AAAS
Research
Volume 2019, Article ID 3903187, 10 pages
https://doi.org/10.34133/2019/3903187

Research
OFFICIAL JOURNAL OF CAST

Research Article

Numerical Solution for the Extrapolation Problem of Analytic Functions

Nikolaos P. Bakas*

Intelligent Systems Lab & Civil Engineering Department, School of Architecture, Engineering, Land and Environmental Sciences, Neapolis University Pafos, 2 Danais Avenue, 8042 Paphos, Cyprus

*Correspondence should be addressed to Nikolaos P. Bakas; n.bakas@nup.ac.cy

Received 8 November 2018; Accepted 18 March 2019; Published 28 May 2019

Copyright © 2019 Nikolaos P. Bakas. Exclusive Licensee Science and Technology Review Publishing House. Distributed under a Creative Commons Attribution License (CC BY 4.0).

In this work, a numerical solution for the extrapolation problem of a discrete set of n values of an unknown analytic function is developed. The proposed method is based on a novel numerical scheme for the rapid calculation of higher order derivatives, exhibiting high accuracy, with error magnitude of $O(10^{-100})$ or less. A variety of integrated radial basis functions are utilized for the solution, as well as variable precision arithmetic for the calculations. Multiple alterations in the function's direction, with no curvature or periodicity information specified, are efficiently foreseen. Interestingly, the proposed procedure can be extended in multiple dimensions. The attained extrapolation spans are greater than two times the given domain length. The significance of the approximation errors is comprehensively analyzed and reported, for 5832 test cases.

1. Introduction

Prediction always differs from observation [1], and extrapolation remains an open challenge [2] or even a *hopelessly ill-conditioned* [3] problem, in a vast variety of scientific disciplines. It relies on numerical methods which attempt to predict the future, unknown values of a studied phenomenon, given a limited set of observations. Its importance is reflected in the Scopus database [4], by a search with the term extrapolation, yielding nearly 150 thousands of research items, in a wide range of scientific fields (Table S1). The visual presentation of the corresponding keywords (Figure S1) in a bibliometric map [5] reveals that extrapolation is closely associated with *modeling, simulation, uncertainty,* and *interpolation*, which is the reciprocal problem to simulate the unknown mechanism which produced the given data, aiming to predict utilizing the constructed model. Contradictorily to interpolation and despite the significance of extrapolation and forecasting methods, they exhibit a decreasing pattern in literature since the early 1980s (Figure S2), indicating the futility to predict for extended time frames. The given data is often called *time-series* and their extrapolation *forecasting*, which usually depend on machine learning algorithms such as support vector machines [6–8] and artificial neural networks [2, 9, 10], which construct nonlinear regression models, to predict beyond the known domain. Prediction algorithms are useful in a wide range of scientific disciplines, such as earth sciences [7, 9], finance [8, 11], computer science [12, 13], and engineering [14–16]. Several methods have been proposed for time-series forecasting [2] and competitions regarding accuracy have been conducted, utilizing statistical [11, 17] and machine learning procedures [2, 10, 15]; however, the prediction horizon regards only a small percentage (~ 20-30%) of the given data extension. Similarly, for less uncertain problems as the extrapolation of curves defined by polynomials (splines) [18–20], extrapolation can cause unpredictable results, and their extension should be short [21]. Apparently, prediction procedures are essential for a vast variety of scientific fields, always based on numerical interpolation methods.

More specifically, in the case of analytic functions, where the given data follow an unknown rigor mathematical law, any extrapolation results are too weak to be interesting [3]. Previous theoretical works focus on the interpolation problem, such as the ability of neural networks with one hidden layer to approximate analytic functions [22], proofs regarding the degree of rational approximation of an analytic function [23], and analysis of the rate of decrease of the best approximations of analytic functions [24]. Accuracy and convergence techniques are demonstrated in [25], yet it is

shown theoretically [26] that the approximation of analytic functions cannot converge exponentially. Approximation of analytic functions is investigated utilizing a variety of approximators [27–29], such as Hermite [30, 31] and Airy functions [32]. Computational works regard the approximation of a function [33] as well as its derivatives [34, 35], investigating the interpolation errors among the given nodal points. Such errors are higher than the approximation errors (Runge phenomenon) especially at the boundaries [33], affecting dramatically the extrapolation outside the given domain. Moreover, the round-off errors of computing machines follow some stochastic law [36] and recent works deal with the high impact of infinitesimal errors in the given data, utilizing extended arithmetic precision [37, 38]. Accordingly, the constitution of an accurate numerical method to approximate an analytic function and its derivatives, which is vital for the extrapolation problem, remains a challenge.

The purpose of this study was to provide a generic numerical solution for the extrapolation problem. It was attained for extended extrapolation horizons of even greater than 200% the given domain length if the set of data are derived from an unknown analytic function and their precision is high. The rationale of the proposed method adheres to the following three stages: (a) interpolation of the set of values $(f_1, f_2, \ldots, f_N)$ at specified points $(x_1, x_2, \ldots, x_N)$ using integrated radial basis functions (IRBFs), (b) computation of high order derivatives $\{f', f'', f^{(3)}, \ldots\}$ at any point of the closed domain $[x_1, x_N]$ with high accuracy based on a novel numerical scheme, and (c) successive application of the Taylor series formula to extrapolate f at points outside the given domain. The method is capable of interpolating within the given data with high precision, avoiding the Runge phenomenon at the boundaries [33], as well as of computing the higher order derivatives with remarkable accuracy, which are fundamental problems in numerous applications. The effects of the method's parameters on the prediction errors were extensively investigated by their analysis of variance (ANOVA) and feature extraction utilizing the Random Forest [39] method, for 5832 test cases. Illustrative examples of highly nonlinear analytic functions in two and three dimensions demonstrate the prediction extents attained by the proposed method.

2. Numerical Solution

2.1. Extrapolation of One Infinitesimal dx. Let f be an analytic function, which is unknown. It is given that the function takes values $\mathbf{b} = (f_1, f_2, \ldots, f_N)$ at specified points $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ as in Figure 1(a), for a generic analytic function. The extrapolation problem, that is, to predict the values which take the function outside the given domain $[x_1, x_N]$, can be transformed into the computation of n ordinary derivatives of f, $(f', f'', \ldots, f^{(n)})$, at point x_N. In particular, applying the Taylor formula from the point x_N to the next point $x_{N+1} = x_N + dx$, the corresponding value of f is given by

$$f(x_N + dx) = f(x_N) + dx f'(x_N) + \frac{dx^2}{2} f''(x_N) + \cdots + \frac{dx^n}{n!} f^{(n)}(x_N), \tag{1}$$

for an infinitesimal step dx. $f(x_N + dx)$ is the value of f at point $x_N + dx$, depending on the n ordinary derivatives of f at x_N. These n derivatives should be computed, as they are unknown.

