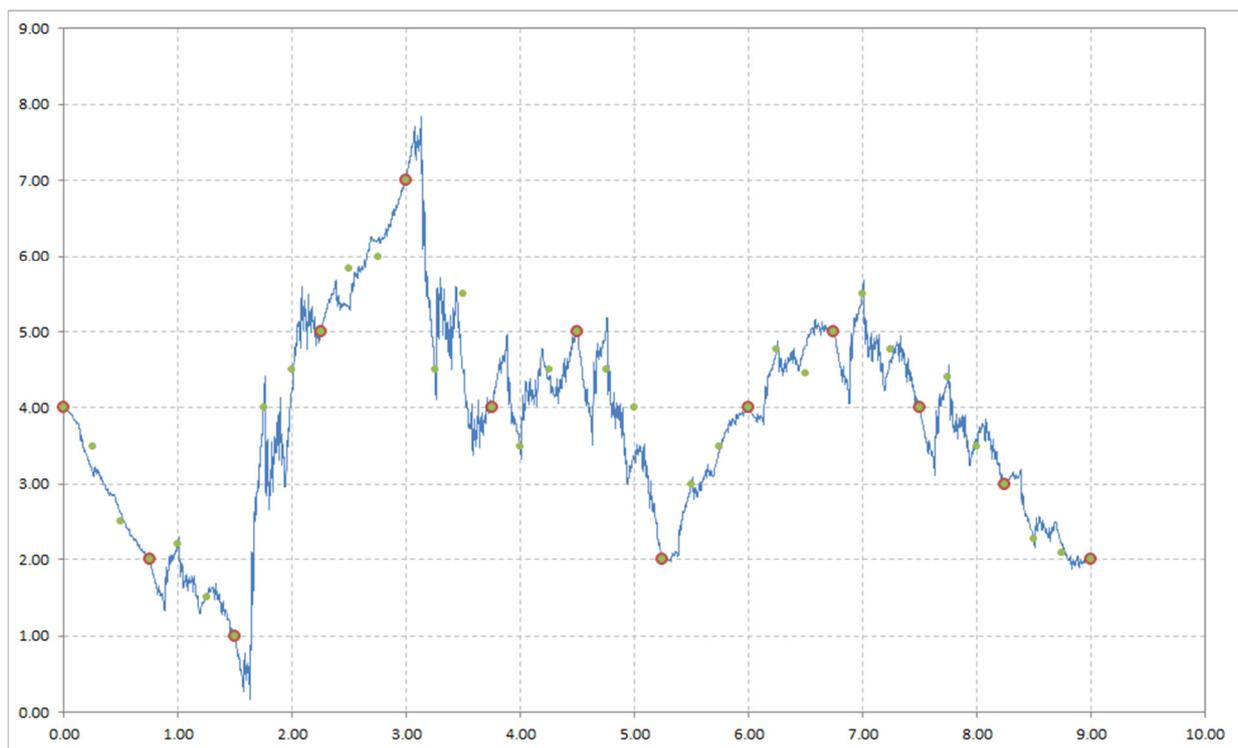


Figure 2. A fractal interpolation function constructed from a set of 13 interpolation points (red) selected from a set of 37 data points (green). The set of interpolation points is $Q = \{(0,4), (0.75,2), (1.5,1), (2.25,5), (3,7), (3.75,4), (4.5,5), (5.25,2), (6,4), (6.75,5), (7.5,4), (8.25,3), (9,2)\}$, while the set of data points is $P = \{(0,4), (0.25,3.5), (0.5,2.5), (0.75,2), (1,2.20), (1.25,1.5), (1.5,1), (1.75,4), (2,4.5), (2.25,5), (2.5,5.82), (2.75,5.99), (3,7), (3.25,4.5), (3.5,5.5), (3.75,4), (4,3.5), (4.25,4.5), (4.5,5), (4.75,4.5), (5,4), (5.25,2), (5.5,3), (5.75,3.5), (6,4), (6.25,4.77), (6.5,4.46), (6.75,5), (7,5.5), (7.25,4.77), (7.5,4), (7.75,4.41), (8,3.5), (8.25,3), (8.5,2.27), (8.75,2.09), (9,2)\}$. The vertical scaling factors have been calculated by the geometric algorithm of [31].

It is worth noting that although one could use all data points as interpolation points thus guaranteeing that the resulting fractal interpolation function passes through all of

them, this is not always desirable in practice. A proper selection of interpolation points, along with optimal calculation of the vertical scaling factors, will result in satisfactory goodness of fit and a considerable compression ratio without overfitting. Note that the interpolation points should not necessarily be equidistant, but can be also selected to minimise an error measure, see e.g., [31].

2.3. Recurrent Fractal Interpolation Functions

The recurrent fractal interpolation functions are a generalization of the self-affine fractal interpolation functions of the previous section. This is achieved by allowing piecewise instead of total self-affinity. Initially, the partitions Δ_1 and Δ_2 of the interval I , the set P of data points and its subsets P_n , the set Q of interpolation points and the interpolation intervals I_n , are all defined as in the previous section. Additionally, each interpolation interval is associated with a pair of data points called *address points*. Specifically, each interpolation interval $I_n = [x_{n-1}, x_n], n = 1, 2, \dots, N$ is associated with two distinct data points $(x'_{n,1}, y'_{n,1}), (x'_{n,2}, y'_{n,2}) \in P$, i.e., $(x'_{n,k}, y'_{n,k}) = (u_{m_k}, v_{m_k})$ for $k = 1, 2$ and some $m_k \in \{0, 1, \dots, M\}$. Each pair of address points defines the *address interval* $[x'_{n,1}, x'_{n,2}]$, where $x'_{n,1} < x'_{n,2}$ by definition. The address points are not necessarily unique among interpolation intervals, i.e., a data point may be used in more than one address interval. Moreover, every address interval should have strictly greater length than its corresponding interpolation interval, i.e., $(x'_{n,2} - x'_{n,1}) > (x_n - x_{n-1})$, for all $n = 1, 2, \dots, N$. The set of data points within the n th address interval $I'_n = [x'_{n,1}, x'_{n,2}], n = 1, 2, \dots, N$, is denoted as $A_n = \{(u_m, v_m) : x'_{n,1} \leq u_m \leq x'_{n,2}\}$.