2.2. Numerical Evaluation of the Derivatives. Around the endpoint x_N, we used the Taylor formula for computing the value of the function at the first predicted point x_{N+1}. In order to calculate the derivatives, we need to consider the function values within the end interval dx from the point with abscissa x_{N-1} to x_N. Accordingly, the Taylor expansion from the endpoint x_N to any previous point $h_j \longrightarrow x_N$ (Figure 1(b)) may also be written in the form

$$f(x_N - h_j) = f(x_N) - h_j f'(x_N) + \frac{h_j^2}{2} f''(x_N) - \cdots + \frac{(-h_j)^n}{n!} f^{(n)}(x_N). \tag{2}$$

By interchanging the positions of $f(x_N)$ and $f(x_N - h_j)$, we may write

$$-f(x_N) = -f(x_N - h_j) - h_j f'(h_j) + \frac{h_j^2}{2} f''(h_j) - \cdots + \frac{(-h_j)^n}{n!} f^{(n)}(h_j), \tag{3}$$

and by applying it to $n+1$ points h_j *within the end interval* dx (Figure 1(b)) and writing the resulting system of equations in matrix form, we obtain

$$-f(x_N) \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} -f(x_N - h_1) - h_1 f'(x_N) + \frac{h_1^2}{2} f''(x_N) - \cdots + \frac{(-h_1)^n}{n!} f^{(n)}(x_N) \\ -f(x_N - h_2) - h_2 f'(x_N) + \frac{h_2^2}{2} f''(x_N) - \cdots + \frac{(-h_2)^n}{n!} f^{(n)}(x_N) \\ \vdots \\ -f(x_N - h_{n+1}) - h_{n+1} f'(x_N) + \frac{h_{n+1}^2}{2} f''(x_N) - \cdots + \frac{(-h_{n+1})^n}{n!} f^{(n)}(x_N) \end{bmatrix}. \tag{4}$$

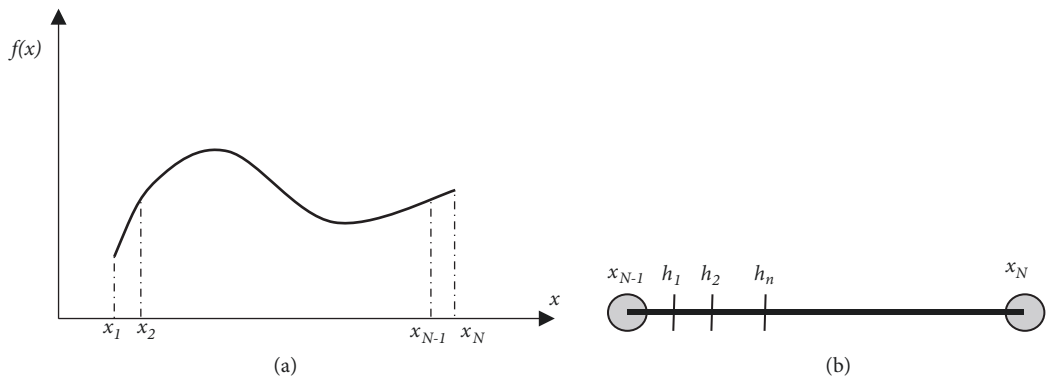


FIGURE 1: Given values of $f(x)$ (a) and interpolation within the edge interval (b).

It is important to note that all the derivatives are considered at the endpoint x_N. Hence, extracting the derivatives as a column vector, we have

$$-f(x_N)\begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix}=\left[\begin{array}{c|cccc}-f(x_N-h_1) & -h_1 & \frac{h_1^2}{2} & \cdots & \frac{(-h_1)^n}{n!}\\ -f(x_N-h_2) & -h_2 & \frac{h_2^2}{2} & \cdots & \frac{(-h_2)^n}{n!}\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ -f(x_N-h_{n+1}) & -h_{n+1} & \frac{h_{n+1}^2}{2} & \cdots & \frac{(-h_{n+1})^n}{n!}\end{array}\right]\begin{bmatrix}1\\f'(x_N)\\f''(x_N)\\\vdots\\f^{(n)}(x_N)\end{bmatrix}. \tag{5}$$

Thus, only the first column of the matrix in (5) varies as it refers to the values of f within the edge interval. As the objective is to compute the n ordinary derivatives of f, (5) is separated in submatrices, in order to rearrange the equations and develop a formula to calculate the vector $\mathbf{D}$, containing the values of the n derivatives of f at point x_N. The solid lines indicate the submatrices **B, C, H**, **D**, and $\mathbf{J}_{n,1}$; hence, (5) can be written in matrix partitions form

$$-f(x_N)\begin{bmatrix}1\\\mathbf{J}_{n,1}\end{bmatrix}=\begin{bmatrix}A & \mathbf{B}\\ \mathbf{C} & \mathbf{H}\end{bmatrix}\begin{bmatrix}1\\\mathbf{D}\end{bmatrix}, \tag{6}$$

where

$$A=-f(x_N-h_1), \tag{6a}$$

$$\mathbf{B}=\begin{bmatrix}-h_1 & \frac{h_1^2}{2} & \cdots & \frac{(-h_1)^n}{n!}\end{bmatrix}, \tag{6b}$$

$$\mathbf{C}=\begin{bmatrix}-f(x_N-h_2) & \cdots & -f(x_N-h_{n+1})\end{bmatrix}^T, \tag{6c}$$

$$\mathbf{H}=\begin{bmatrix}-h_2 & \frac{h_2^2}{2} & \cdots & \frac{(-h_2)^n}{n!}\\ \vdots & \vdots & \ddots & \vdots\\ -h_{n+1} & \frac{h_{n+1}^2}{2} & \cdots & \frac{(-h_{n+1})^n}{n!}\end{bmatrix}, \tag{6d}$$

$$\mathbf{D}=\begin{bmatrix}f'(x_N) & f''(x_N) & \cdots & f^{(n)}(x_N)\end{bmatrix}^T, \tag{6e}$$

$$\mathbf{J}_{n,1}=\begin{bmatrix}1 & \cdots & 1\end{bmatrix}^T. \tag{6f}$$

Since (6) is a system of $n+1$ equations, in order to calculate the vector $\mathbf{D}$ (containing the values of the derivatives of f at point x_N), by (6), we deduce

$$-f(x_N)\mathbf{J}_{n,1}=\mathbf{HD}+\mathbf{C}, \tag{7}$$

which is a system of n equations. Thus, the matrix of the n ordinary derivatives can be directly computed by

$$\mathbf{D}=\mathbf{H}^{-1}\left(-\mathbf{C}-f(x_N)\mathbf{J}_{n,1}\right), \tag{8}$$

where n, and hence the number of the calculated derivatives, can be arbitrarily selected. Equation (8) offers a direct computation of the n ordinary derivatives $\mathbf{D}$. Since $\mathbf{C}$ is still unknown, we can compute it, by interpolating f with IRBFs as defined subsequently.

2.3. Function Approximation with IRBFs. Radial Basis Functions (RBFs) networks are universal approximators while Integrated RBFs are capable of attaining precise approximation for a function and its derivatives [34, 35] as well as for the solution of partial differential equations [37, 38]. The essential formulation of RBFs, adopted in this work, is as follows.

Let x be a variable taking N values in a domain X. The known values of the function are given at the positions $x_1, x_2, \ldots, x_N$ as per Figure 1, while $\forall i \in (1, 2, \ldots, N-1)$ the interval length between two adjacent points x_i, x_{i+1} is equal to dx. The given values $f(x_i)$ of the unknown function f can be approximated with RBFs $\varphi(x)$, centered at the N points. The RBFs approximation is defined by

$$f(x_i) \cong \sum_{j=1}^{N} a_j \varphi\left(r_{ij}\right), \tag{9}$$

where α_j are the unknown weights for the approximation with RBFs, $r_{ij} = |x_i - x_j|$ represents the distance of a given point x_i in the domain from any other point x_j, and $\varphi(r_{ij})$ is the radial basis kernels such as Gaussian, Shifted Multiquadrics, and Sigmoid or their integrals, as in (14a), (14b), (14c), (14d), (14e), (14f). Applying (9) at the N given points results in the approximated values of f, $b_i = f(x_i)$; hence we deduce that

$$b_i = \sum_{j=1}^{N} a_j \varphi\left(r_{ij}\right). \tag{10}$$

If we write (10) N times ($i = 1, \ldots, N$) and define $\mathbf{\Phi}$ as a matrix containing the values of $\varphi(r_{ij})$, with i indicating the rows and j the columns of $\mathbf{\Phi}$, $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_N]^T$ a vector containing the unknown approximation weights, and $\mathbf{b} = [b_1, b_2, \ldots, b_N]^T$ a vector comprising the values of f at the nodal points x_N, we may write

$$\mathbf{b} = \mathbf{\Phi}\boldsymbol{\alpha}. \tag{11}$$

Hence we derive

$$\boldsymbol{\alpha} = \mathbf{\Phi}^{-1}\mathbf{b}. \tag{12}$$

The vector $\mathbf{b}$ is known, and the matrix $\mathbf{\Phi}$ can be calculated from the distances of the known points x_i and x_j, so the weights $\boldsymbol{\alpha}$ of the RBFs approximation of the function f are computed by numerically inverting matrix $\mathbf{\Phi}$ and multiplying it with vector $\mathbf{b}$. After the computation of vectors $\boldsymbol{\alpha}$, the radial basis functions $\varphi(r)$ can be recomputed for each intermediate point h_j, in order to approximate the values of the unknown function f at these points. Accordingly, each approximation vector $\boldsymbol{\varphi}_j$ which corresponds to the points h_j, $\boldsymbol{\varphi}_j = \varphi(|x_i - h_j|)$, when multiplied with the calculated weights $\boldsymbol{\alpha}$ of (12), results in the interpolated values $f(x_N - h_j)$ of (6c), and so

$$\mathbf{C} = \left[\boldsymbol{\varphi}_2 \;\; \boldsymbol{\varphi}_3 \;\; \cdots \;\; \boldsymbol{\varphi}_{n+1}\right]^T \boldsymbol{\alpha}, \tag{13}$$

where $\boldsymbol{\varphi}_j$ are the approximation vectors for each h_j and $\boldsymbol{\alpha}$ the RBFs weights. The vectors $\boldsymbol{\varphi}_j$ are computed utilizing (14a), (14b), (14c), (14d), (14e), (14f), resulting in the vector $\mathbf{C}$, which is the function's f approximation at points h_j (Figure 1(b)).

The definitions of the radial basis functions φ utilized in (12), (13) are specified in the following group of (14a), (14b), (14c), (14d), (14e), (14f): the Gaussian and the Shifted Logarithmic, as well as integrations of them, for one and two times. Equations (14a), (14b), (14c), (14d), (14e), (14f) are written concerning $r = |x - x_j|$, in order to be generic for any $x \in [x_1, x_N]$ and hence for points x_i and h_j.

The Gaussian RBF is defined by

$$\varphi(r) = e^{-r^2/c^2}, \tag{14a}$$

where c is a constant influencing the shape of φ [40–42]. By integration of the Gaussian kernel, we obtain

$$\varphi(r) = \int e^{-r^2/c^2}\, dx = \frac{\mathrm{c}}{2}\sqrt{\pi}\,\mathrm{erf}\left(\frac{\left(x - x_j\right)}{c}\right), \tag{14b}$$

and if we integrate for a second time, then

$$\begin{aligned}\varphi(r) &= \int \frac{\mathrm{c}}{2}\sqrt{\pi}\,\mathrm{erf}\left(\frac{\left(x - x_j\right)}{c}\right) dx \\ &= \frac{\left(\mathrm{c}^2 \mathrm{e}^{-(x-x_j)^2/c^2} + \mathrm{c}\sqrt{\pi}\left(x - x_j\right)\mathrm{erf}\left(\left(x - x_j\right)/c\right)\right)}{2},\end{aligned} \tag{14c}$$

where erf is the error function, $\mathrm{erf}(x) = (1/\sqrt{\pi})\int_{-x}^{x} e^{-t^2} dt$, exhibiting a sigmoid scheme, which is commonly used in ANNs [43].

Respectively, the Shifted Logarithmic RBF is defined by

$$\varphi(r) = \ln\left(r^2 + c^2\right), \tag{14d}$$

and by integration of the kernel we obtain

$$\begin{aligned}\varphi(r) &= \int \ln\left(r^2 + c^2\right) dx \\ &= \ln\left(\left(x - x_j\right)^2 + c^2\right)\left(x - x_j\right) - 2x \\ &\quad + 2ca\tan\frac{x - x_j}{c},\end{aligned} \tag{14e}$$

and if we integrate for a second time, then

$$\begin{aligned}\varphi(r) &= \int \ln\left(\left(x - x_j\right)^2 + c^2\right)\left(x - x_j\right) - 2x \\ &\quad + 2c\tan^{-1}\frac{x - x_j}{c} dx = xx_j \\ &\quad - \frac{\left(\left(x - x_j\right)c^2 + x^2 + {x_j}^2\right)\ln\left(\left(x - x_j\right)^2 + c^2\right) - 3x^2}{2} \\ &\quad + 2c\left(x - x_j\right)\tan^{-1}\left(\frac{x - x_j}{c}\right) - xx_j \\ &\quad \cdot \ln\left(\left(x - x_j\right)^2 + c^2\right).\end{aligned} \tag{14f}$$

2.4. Iterative Implementation of Taylor Series Expansion. Exploiting the computed derivatives with high accuracy at the end of the domain x_N, the predicted value of the function can be computed at the next step x_{N+1}, by the Taylor series