Let $\{\mathbb{R}^2; w_n, n = 1, 2, \dots, N\}$ be an iterated function system with affine transformations

$$w_n = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_n & 0 \\ c_n & s_n \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d_n \\ e_n \end{bmatrix}$$

satisfying the constraints

$$w_n \begin{bmatrix} x'_{n,1} \\ y'_{n,1} \end{bmatrix} = \begin{bmatrix} x_{n-1} \\ y_{n-1} \end{bmatrix} \text{ and } w_n \begin{bmatrix} x'_{n,2} \\ y'_{n,2} \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \end{bmatrix}$$

for every $n = 1, 2, \dots, N$. In other words, each address interval is mapped to its corresponding interpolation interval. Solving these constraint equations leads to

$$a_n = \frac{x_n - x_{n-1}}{x'_{n,2} - x'_{n,1}}, d_n = \frac{x'_{n,2}x_{n-1} - x'_{n,1}x_n}{x'_{n,2} - x'_{n,1}}$$

$$c_n = \frac{y_n - y_{n-1}}{x'_{n,2} - x'_{n,1}} - s_n \frac{y'_{n,2} - y'_{n,1}}{x'_{n,2} - x'_{n,1}}, e_n = \frac{x'_{n,2}y_{n-1} - x'_{n,1}y_n}{x'_{n,2} - x'_{n,1}} - s_n \frac{x'_{n,2}y'_{n,1} - x'_{n,1}y'_{n,2}}{x'_{n,2} - x'_{n,1}}$$

i.e., the real parameters a_n, c_n, d_n, e_n are uniquely determined by the interpolation points and the address points, while the real parameters s_n are free parameters of the transformations such that $|s_n| < 1, n = 1, 2, \dots, N$. This constraint is necessary to guarantee that the resulting IFS is hyperbolic with respect to an appropriate metric. As previously, the transformations w_n are shear transformations and the parameters s_n are called vertical scaling factors or contractivity factors of the transformations w_n .

The attractor of the aforementioned IFS, i.e., the unique set $G = \bigcup_{n=1}^N w_n(G)$, is the graph of a continuous function $f : [x_0, x_N] \rightarrow \mathbb{R}$ that passes through all interpolation points $(x_i, y_i), i = 0, 1, \dots, N$. This function is called *recurrent fractal interpolation function (RFIF)* corresponding to these points. It is a *piecewise self-affine* function since each affine transformation w_n maps the part of the graph of the function within the corresponding address interval to its section within the corresponding interpolation interval. Note that

when all address intervals are equal to the interval I , then the recurrent fractal interpolation function reduces to a fractal interpolation function of the previous section.

An example of a recurrent fractal interpolation function is presented in Figure 3, where the set of interpolation points of Figure 1 is used; the address intervals are distinct and the vertical scaling factors are $s_n = 0.25, n = 1, \dots, 10$. Although the same vertical scaling factors as in Figure 1 have been used, the resulting interpolant is smoother. This is because each address intervals contains fewer interpolation points than the whole function, thus resulting in smaller variation within the respective interpolation interval to which it is mapped.

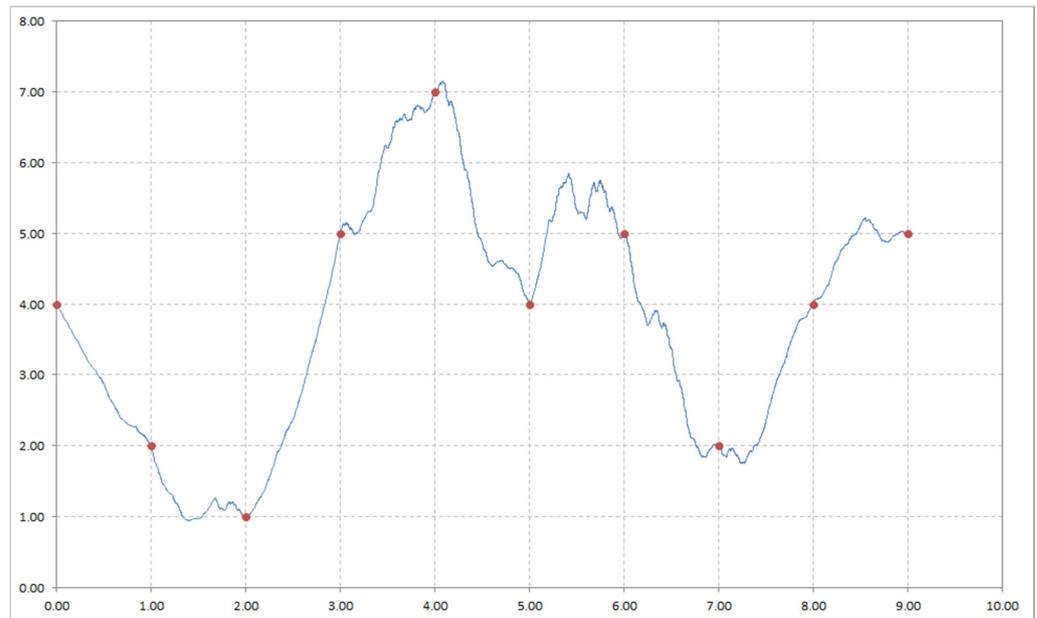


Figure 3. A recurrent fractal interpolation function constructed from the set of interpolation points Q (red) of Figure 1, using the address intervals $\{[Q^0, Q^2], [Q^1, Q^4], [Q^1, Q^3], [Q^4, Q^7], [Q^5, Q^9], [Q^2, Q^7], [Q^4, Q^8], [Q^6, Q^9], [Q^3, Q^5]\}$, where Q^i denotes the i th interpolation point. The vertical scaling factors are $s_n = 0.25, n = 1, 2, \dots, 10$.

The remaining data points $P \setminus Q$ are approximated by the recurrent fractal interpolation function, since it does not necessarily pass through them. In order to optimise the fit, the vertical scaling factors are calculated by methods similar to those described in the previous section for the fractal interpolation functions. These include methods that minimise an error measure, such as the analytic and geometric methods of [31], or the bounding volumes method of [32,33]. Other methods employ the use of fractal dimension ([34]) or wavelets ([35]). The remarks of previous section about the selection of the interpolation points apply here as well.

An additional example is depicted in Figure 4. The set of data points and interpolation points of Figure 2 is used, the address intervals are of variable length and the vertical scaling factors are calculated by the analytic algorithm of [31]. Note that the recurrent interpolant approximates even better the remaining data points. This is allowed by the property of partial, piecewise self-affinity, in contrast to the total self-affinity of the fractal interpolant of Figure 2.