Input: set of values $\mathbf{b} = (f_1, f_2, \ldots, f_N)$ at specified points $\mathbf{x} = (x_1, x_2, \ldots, x_N)$, *n, IRBF kernel, c*
Output: set of values $\{f_{N+1}, f_{N+2}, \ldots\}$ at new at points $\{x_{N+1}, x_{N+2}, \ldots\}$
1: Formulate vectors $\boldsymbol{\varphi}_j$, matrix $\boldsymbol{\Phi}$, and Calculate $\boldsymbol{\Phi}^{-1}$ ◂Equations ((9), (10), (12), (14a), (14b), (14c), (14d), (14e), (14f))
2: Formulate matrix $\mathbf{H}$ and Calculate $\mathbf{H}^{-1}$ ◂Equation (6d)
3: Formulate vector $\mathbf{J}_{n,1}$ ◂Equation (6f)
3: **ForEach** $x_{New} = x_N + dx$
4: Calculate vectors $\boldsymbol{\alpha}$ and $\mathbf{C}$ ◂Equations ((12), (13))
5: Calculate vector $\mathbf{D}$ ◂Equation (8)
6: Calculate $f(x_{New})$ ◂Equation (1)
7: Renew vectors $\mathbf{b}$ & $\mathbf{x}$ ◂Equation (15a) and (15b)
9: **End**

Algorithm 1: Extrapolating given data of an unknown function.

expansion of (1). This procedure is repeated iteratively, for each next point $x_{N+2}, x_{N+3}, \ldots$ utilizing each time the shifted vectors

$$\mathbf{b} \longleftarrow \left[\mathbf{b}_{2:N} \quad f\left(x_N + dx\right)\right], \tag{15a}$$

$$\mathbf{x} \longleftarrow \left[\mathbf{x}_{2:N} \quad x_N + dx\right]. \tag{15b}$$

The N points are initially the positions where the function values are given. This procedure is iteratively applied using at each step the $N - 1$ previous values of the function (eliminating each time the first point) and one new value of the function, calculated by the procedure ((15a) and (15b)). Keeping constant the length dx between a predicted point and its subsequent, the matrix $\boldsymbol{\Phi}$ remains the same, as in the initial interpolation domain. The same stands for the inverted matrix $\boldsymbol{\Phi}^{-1}$ as well as the matrices $\boldsymbol{\varphi}_j$, and despite the sequential shifts outside the original domain, the relative distances r of (14a), (14b), (14c), (14d), (14e), (14f) remain the same. The same stands for the matrices $\mathbf{H}$ and $\mathbf{H}^{-1}$. Hence, the interpolation of the function within the end increment dx each time is accomplished by matrix multiplication and not inversion, which is performed only one time for $\boldsymbol{\Phi}$ and one time for $\mathbf{H}$. Utilizing this approach, during the iterations of the extrapolation procedure, the computational time is decreased dramatically, as these matrices are computed only once.

2.5. The Extrapolation Algorithm. In Algorithm 1, the procedure of extrapolating given values of f is demonstrated in algorithmic form, utilizing at each step the appropriate equations as formulated in Sections 2.1–2.4.

3. Results

3.1. Parameters Affecting the Method and Preparation of Dataset to Test Performance. The efficiency of the procedure was verified through numerical examples for a variety of highly nonlinear functions and their extrapolation spans. The extrapolation span is based on the accuracy of the interpolation method ((9), (10), (12), (13), (14a), (14b), (14c), (14d), (14e), (14f)) as well as the numerical differentiation ((1), (5), (6), (6a), (6b), (6c), (6d), (6e), (6f), (8)). As demonstrated in Figure 1(b), the intervals for the differentiation h_j are selected near the $N - 1^{\text{st}}$ node of the domain discretization, in order to achieve the highest possible accuracy, because this region is within the selected dx, and simultaneously, the interpolation error is minimized near this node, due to the Runge phenomenon [33]. Hence we select

$$h_j = dx - \frac{(j-1)\,dx}{10^l}, \tag{16}$$

where l indicates the limit distance near the node x_{N-1}. However, for extremely low values of h, the matrix $\mathbf{H}$ (6d) contains elements near zero, and its inversion grows into unstable one (high condition number and inversion errors).

The features which have an effect on the calculations, as well as their values (in parentheses) utilized in the parametric investigation, are the number of Taylor terms utilized, indicated as number of derivatives, (25, 50, 75); the number of computer digits used for the arbitrary precision calculations (500, 1000, 2000); the span of the given domain L (1/2, 1); the number of the divisions N (50, 100, 200); the limit l (5, 10, 20); the number of IRBFs integrations (0, 1, 2); the kernel of RBFs networks (Gaussian = 1, Shifted Logarithmic = 2); their shape parameter c (1/5, 1, 5); and the unknown function which is studied (sin(x), $e^{\cos(x)}$). Accordingly, the resulting database consists of 5832 records of test cases. The output of each parametric execution of the proposed procedure was the condition number of the $\mathbf{H}$ and $\boldsymbol{\Phi}$ matrices, their inversion errors, and the error of the first and second derivatives according to the proposed procedure. Moreover, the error of the first step of the extrapolation was computed (ε), in order to investigate its dependence on the problem statement parameters (N, L, dx, f) as well as the method's attributes (#derivatives, digits, kernel, c, #integrations).

3.2. Effects of Parameters on Extrapolation Accuracy and Computational Time. The $\varepsilon' = \log_{10}(|\varepsilon| + 10^{-323})$ was utilized as a measure for the extrapolation error, as the values of ε exhibit a variation within the domain $4.623 * 10^{-283}$ to the IEEE® arithmetic representation for positive infinity [44]. Given that the lower values of error ε' are important, the analysis of variance was conducted for the cases with values of $\varepsilon < 10^{-50}$; hence the resulting database consists of 2292 records. ANOVA found statistically significant differences between the means (MD) of ε' for 25 and 50 derivatives

(MD = 43.5803, p-value = $9.5606 * 10^{-10}$) and for 25 and 75 derivatives (MD = 54.2206, p-value = $9.5606 * 10^{-10}$) as demonstrated in Figure S3 and Table S2. Similarly, the number of digits (Figure S4 and Table S3) was found significant for the ε' (MD = 20.3576, p-value = $9.6963 * 10^{-10}$ for 500 to 1000 digits and MD = 33.9628, p-value = $9.5606 * 10^{-10}$ for 500 to 2000 digits). The number of divisions exhibits a clear difference as well, in Figure S5 and Table S4 (MD = 22.7727, p-value = $9.5606 * 10^{-10}$ for 50 to 100 divisions and MD = 49.6978, p-value = $9.5606 * 10^{-10}$ for 50 to 200 divisions). The univariate linear correlation of the condition number of $\mathbf{\Phi}$ with ε' (Figure S12) exhibits an R^2 of 0.2559 (p-value = $3.1608 * 10^{-149}$). For the condition number of $\mathbf{H}$ as well as the inversion errors of $\mathbf{H}$ and $\mathbf{\Phi}$, the correlations are even lower (Figures S13-S15); however, the majority of the values are 10^{323} for the condition of $\mathbf{H}$ and 10^{-323} for the inversion errors.