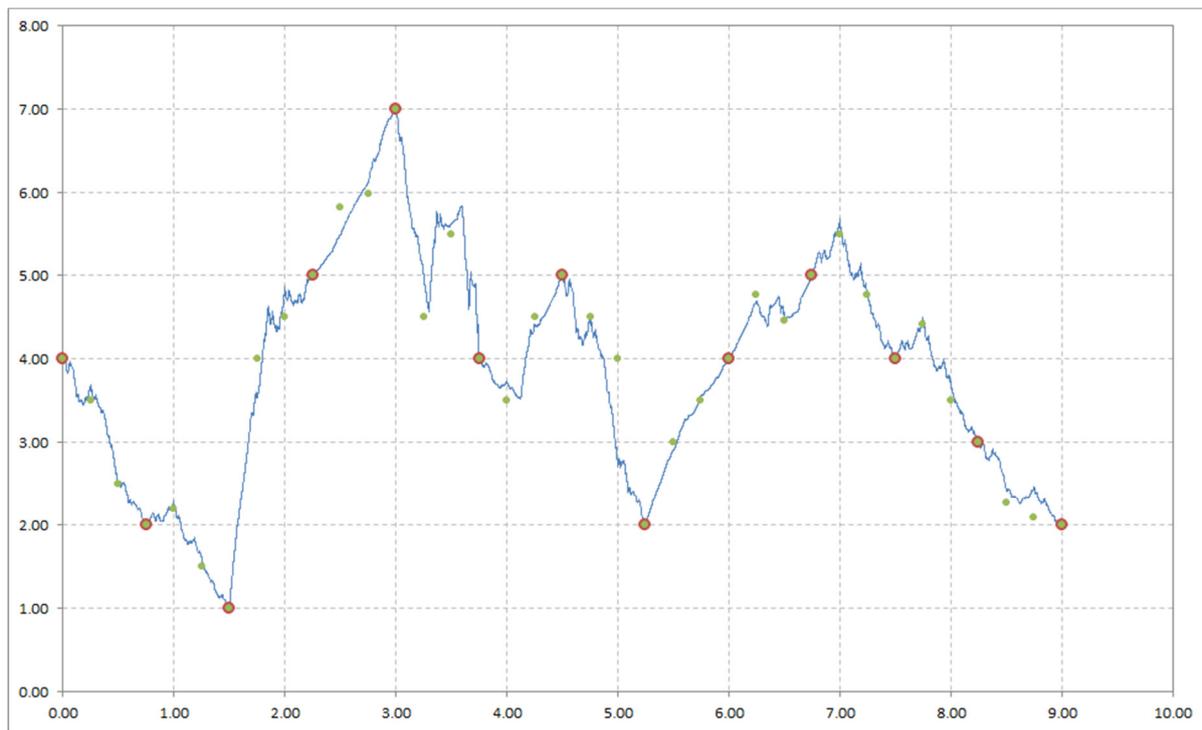


Figure 4. A recurrent fractal interpolation function constructed from the interpolation points (red) and data points (green) of Figure 2, using the address intervals $[p^0, p^6]$, $[p^{24}, p^{36}]$, $[p^{21}, p^{36}]$, $[p^{21}, p^{27}]$, $[p^0, p^{15}]$, $[p^0, p^{12}]$, $[p^0, p^6]$, $[p^{21}, p^{36}]$, $[p^9, p^{18}]$, $[p^{24}, p^{36}]$, $[p^{24}, p^{36}]$, $[p^0, p^6]$, where P^i denotes the i th data point. The vertical scaling factors have been calculated by the analytic algorithm of [31].

3. Financial Time Series Modelling

In this section, we present three cases of financial time series modelling. Each case is different with respect to the extent or the characteristics of the underlying data set. Our aim is to show the suitability of fractal interpolation in a diversity of applications. There are inherent difficulties in modelling and predicting financial data series of different forms. Cryptocurrency prices exhibit significant volatility [36], stock market indices are susceptible to external shocks [37], while GDP data is affected by economic cycles which can have indeterminate duration [38]. Thus, an adaptable methodology that would permit short-term forecasts or accurate interpolation to higher frequency data could be very useful to researchers.

3.1. Dataset 1—Bitcoin Prices

The first set of data consists of bitcoin historical prices. Bitcoin is a decentralized digital cryptocurrency that allows direct transactions between members of a peer-to-peer network without the need of a trusted third party [39]. It emerged in 2009 and has gained widespread popularity and use since then. The data set has a two-year time span. Specifically, it consists of weekly prices from 23 December 2018 to 16 December 2020 containing a total of 105 data points (data obtained from Bloomberg Professional (accessed on 18 April 2021)). The time series has been modelled by a recurrent fractal interpolation function as depicted in Figure 5. For constructing the interpolant, every 4th data point is used as interpolation point, resulting in a total of 27 interpolation points. The address intervals have a fixed span of 25 data points; for each interpolation interval I_n , the optimal address interval has been selected between possible candidates by minimising the Hausdorff distance between the original and mapped data points, i.e., $A_n = \arg \left\{ \min_{A \in A(P,l)} h(P_n, w_n(A)) \right\}, n = 1, 2, \dots, N$, where $A(p, l)$ denotes the subsets of P spanning l consecutive data points. The vertical

scaling factors are calculated using the analytic algorithm of [31]. Note that although only 1/4 of the data points are used, the fractal interpolant represents successfully the time series; the fluctuations of the underlying dataset are correctly captured. However, this would not be the case if a smooth interpolant, e.g., a cubic spline, was used for the same set of interpolation points.

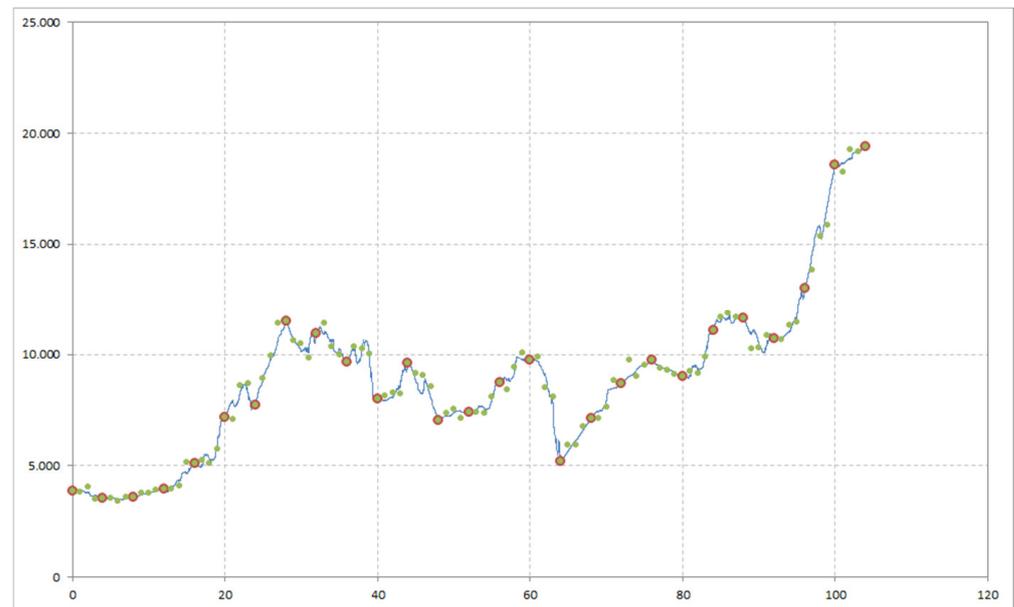


Figure 5. A time series of bitcoin prices (23 December 2018–16 December 2020) modelled by a recurrent fractal interpolation function; every 4th data point (green) is used as interpolation point (red), the address intervals have fixed span of 25 points and are optimally selected (see text), while the vertical scaling factors have been calculated by the analytic algorithm of [31]. The x-axis indicates the data points index.

3.2. Dataset 2—S&P 500

The second set of data consists of S&P 500 historical values. The S&P 500 is a stock market index; specifically, it is a capitalization weighted index of 500 large cap companies publicly traded in the U.S. (S&P 500, available online: https://en.wikipedia.org/wiki/S%26P_500 (accessed on 2 December 2021)). It is considered to be one of the best indicators of U.S. equities performance and it is widely followed by investors and institutions. The data set has a 15-year time span approximately. Specifically, it consists of daily values from 9 December 2005 to 1 March 2021 containing a total of 3830 data points (data obtained from Bloomberg Professional (accessed on 18 April 2021)). The time series has been modelled by a recurrent fractal interpolation function as depicted in Figure 6. For constructing the interpolant, every 10th data point is used as interpolation point, resulting in a total of 383 interpolation points. The address intervals have a fixed span of 25 data points; for each interpolation interval I_n , the optimal address interval is selected between possible candidates by minimising the Hausdorff distance between the original and mapped data points, as in the first dataset. The vertical scaling factors are calculated using the analytic algorithm of [31]. Note that although only 1/10 of the data points are used, the fractal interpolant is able to represent accurately the time series. This is more clearly shown in Figure 7, which depicts the part of Figure 6 between the 3630th and the 3820th data points. In Figure 7, we see that the fractal interpolant closely follows the data points despite the considerable sparsity of the interpolation points. This is explained by the existence of partial self-affinity in the underlying data set, which the recurrent fractal interpolant successfully utilizes by selecting the optimal address interval for each interpolation interval.

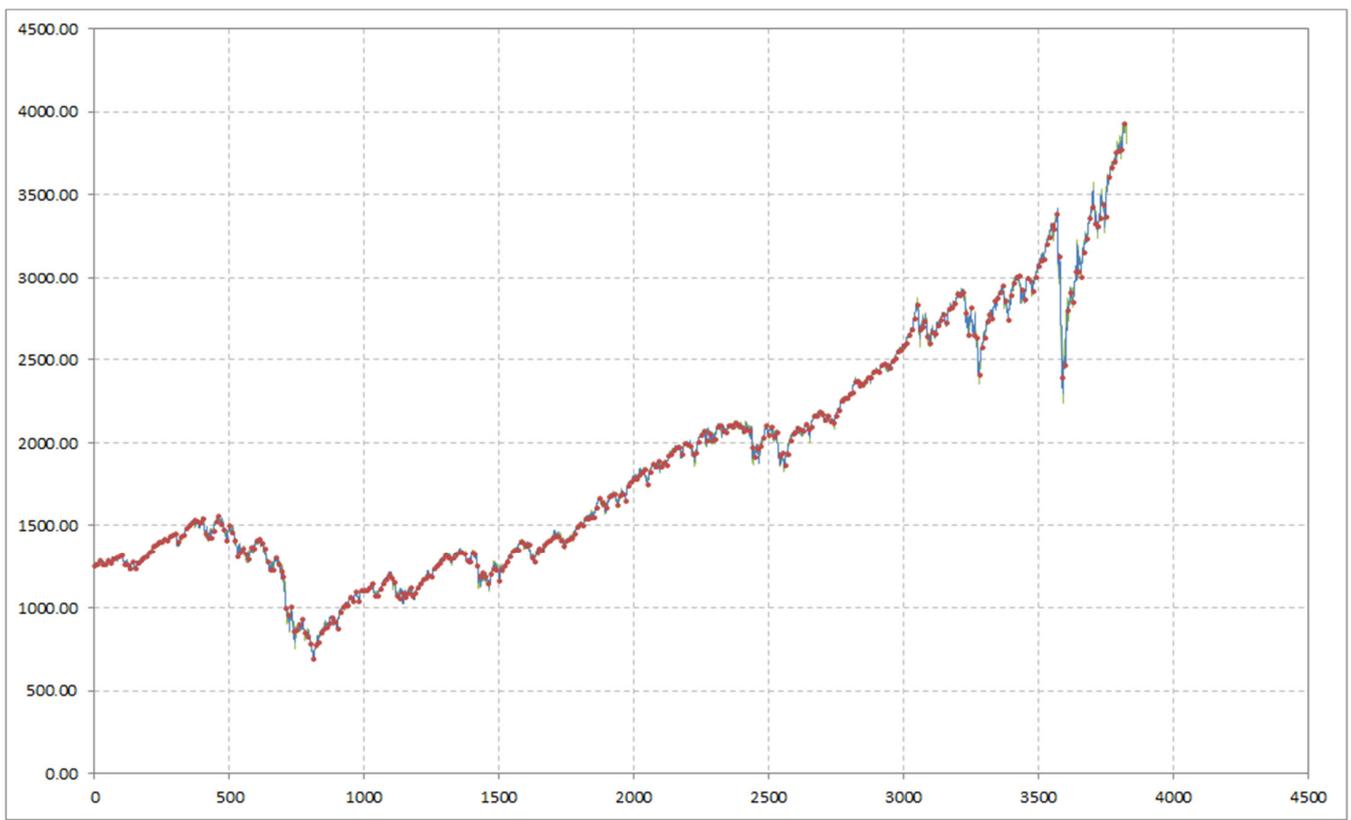


Figure 6. A time series of S&P 500 values (9 December 2005–March 2021) modelled by a recurrent fractal interpolation function; every 10th data point (green) is used as interpolation point (red), the address intervals have fixed span of 25 points and are optimally selected (see text), while the vertical scaling factors have been calculated by the analytic algorithm of [31]. The x-axis indicates the data points index.