In order to further investigate more complex associations among the studied parameters and the extrapolation error ε', the random forests method [39] was utilized. The numerical dataset was divided into a train set (85% of observations) and a test set (15%) in order to constitute and investigate the reliability of the predictive model. The R^2 for the predicted versus actual ε' for the test set was 0.8954 (Figure S18), indicating a reliable model. The features significance was evaluated in terms of their contribution to the prediction error in the constituted model (Figure S19A), signifying high values for the studied function, the condition of $\mathbf{\Phi}$, the number of derivatives, and the domain length. Similarly, considering the computation time as dependent variable, the R^2 for the test set was 0.9658 and the highest predictive features were the number of divisions, the number of integrations, and number of digits (Figure S19B).

3.3. Illustrative Functions f and Their Extrapolation Spans. In Figure 2 the extrapolation results are demonstrated by four test functions as indicated. The initial domain contains only the known given data for x and $f(x)$, and after the vertical lines, each graph contains the exact function values, the predicted and the normalized extrapolation error $\varepsilon'' = (\log_{10}(|\varepsilon|) - \min(\log_{10}(|\varepsilon|)))/(\max(\log_{10}(|\varepsilon|)) - \min(\log_{10}(|\varepsilon|)))$, which take values in the [0, 1]. The text cases include a vast variety of analytic functions as well as their combinations. In Figures 2(a)–2(d) the given curvature or periodicity information is meager, compared to the predicted evolution of the function's values. Interestingly, the logged error plot exhibits a logarithmic scheme, indicating a weak form of stability [3].

In Figure 3, the proposed procedure is implied firstly in the x-axis and then in the y-axis for the function $f(x, y) = \sin(e^{2x}) + \cos(e^{3y/2})$. Given only the cyan region, the method can predict the highly nonlinear colored surface in the x, y, z space.

4. Discussion

4.1. Numerical Differentiation. Numerical differentiation is highly sensitive to noise [34], especially for the higher order derivatives. After an extensive examination the literature utilizing IRBFs [35, 45, 46] for the derivatives approximation, or for the solution of specific problems [47, 48] as well as other differentiation methods [49, 50] and a variety of formulations for IRBFs ((14a), (14b), (14c), (14d), (14e), (14f)), the proposed procedure accomplished striking accuracy, with error magnitude ε of $O(10^{-100})$ or less (Figures S3-S11 and supplementary database), for the derivatives' computation. This finally permitted the implementation of the Taylor method for significant extrapolation extents (Figures 2 and 3 and Data File S1). The numerical calculation of the derivatives with variable precision arithmetic offers high accuracy [37, 38]. The inverse problem of numerical integration exhibits lower errors with $\varepsilon \sim O(10^{-1000})$; however, the integration is less sensitive to—even small—errors in the given data [51]. Cheng [37], similarly to this work, found that the errors of the derivatives approximation are of one or more order of magnitude higher than the function's (Figures S16, S17, and Data File S1). However, the digits studied were 100-200, while the precise computation of the derivatives was vital for the extrapolation, and so for a higher number of digits. Similarly, Mai-Duy et al. [52] examined the compactly integrated radial basis functions, with errors for the derivatives $\varepsilon \sim O(10^{-10})$ for fifty digits accuracy.

4.2. RBFs' Matrix Φ and Shape Parameter. The approximation scheme ((9), (10), (12), (13), (14a), (14b), (14c), (14d), (14e), (14f)) permits a precise interpolation, within the given domain, with $\varepsilon \sim O(10^{-100})$ or less, as found in the numerical experiments (supplementary database). Accordingly, the approximation of the function's values is attainable with high accuracy, not only at the nodes x_i, but for any intermediate point between two given points and, hence, for points h_j (Figure 1(b)), permitting the precise calculation of vector $\mathbf{C}$ (13). Arbitrary precision found capable of avoiding the ill-conditioning of $\mathbf{\Phi}$ was also indicated by Cheng [37] and Huang [38]. Using arbitrary precision in the calculations, the inversion errors in $\mathbf{\Phi\Phi^{-1}}$-$\mathbf{I}$ were found of $O(<10^{-100})$ or even exact zero (supplementary database). The arbitrary precision has also been used for the robust computation of $\mathbf{\Phi}$ by Cheng [37], where the Gaussian RBFs and inversed multiquadrics exhibit accuracy of interpolation with $\varepsilon \sim O(\sim 10^{-20}$-$10^{-30})$.

The association of the condition number of $\mathbf{\Phi}$ with ε' exhibits low R^2 (0.25595) with slightly negative slope (Figure S12), although the high condition number is considered to increase instability [37, 41, 52] indicating the need for high precision to the calculations; however the majority of the values are 10^{323}, that is, the maximum real number considered by the software [44]. Sensitivity analysis was selected instead of optimizing each method's parameters such as times of RBFs integration, the kernel function, or its shape parameter [40–42], as the interpolation errors in a number of the test cases were equal to zero at the nodes, eliminating any relevant objective function. The number of integrations of RBF φ increased computational time, as it magnifies the $\mathbf{\Phi}$ and $\mathbf{\Phi^{-1}}$ matrices, due to the more complex formulas ((14b), (14c), (14e), (14f)).

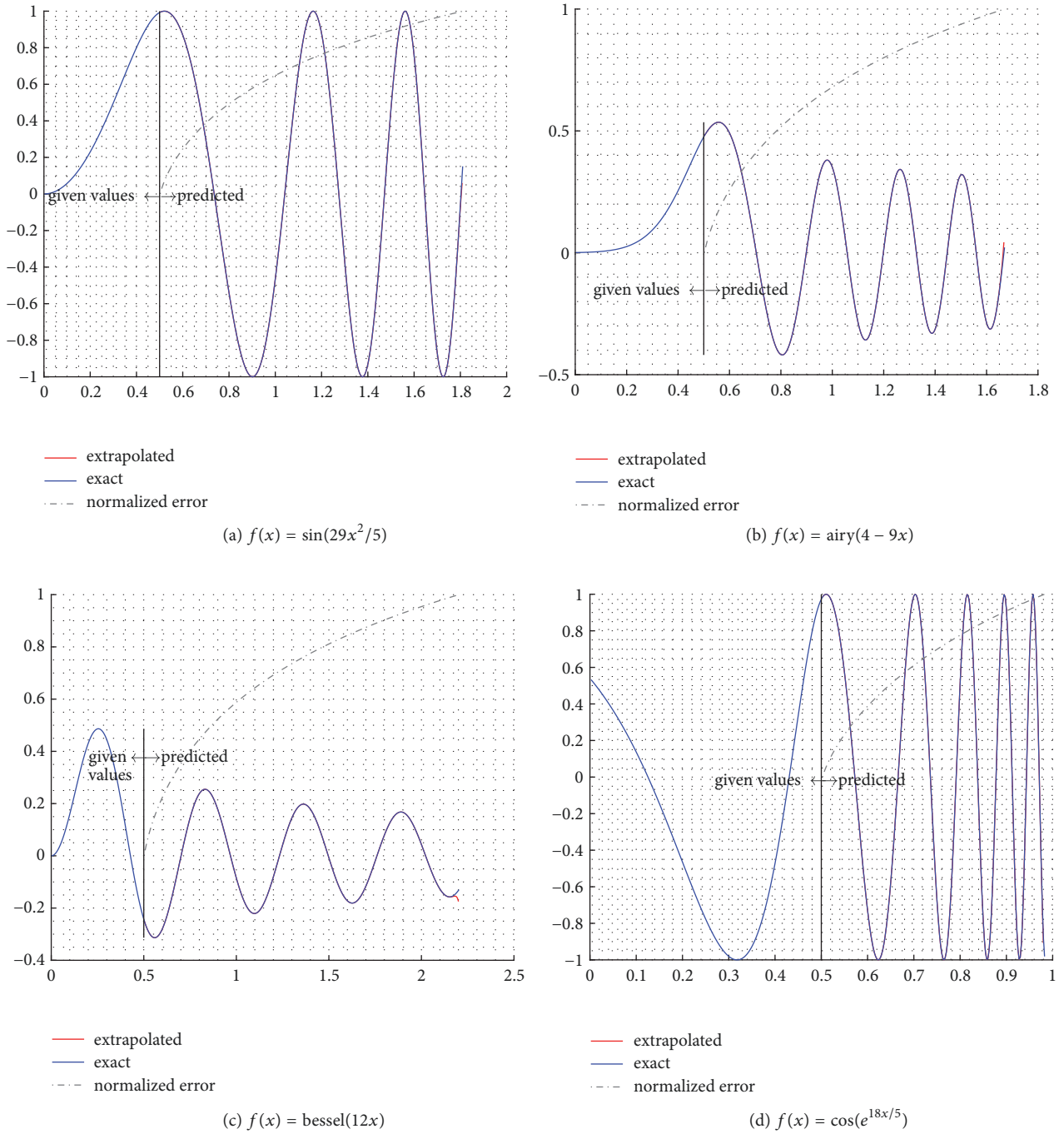


Figure 2: Extrapolation of analytic functions $f(x)$. The prediction starts after the vertical line.

4.3. Summary of Findings and Limitations of the Proposed Solution. In brief, a method for the extrapolation of analytic functions is developed, accompanied by a systematic investigation of the involved parameters. The proposed scheme for the numerical differentiation exhibited low enough errors to permit adequate extrapolation spans. The constituted database of the numerical experiments highlights the fact that, for the same problem formulation (L, N, dx, f), the derivatives calculation exhibits a high variation, indicating the vagueness of the transference from the presumed theory to the actual calculations. The numerical investigation of the extrapolation errors suggests that only utilizing high accuracy and a precise approximation scheme for a function as well as its derivatives, the extrapolation is attainable. Thus, for real-world phenomena, with laboratory accuracies even of $O(10^{-20})$, the predictions are limited to short length. Only if the measurements contain some hundreds of significant digits, the proposed solution is efficient. As this is difficult to be accomplished by laboratory instruments, this work's findings provide strong evidence that we are far from lengthy predictions in physics, biology, and engineering and also even more far from phenomena studied in health and social

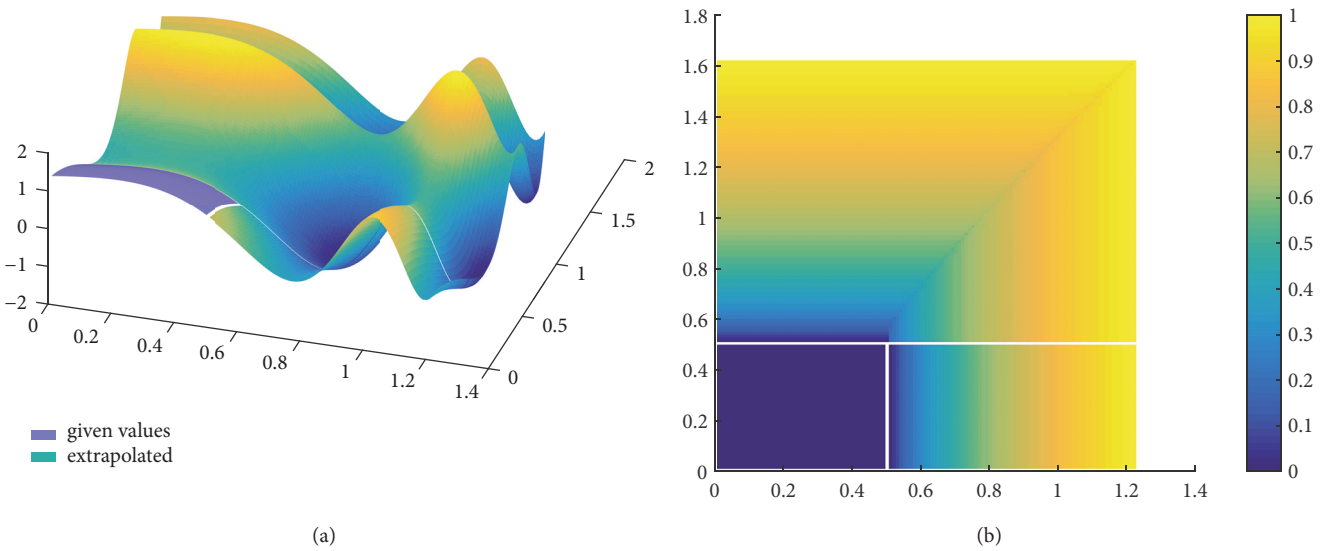


FIGURE 3: Extrapolation of function $f(x, y) = \sin(e^{2x}) + \cos(e^{3y/2})$ (a) and normalized errors (b).

sciences. However, the parametric investigation suggests that the precision in calculations and the utilized methods are vastly significant, as the extrapolation horizons achieved by the proposed numerical scheme are about an order of magnitude higher than those in the existing literature, highlighting the potentiality of predictions.

5. Materials and Methods

All the software operations ran on an Intel i7-6700 CPU @3.40GHz with 32GB memory and SSD hard disk, and the execution time was accordingly tracked. The errors of the derivatives in the relevant literature are reported in the Supplementary Data File S1, and they highlight the immense accuracy of numerical differentiation achieved in this work.

Conflicts of Interest

The author declares no conflicts of interest.

Acknowledgments

This study would not have been possible without professor's John T. Katsikadelis helpful discussions, valuable suggestions, and clarifications regarding the integrated radial basis functions as well as his confidence that the problem is solvable. Special thanks are due to Dr. Natia R. Anastasi for reviewing the original manuscript. I would also like to thank Dr. Savvas Chatzichristofis, Director of Informatics at Neapolis University Pafos, Nikitas Giannakopoulos, Ass. Professor of Medicine, University of Würzburg, Germany, and Angeliki Sivitanidou and Thomas Dimopoulos, Lecturers at Neapolis University, for suggestions regarding language.