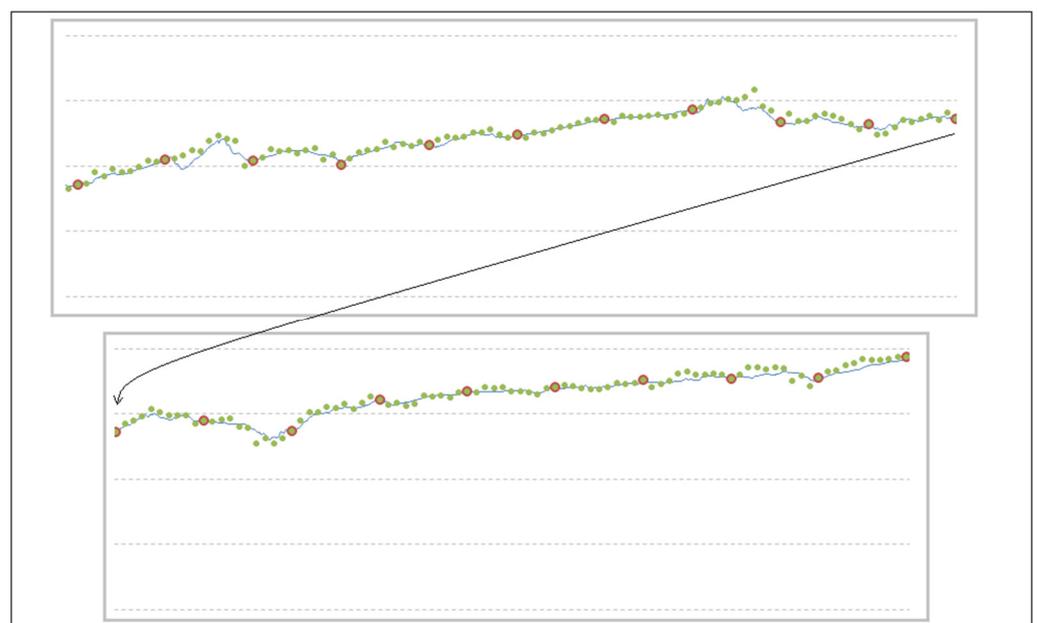


Figure 7. A zoom in Figure 6 between the 3630th and the 3820th data points (green) including the respective interpolation points (red). The figure is split into two parts (continuation as shown by the arrow).

3.3. Dataset 3—U.S.A. GDP

The third set of data consists of U.S.A. gross domestic product (GDP) historical values. The GDP is “an aggregate measure of production equal to the sum of the gross values added of all resident institutional units engaged in production” (OECD Glossary of Statistical Terms—Gross Domestic Product (GDP) definition, available online: <https://stats.oecd.org/glossary/detail.asp?ID=1163> (accessed on 10 December 2021)). The data set has a 74-year time span. Specifically, it consists of quarterly values from 1947-Q1 to 2020-Q4 containing a total of 296 data points (data obtained from Bloomberg Professional (accessed on 18 April 2021)). The time series has been modelled by a recurrent fractal interpolation function as depicted in Figure 8. For constructing the interpolant, every 5th data point is used as interpolation point, resulting in a total of 60 interpolation points. The address intervals have a fixed span of 20 data points; for each interpolation interval I_n , the optimal address interval has been selected between possible candidates by minimising the Hausdorff distance between the original and mapped data points, as in the previous sets of data. The vertical scaling factors are calculated using the analytic algorithm of [31]. Note that this dataset has a smoother structure than the previous ones presenting fewer fluctuations. In this case also, the fractal interpolant accurately models the time series with only 1/5 of the data points being used as interpolation points.

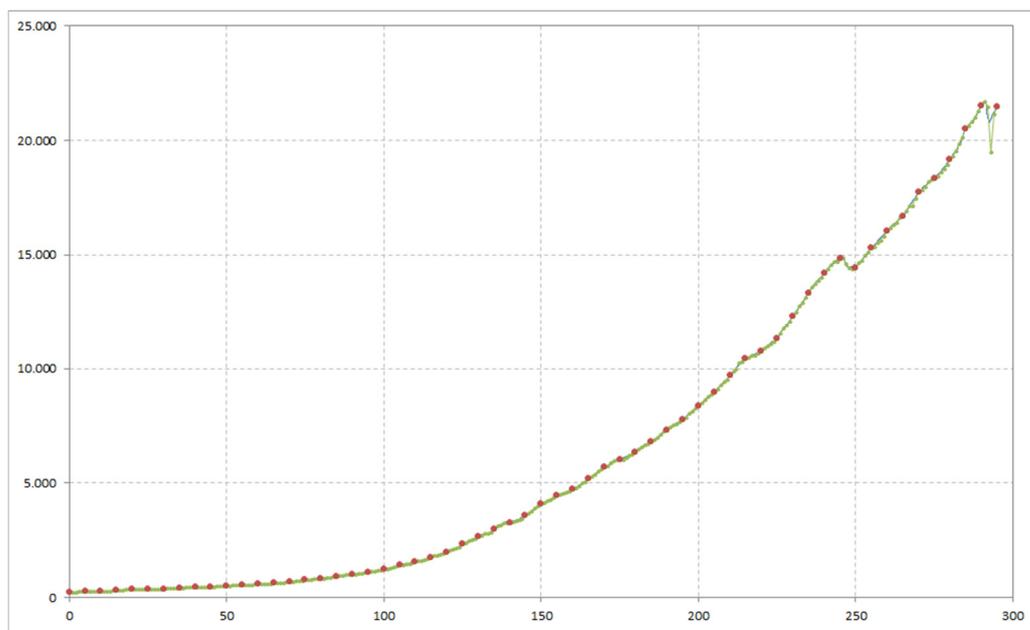


Figure 8. A time series of U.S. GDP values (1947-Q1–2020-Q4) modelled by a recurrent fractal interpolation function; every 5th data point (green) is used as interpolation point (red), the address intervals have fixed span of 20 points and are optimally selected (see text), while the vertical scaling factors have been calculated by the analytic algorithm of [31]. The x-axis indicates the data points index.

3.4. Comparison to Existing Methods

In order to further evaluate the proposed fractal interpolation method, we compare it to existing methods for financial time series modelling. Specifically, we compare it to the autoregressive integrated moving average (ARIMA) and generalized autoregressive conditional heteroskedasticity (GARCH) methods, which are typically the methods of choice for financial time series modelling.

Let Y_t be a real valued time-series. The ARIMA(p, d, q) model is defined by

$$y_t = \sum_{i=1}^p a_i y_{t-i} + \varepsilon_t + \sum_{j=1}^q \vartheta_j \varepsilon_{t-j}$$

where y_t is the d -order differenced values of Y_t , while a_i and θ_i are the model coefficients and ε_t are the error terms. The GARCH(p, q) model is defined by

$$\sigma_t^2 = a_0 + \sum_{i=1}^q a_i \varepsilon_{t-i}^2 + \varepsilon_t + \sum_{j=1}^p \beta_j \sigma_{t-j}^2$$

where σ_t^2 is the variance, a_i and β_i are the model coefficients and ε_t are the error terms.

The comparison of the methods is performed using the three datasets previously presented. For this test, the ARIMA(1, 1, 0) model is employed, i.e., a differenced first-order autoregressive model, along with the GARCH(1, 1) model. The order of the models was chosen using the Akaike information criterion (AIC). Note that the model order used in this test is commonly used in financial time-series. The recurrent fractal interpolation functions are constructed as described in Sections 3.1–3.3.

The results of the comparison are presented in Figures 9–11 and summarized in Tables 1–3. The results show that the proposed fractal interpolation method has clearly outperformed the ARIMA and GARCH methods. This indicates that fractal interpolation can provide an adequate methodology for financial time series modelling.