Supplementary Materials

The supplementary materials are divided into two parts: (A) The supplementary figures and tables for the statistical analyses of parameters affecting extrapolation accuracy, and the bibliometric review figures. (B) The supplementary database: Data File S1. experiments.xlsx (attached): database for the parametric investigation of extrapolation errors. Figure S1: bibliometric map of papers with keyword extrapolation. Utilizing a bibliometric procedure [5] for a co-keywords based bibliometric analysis for the top 2000 papers in terms of their citations, containing extrapolation as keyword, the keywords highly associated with extrapolation—closer on the bibliometric map—are interpolation, risk assessment, prediction, uncertainty, modeling, etc. The almost coincident positions of extrapolation and interpolation indicate that the techniques aiming to extend a series of values beyond the observed domain are mainly based on interpolation procedures, which is an effort to mathematically formulate the phenomenon [53]. Figure S2: Evolution of terms in Google Books. Commencing the eighties and on, the effort of the researchers is decreasing, reflecting the failure to predict time-series values. This is demonstrated through a search in over 25 million—as of October 2015—books [54], for the words forecasting and extrapolation and the closely related topic of interpolation [53]. Figure S3: variance of ε' for number of derivatives. Figure S4: variance of ε' for number of digits. Figure S5: variance of ε' for number of divisions. Figure S6: variance of ε' for number of integrations. Nonsignificant differences (as per Table S5). Figure S7: variance of ε' for domain length. Nonsignificant differences (as per Table S6). Figure S8: variance of ε' for function. The function $\sin(x)$ exhibits less errors than the more complex $e^{\cos(x)}$ (MD = 15.6422, p-value = $1.0597 * 10^{-10}$, from Table S7). Figure S9: variance of ε' for kernel. The Gaussian kernel exhibits lower errors than the Shifted Logarithmic (MD = 9.9051, p-value = $3.3708 * 10^{-7}$ - Table S8). Figure S10: variance of ε' for limit l. Nonsignificant differences (as per Table S9). Figure S11: variance of ε' for shape parameter. The shape parameter 0.2 exhibits higher errors than 5 (MD = 8.5855, p-value = 0.0065 - Table S10). Figure S12: scatter plot for condition number of

$\mathbf{\Phi}$ and ε'. Figure S13: scatter plot for condition number of $\mathbf{H}$ and ε'. Figure S14: scatter plot for inversion error of $\mathbf{\Phi}$ and ε'. Figure S15: scatter plot for inversion error of $\mathbf{H}$ and ε'. Figure S16: scatter plot for error of first derivative and ε'. Figure S17: scatter plot for error of second derivative and ε'. Figure S18: scatter plot for actual versus predicted ε'. Table S2: differences between the means of ε' for groups with an uneven number of derivatives. Table S3: differences between the means of ε' for groups with an uneven number of digits. Table S4: differences between the means of ε' for groups with an uneven number of divisions. Table S5: differences between the means of ε' for groups with an uneven number of integrations. Table S6: differences between the means of ε' for groups with uneven domain lengths. Table S7: differences between the means of ε' for groups with uneven functions f. Table S8: differences between the means of ε' for groups with uneven RBF kernel. Table S9: differences between the means of ε' for groups with uneven limit l. Table S10: differences between the means of ε' for groups with uneven RBFs shape parameter. Figure S19: features significance. *(Supplementary Materials)*

References

[1] N. Oreskes, K. Shrader-Frechette, and K. Belitz, "Verification, validation, and confirmation of numerical models in the earth sciences," *Science*, vol. 263, no. 5147, pp. 641–646, 1994.

[2] S. Ben Taieb, G. Bontempi, A. F. Atiya, and A. Sorjamaa, "A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition," *Expert Systems with Applications*, vol. 39, no. 8, pp. 7067–7083, 2012.

[3] L. Demanet and A. Townsend, *Stable Extrapolation of Analytic Functions*, 2016, https://arxiv.org/abs/1605.09601.

[4] "Scopus," http://www.scopus.com/scopus/homeurl.

[5] V. Plevris, N. Bakas, G. Markeset, and J. Bellos, "Literature review of masonry structures under earthquake excitation utilizing machine learning algorithms," in *Proceedings of the 6th International Conference on Computational Methods in Structural Dynamics and Earthquake Engineering, COMPDYN 2017*, pp. 2685–2694, Greece, June 2017.

[6] N. Sapankevych and R. Sankar, "Time series prediction using support vector machines: a survey," *IEEE Computational Intelligence Magazine*, vol. 4, no. 2, pp. 24–38, 2009.

[7] S. Ahmad, A. Kalra, and H. Stephen, "Estimating soil moisture using remote sensing data: a machine learning approach," *Advances in Water Resources*, vol. 33, no. 1, pp. 69–80, 2010.

[8] H. Yang, K. Huang, I. King, and M. R. Lyu, "Localized support vector regression for time series prediction," *Neurocomputing*, vol. 72, no. 10–12, pp. 2659–2669, 2009.

[9] F. Recknagel, "Applications of machine learning to ecological modelling," *Ecological Modelling*, vol. 146, no. 1-3, pp. 303–310, 2001.

[10] S. F. Crone, M. Hibon, and K. Nikolopoulos, "Advances in forecasting with neural networks? Empirical evidence from the NN3 competition on time series prediction," *International Journal of Forecasting*, vol. 27, no. 3, pp. 635–660, 2011.

[11] S. Makridakis, C. Chatfield, M. Hibon et al., "The M2-competition: A real-time judgmentally based forecasting study," *International Journal of Forecasting*, vol. 9, no. 1, pp. 5–22, 1993.

[12] J. F. Murray, G. F. Hughes, and K. Kreutz-Delgado, "Machine learning methods for predicting failures in hard drives: a multiple-instance application," *Journal of Machine Learning Research*, vol. 6, pp. 783–816, 2005.

[13] C. Richard, J. C. Bermudez, and P. Honeine, "Online prediction of time serise data with kernels," *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 1058–1067, 2008.

[14] T. Hong, P. Pinson, and S. Fan, "Global energy forecasting competition 2012," *International Journal of Forecasting*, vol. 30, no. 2, pp. 357–363, 2014.

[15] P. Young, "Technological growth curves. A competition of forecasting models," *Technological Forecasting & Social Change*, vol. 44, no. 4, pp. 375–389, 1993.

[16] B.-J. Chen, M.-W. Chang, and C.-J. Lin, "Load forecasting using support vector machines: A study on EUNITE Competition 2001," *IEEE Transactions on Power Systems*, vol. 19, no. 4, pp. 1821–1830, 2004.

[17] S. Makridakis, A. Andersen, R. Carbone et al., "The accuracy of extrapolation (time series) methods: Results of a forecasting competition," *Journal of Forecasting*, vol. 1, no. 2, pp. 111–153, 1982.

[18] Y. Lu, K. Shi, J. Yong, H. Gu, and H. Song, "A B-spline curve extension algorithm," *Science China Information Sciences*, vol. 59, no. 3, Article ID 32103, 2016.

[19] G. Zhang and R. Little, "Extensions of the penalized spline of propensity prediction method of imputation," *Biometrics: Journal of the International Biometric Society*, vol. 65, no. 3, pp. 911–918, 2009.

[20] H. Fan, C. M. Zhang, and J. J. Li, "Extension algorithm for B-splines with GC^2-continuous," *Chinese Journal of Computers*, vol. 28, no. 6, pp. 933–938, 2005.