Table 1. Results of the three methods for the first dataset (Bitcoin prices).

Dataset 1—Bitcoin Prices			
	Mean Abs. Error	Mean Abs. % Error	RMSE
ARIMA	547.71	6.46%	740.16
GARCH	557.96	6.55%	762.67
RFIF	198.40	2.34%	289.18

Table 2. Results of the three methods for the second dataset (S&P 500).

Dataset 2—S&P 500			
	Mean Abs. Error	Mean Abs. % Error	RMSE
ARIMA	13.95	0.80%	22.57
GARCH	23.91	1.31%	32.12
RFIF	10.42	0.59%	17.65

Table 3. Results of the three methods for the third dataset (U.S.A. GDP).

Dataset 3—U.S.A. GDP			
	Mean Abs. Error	Mean Abs. % Error	RMSE
ARIMA	55.51	1.10%	158.42
GARCH	88.11	1.52%	171.20
RFIF	20.07	0.35%	92.76

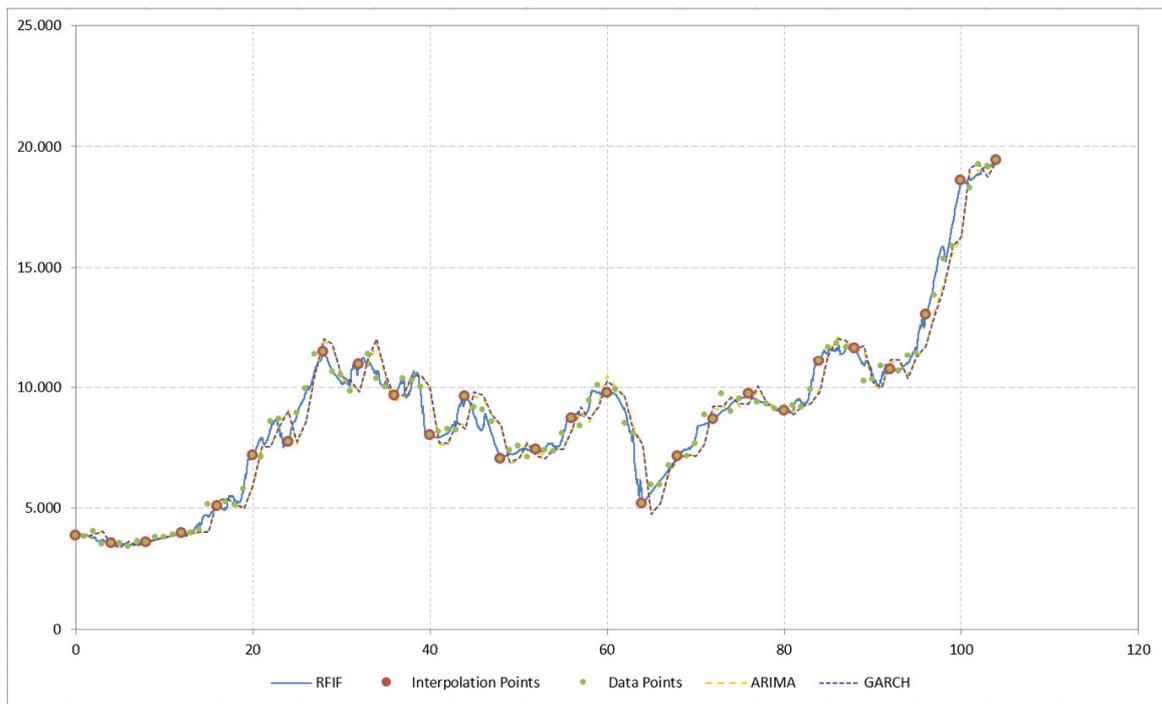


Figure 9. A time series of bitcoin prices (23 December 2018–16 December 2020) modelled by (i) a recurrent fractal interpolation function, (ii) the ARIMA(1, 1, 0) model, (iii) the GARCH(1,1) model.

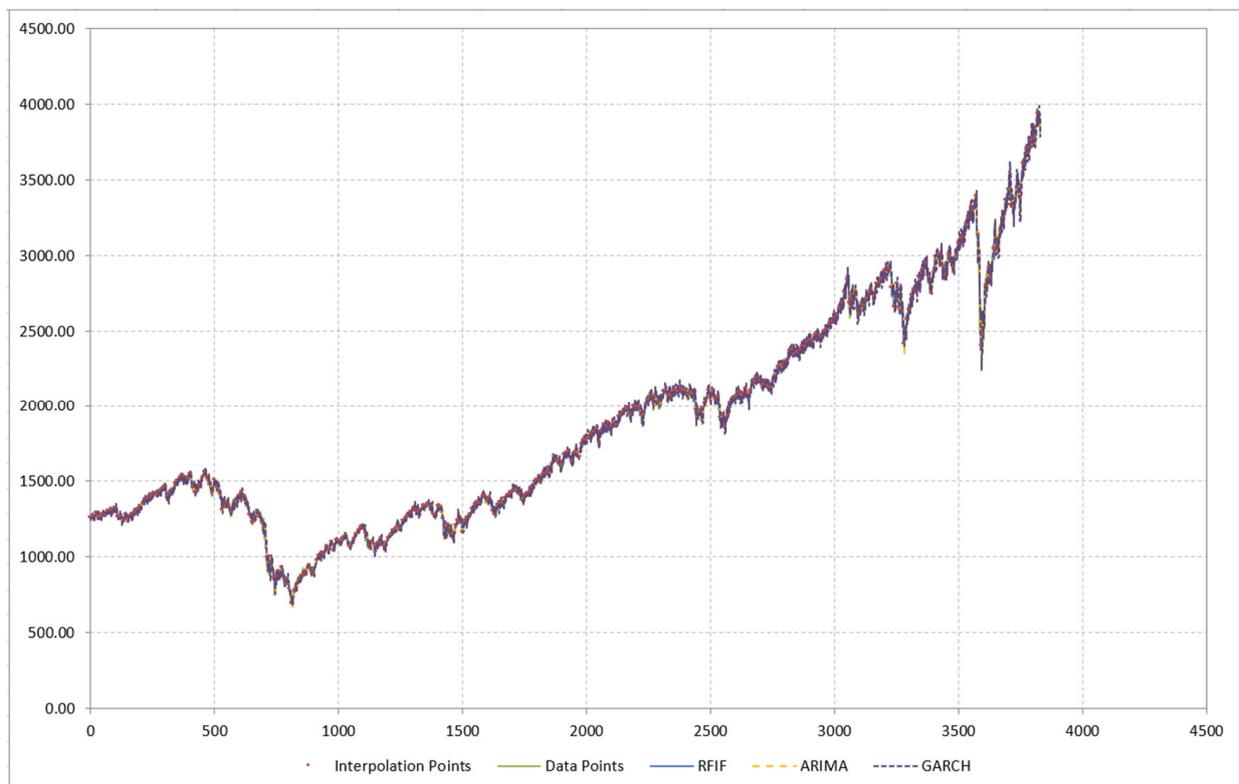


Figure 10. A time series of S&P 500 values (9 December 2005–March /2021) modelled by (i) a recurrent fractal interpolation function, (ii) the ARIMA(1, 1, 0) model, (iii) the GARCH(1,1) model.