[21] S. Shetty and P. R. White, "Curvature-continuous extensions for rational B-spline curves and surfaces," *Computer-Aided Design*, vol. 23, no. 7, pp. 484–491, 1991.

[22] H. N. Mhaskar, "Neural networks for optimal approximation of smooth and analytic functions," *Neural Computation*, vol. 8, no. 1, pp. 164–177, 1996.

[23] A. A. Gonchar and E. A. Rakhmanov, "Equilibrium distributions and the rate of rational approximation of analytic functions," *Mathematics of the USSR – Sbornik*, vol. 62, no. 2, pp. 305–348, 1989.

[24] V. A. Prokhorov, "Rational approximation of analytic functions," *Sbornik: Mathematics*, vol. 78, no. 1, pp. 139–164, 1994.

[25] C. Schwartz, "High-accuracy approximation techniques for analytic functions," *Journal of Mathematical Physics*, vol. 26, no. 3, pp. 411–415, 1985.

[26] R. B. Platte, L. N. Trefethen, and A. B. J. Kuijlaars, "Impossibility of fast stable approximation of analytic functions from equispaced samples," *SIAM Review*, vol. 53, no. 2, pp. 308–318, 2011.

[27] R. Kargar, N. R. Pascu, and A. Ebadian, "Locally univalent approximations of analytic functions," *Journal of Mathematical Analysis and Applications*, vol. 453, no. 2, pp. 1005–1021, 2017.

[28] T. CoşKun, "Approximation of analytic functions of several variables by linear k-positive operators," *Turkish Journal of Mathematics*, vol. 41, no. 2, pp. 426–435, 2017.

[29] M. Griebel and J. Oettershagen, "On tensor product approximation of analytic functions," *Journal of Approximation Theory*, vol. 207, pp. 348–379, 2016.

[30] S.-M. Jung, "Approximation of analytic functions by Hermite functions," *Bulletin des Sciences Mathematiques*, vol. 133, no. 7, pp. 756–764, 2009.

[31] Z. Bustamante and G. Lopes Lagomasino, "Hermite-pade approximation for nikishin systems of analytic functions," *Sbornik Mathematics*, vol. 77, no. 2, pp. 367–384, 1994.

[32] S.-M. Jung, "Approximation of analytic functions by Airy functions," *Integral Transforms and Special Functions*, vol. 19, no. 11-12, pp. 885–891, 2008.

[33] B. Fornberg and J. Zuev, "The Runge phenomenon and spatially variable shape parameters in RBF interpolation," *Computers & Mathematics with Applications*, vol. 54, no. 3, pp. 379–398, 2007.

[34] N. Mai-Duy and T. Tran-Cong, "Approximation of function and its derivatives using radial basis function networks," *Applied Mathematical Modelling*, vol. 27, no. 3, pp. 197–220, 2003.

[35] C. Shu and Y. L. Wu, "Integrated radial basis functions-based differential quadrature method and its performance," *International Journal for Numerical Methods in Fluids*, vol. 53, no. 6, pp. 969–984, 2007.

[36] G. Turchetti, S. Vaienti, and F. Zanlungo, "Asymptotic distribution of global errors in the numerical computations of dynamical systems," *Physica A: Statistical Mechanics and its Applications*, vol. 389, no. 21, pp. 4994–5006, 2010.

[37] A. H. Cheng, "Multiquadric and its shape parameter—a numerical investigation of error estimate, condition number, and round-off error by arbitrary precision computation," *Engineering Analysis with Boundary Elements*, vol. 36, no. 2, pp. 220–239, 2012.

[38] C.-S. Huang, C.-F. Lee, and A. H.-D. Cheng, "Error estimate, optimal shape factor, and high precision computation of multiquadric collocation method," *Engineering Analysis with Boundary Elements*, vol. 31, no. 7, pp. 614–623, 2007.

[39] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[40] S. Rippa, "An algorithm for selecting a good value for the parameter c in radial basis function interpolation," *Advances in Computational Mathematics*, vol. 11, no. 2-3, pp. 193–210, 1999.

[41] J. G. Wang and G. R. Liu, "On the optimal shape parameters of radial basis functions used for 2-D meshless methods," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 23-24, pp. 2611–2630, 2002.

[42] G. E. Fasshauer and J. G. Zhang, "On choosing "optimal" shape parameters for RBF approximation," *Numerical Algorithms*, vol. 45, no. 1–4, pp. 345–368, 2007.

[43] M. Hou and X. Han, "Constructive approximation to multivariate function by decay RBF neural network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 21, no. 9, pp. 1517–1523, 2010.

[44] *MATLAB, version R2017a*, MathWorks Inc., Natick, Mass, USA, 2017.

[45] C. M. T. Tien, N. Thai-Quang, N. Mai-Duy, C.-D. Tran, and T. Tran-Cong, "A three-point coupled compact integrated RBF scheme for second-order differential problems," *CMES: Computer Modeling in Engineering & Sciences*, vol. 104, no. 6, pp. 425–469, 2015.

[46] C. M. T. Tien, N. Pham-Sy, N. Mai-Duy, C.-D. Tran, and T. Tran-Cong, "A high-order coupled compact integrated RBF approximation based domain decomposition algorithm for second-order differential problems," *CMES: Computer Modeling in Engineering & Sciences*, vol. 104, no. 4, pp. 251–304, 2015.

[47] N. G. Babouskos and J. T. Katsikadelis, "Optimum design of thin plates via frequency optimization using BEM," *Archive of Applied Mechanics*, vol. 85, no. 9-10, pp. 1175–1190, 2015.

[48] A. J. Yiotis and J. T. Katsikadelis, "Buckling of cylindrical shell panels: a MAEM solution," *Archive of Applied Mechanics*, vol. 85, no. 9-10, pp. 1545–1557, 2015.

[49] Y. Fu and J. Zhang, "Size-dependent pull-in phenomena in electrically actuated nanobeams incorporating surface energies," *Applied Mathematical Modelling: Simulation and Computation for Engineering and Environmental Systems*, vol. 35, no. 2, pp. 941–951, 2011.

[50] K. Hong and J. Chen, *Boundary Element Method*, Taipei, Taiwan, 1993.

[51] D. H. Bailey, K. Jeyabalan, and X. S. Li, "A comparison of three high-precision quadrature schemes," *Experimental Mathematics*, vol. 14, no. 3, pp. 317–329, 2005.

[52] N. Mai-Duy, T. T. Le, C. M. Tien, D. Ngo-Cong, and T. Tran-Cong, "Compact approximation stencils based on integrated flat radial basis functions," *Engineering Analysis with Boundary Elements*, vol. 74, pp. 79–87, 2017.

[53] S. Makridakis and N. Bakas, "Forecasting and uncertainty: a survey," *Risk and Decision Analysis*, vol. 6, no. 1, pp. 37–64, 2016.

[54] S. Heyman, *Google Books: a Complex and Controversial Experiment*, New York Times, 2015.