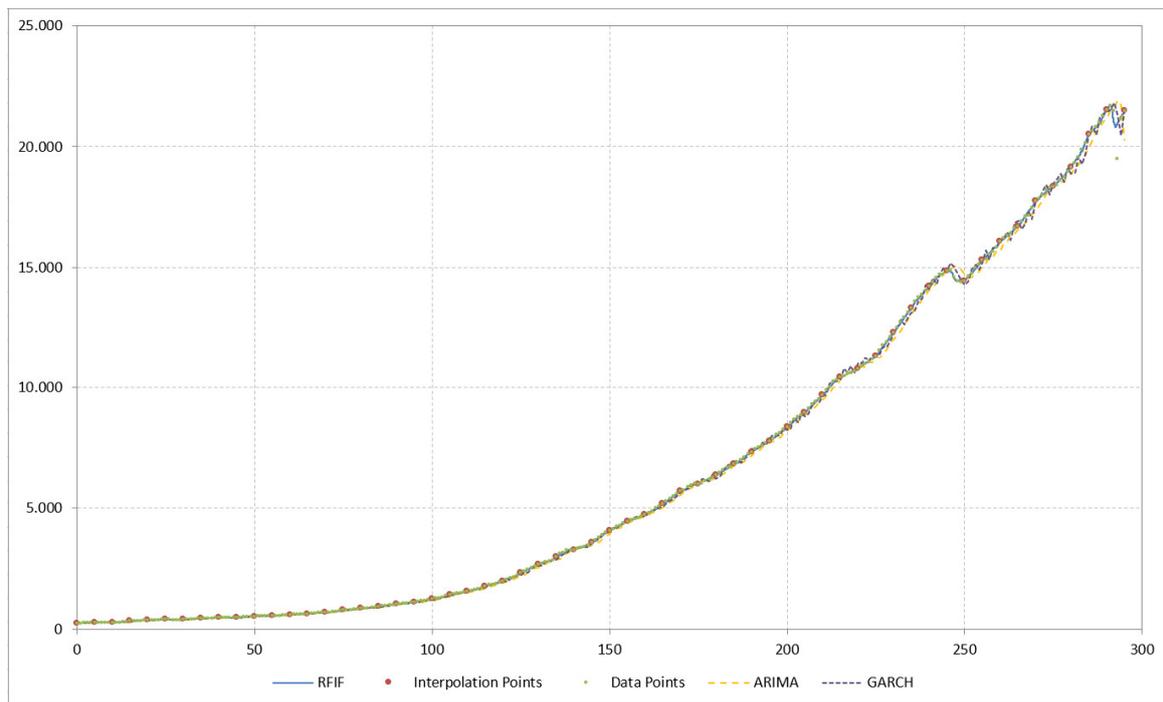


Figure 11. A time series of U.S. GDP values (1947-Q1–2020-Q4) modelled by (i) a recurrent fractal interpolation function, (ii) the ARIMA(1, 1, 0) model, (iii) the GARCH(1,1) model.

4. Conclusions

The application of fractal interpolation to financial time series was examined. Our motivation stems from the fact that financial time series are often non-smooth, presenting fluctuations and abrupt changes, thus rendering fractal interpolation functions a suitable candidate for their modelling. Our tests have included three different sets of data (bitcoin prices, S&P 500, U.S.A. GDP) with different time span and characteristics. The results indicate that fractal interpolation functions are successful in modelling financial time series even with a sparse selection of interpolation points. This stems from the fact that fractal interpolants can inherently model non-smooth sets of data which frequently occur in financial applications; moreover, they are able to model smooth structures as well.

The proposed methodology is expected to be especially useful in cases of non-smooth time series. Such examples would be prices/indexes of unstable assets such as cryptocurrencies, time series spanning a turbulent period of financial crisis, or time series of high frequency data. Moreover, it can be used for filling missing data within a time series. In this context, it is interesting to note that a fractal interpolant presents detail at all scales, so it can also be used for completing missing data even in higher frequencies than the original time series. Future work will focus on creating a generic, systematic framework for modelling and forecasting univariate as well as multivariate financial time series.

Author Contributions: Conceptualization, P.M., V.D. and E.P.; methodology, P.M.; software, P.M.; visualization, P.M.; investigation, P.M., V.D. and E.P.; validation, P.M., V.D. and E.P.; writing—original draft preparation, P.M.; writing—review and editing, P.M., V.D. and E.P.; supervision, P.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available upon request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tsay, R.S. *Analysis of Financial Time Series*, 3rd ed.; John Wiley & Sons: Hoboken, NJ, USA, 2010.
2. Taylor, S.J. *Modelling Financial Time Series*, 2nd ed.; World Scientific Publishing Co.: Singapore, 2008.
3. Barnsley, M.F. Fractal functions and interpolation. *Constr. Approx.* **1986**, *2*, 303–329. [[CrossRef](#)]
4. Barnsley, M.F. *Fractals Everywhere*, 3rd ed.; Dover Publications: Mineola, NY, USA, 2012.
5. Manousopoulos, P.; Drakopoulos, V.; Theoharis, T. Curve fitting by fractal interpolation. *Trans. Comput. Sci.* **2008**, *1*, 85–103.
6. Manousopoulos, P.; Drakopoulos, V. *On the Application of Fractal Interpolation Functions within the Reliability Engineering Framework, Statistical Modeling of Reliability Structures and Industrial Processes*; Taylor & Francis Group; CRC Press: Boca Raton, FL, USA, 2022.
7. Li, Z.; Han, J.; Song, Y. On the forecasting of high-frequency financial time series based on ARIMA model improved by deep learning. *J. Forecast.* **2020**, *39*, 1081–1097. [[CrossRef](#)]
8. Samitas, A.; Kampouris, E.; Polyzos, E.; Spyridou, A. Spillover effects between Greece and Cyprus: A DCC model on the interdependence of small economies. *Invest. Manag. Financ. Innov.* **2020**, *17*, 121–135.
9. Sun, H.; Yu, B. Forecasting financial returns volatility: A GARCH-SVR model. *Comput. Econ.* **2020**, *55*, 451–471. [[CrossRef](#)]
10. Pantos, T.; Polyzos, E.; Armenatzoglou, A.; Kampouris, E. Volatility spillovers in electricity markets: Evidence from the United States. *Int. J. Energy Econ. Policy* **2019**, *9*, 131–143. [[CrossRef](#)]
11. Atsalakis, G.S.; Valavanis, K.P. Surveying stock market forecasting techniques—Part II: Soft computing methods. *Expert Syst. Appl.* **2009**, *36*, 5932–5941. [[CrossRef](#)]
12. Lee, M.; Chen, C.D. The intraday behaviors and relationships with its underlying assets: Evidence on option market in Taiwan. *Int. Rev. Financ. Anal.* **2005**, *14*, 587–603. [[CrossRef](#)]
13. Bhattacharjee, B.; Kumar, R.; Senthilkumar, A. Unidirectional and bidirectional LSTM models for edge weight predictions in dynamic cross-market equity networks. *Int. Rev. Financ. Anal.* **2022**, *84*, 102384. [[CrossRef](#)]
14. Polyzos, E.; Fotiadis, A.; Samitas, A. *COVID-19 Tourism Recovery in the ASEAN and East Asia Region: Asymmetric Patterns and Implications*; ERIA Discussion Paper Series, Paper No. 379; ERIA: Jakarta, Indonesia, 2021.
15. Ozbayoglu, A.M.; Gudelek, M.U.; Sezer, O.B. Deep learning for financial applications: A survey. *Appl. Soft Comput.* **2020**, *93*, 106384. [[CrossRef](#)]
16. Henrique, B.M.; Sobreiro, V.A.; Kimura, H. Literature review: Machine learning techniques applied to financial market prediction. *Expert Syst. Appl.* **2019**, *124*, 226–251. [[CrossRef](#)]
17. Bailey, D.H.; Borwein, J.M.; de Prado, M.L.; Zhu, Q.J. Pseudomathematics and financial charlatanism: The effects of backtest over fitting on out-of-sample performance. *Not. AMS* **2014**, *61*, 458–471.
18. Chen, D.; Ye, J.; Ye, W. Interpretable selective learning in credit risk. *Res. Int. Bus. Financ.* **2023**, *65*, 101940. [[CrossRef](#)]
19. Ghosh, I.; Alfaro-Cortés, E.; Gámez, M.; García-Rubio, N. Prediction and interpretation of daily NFT and DeFi prices dynamics: Inspection through ensemble machine learning & XAI. *Int. Rev. Financ. Anal.* **2023**, *87*, 102558.
20. Goodell, J.W.; Jabeur, S.B.; Saâdaoui, F.; Nasir, M.A. Explainable artificial intelligence modeling to forecast bitcoin prices. *Int. Rev. Financ. Anal.* **2023**, *88*, 102702. [[CrossRef](#)]
21. Evertsz, C.J.G. Fractal geometry of financial time series. *Fractals* **1995**, *3*, 609–616. [[CrossRef](#)]
22. Richards, G.R. A fractal forecasting model for financial time series. *J. Forecast.* **2004**, *23*, 586–601. [[CrossRef](#)]
23. Kapecka, A. Fractal Analysis of Financial Time Series Using Fractal Dimension and Pointwise Holder Exponents. *Dyn. Econ. Model.* **2013**, *13*, 107–126.
24. Bhatt, S.J.; Dedania, H.V.; Shah Vipul, R. Fractal Dimensional Analysis in Financial Time Series. *Int. J. Financ. Manag.* **2015**, *5*, 46–52. [[CrossRef](#)]
25. Leon-Ogazon, M.A.; Romero-Flores, E.A.; Morales-Acoltzi, T.; Machorro-Rodríguez, A.; Salazar-Medina, M. Fractal Interpolation in the Financial Analysis of a Company. *Int. J. Bus. Adm.* **2017**, *8*, 80–86. [[CrossRef](#)]
26. Bianchi, S.; Pianese, A. Time-varying Hurst–Hölder exponents and the dynamics of (in)efficiency in stock markets. *Chaos Solitons Fractals* **2018**, *109*, 64–75. [[CrossRef](#)]
27. Cho, P.; Kim, K. Global Collective Dynamics of Financial Market Efficiency Using Attention Entropy with Hierarchical Clustering. *Fractal Fract.* **2022**, *6*, 562. [[CrossRef](#)]
28. Lee, M.; Cho, Y.; Ock, S.E.; Song, J.W. Analyzing Asymmetric Volatility and Multifractal Behavior in Cryptocurrencies Using Capital Asset Pricing Model Filter. *Fractal Fract.* **2023**, *7*, 85. [[CrossRef](#)]
29. Li, X.; Su, F. The Dynamic Effects of COVID-19 and the March 2020 Crash on the Multifractality of NASDAQ Insurance Stock Markets. *Fractal Fract.* **2023**, *7*, 91. [[CrossRef](#)]
30. Lu, K.-C.; Chen, K.-S. Uncovering Information Linkages between Bitcoin, Sustainable Finance and the Impact of COVID-19: Fractal and Entropy Analysis. *Fractal Fract.* **2023**, *7*, 424. [[CrossRef](#)]
31. Mazel, D.S.; Hayes, M.H. Using iterated function systems to model discrete sequences. *IEEE Trans. Signal Process.* **1992**, *40*, 1724–1734. [[CrossRef](#)]
32. Manousopoulos, P.; Drakopoulos, V.; Theoharis, T. Parameter identification of 1D fractal interpolation functions using bounding volumes. *J. Comput. Appl. Math.* **2009**, *233*, 1063–1082. [[CrossRef](#)]
33. Manousopoulos, P.; Drakopoulos, V.; Theoharis, T. Parameter Identification of 1D Recurrent Fractal Interpolation Functions with Applications to Imaging and Signal Processing. *J. Math. Imaging Vis.* **2011**, *40*, 162–170. [[CrossRef](#)]

34. Uemura, S.; Haseyama, M.; Kitajima, H. Efficient contour shape description by using fractal interpolation functions. In Proceedings of the IEEE Proceedings of International Conference on Image Processing, Rochester, NY, USA, 22–25 September; 2002; pp. 485–488.
35. Brinks, R. A hybrid algorithm for the solution of the inverse problem in fractal interpolation. *Fractals* **2005**, *13*, 215–226. [[CrossRef](#)]
36. Walther, T.; Klein, T.; Bouri, E. Exogenous drivers of Bitcoin and Cryptocurrency volatility—A mixed data sampling approach to forecasting. *J. Int. Financ. Mark. Inst. Money* **2019**, *63*, 101133. [[CrossRef](#)]
37. Martens, M. Measuring and forecasting S&P 500 index-futures volatility using high-frequency data. *J. Futures Mark. Futures Options Other Deriv. Prod.* **2002**, *22*, 497–518.
38. Yoon, J. Forecasting of real GDP growth using machine learning models: Gradient boosting and random forest approach. *Comput. Econ.* **2021**, *57*, 247–265. [[CrossRef](#)]
39. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. Available online: <https://bitcoin.org/bitcoin.pdf> (accessed on 2 December 2021).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.