Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 6615601, 21 pages
https://doi.org/10.1155/2021/6615601

Research Article

Use of Multiquadric Functions for Multivariable Representation of the Aerodynamic Coefficients of Airfoils

Filipe Ribeiro,[1] Pedro Albuquerque,[2] Pedro Gamboa,[3] and Kouamana Bousson[4]

[1]*Universidade da Beira Interior, Covilhã, Portugal*
[2]*CEiiA Engineering & Product Development Center, Évora, Portugal*
[3]*Centre for Mechanical and Aerospace Science and Technologies (C-MAST-UBI), Universidade da Beira Interior, Covilhã, Portugal*
[4]*Associated Laboratory for Energy, Transports and Aeronautics (LAETA-UBI), Universidade da Beira Interior, Covilhã, Portugal*

Correspondence should be addressed to Pedro Gamboa; pgamboa@ubi.pt

Received 3 November 2020; Revised 29 March 2021; Accepted 8 April 2021; Published 23 April 2021

Academic Editor: Wubshet Ibrahim

Copyright © 2021 Filipe Ribeiro et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Given an array (or matrix) of values for a function of one or more variables, it is often desired to find a value between two given points. Multivariable interpolation and approximation by radial basis functions are important subjects in approximation theory that have many applications in Science and Engineering fields. During the last decades, radial basis functions (RBFs) have found increasingly widespread use for functional approximation of scattered data. This research work aims at benchmarking two different approaches: an approximation by radial basis functions and a piecewise linear multivariable interpolation in terms of their effectiveness and efficiency in order to conclude about the advantages and disadvantages of each approach in approximating the aerodynamic coefficients of airfoils. The main focus of this article is to study the main factors that affect the accuracy of the multiquadric functions, including the location and quantity of centers and the choice of the form factor. It also benchmarks them against piecewise linear multivariable interpolation regarding their precision throughout the selected domain and the computational cost required to accomplish a given amount of solutions associated with the aerodynamic coefficients of lift, drag and pitching moment. The approximation functions are applied to two different multidimensional cases: two independent variables, where the aerodynamic coefficients depend on the Reynolds number (Re) and the angle-of-attack (α), and four independent variables, where the aerodynamic coefficients depend on Re, α, flap chord ratio (c_{flap}), and flap deflection (δ_{flap}).

1. Introduction

The application of multiquadric functions to approximate a given set of sampling points was introduced for the first time in 1968 by Hardy, mainly due to the limitations of the various harmonic and polynomial series to represent topography surfaces. This new analytical method intends to represent irregular surfaces through the sum of quadric surfaces that emerge from certain points of interest within the domain of the function, which are called center points (or abbreviated just by centers), which are radially symmetrical with respect to that same point. The multiquadric function is associated with unknown parameters that need to be calculated to create an approximation with low interpolation error and with a realistic evolution between any two points belonging to the domain, this being the most laborious task of the whole procedure [1]. In 1982, Franke made an important step in the evolution of the method, comparing 29 different interpolation algorithms (including the multiquadric functions) for a set of 6 different functions, having concluded that the multiquadric functions present numerous advantages over the other methods, with respect to the precision and evolution of the surface between two known points, obtaining a smoother and visually more appealing surface [2]. Micchelli analyzed the coefficients of the radial basis functions (RBFs) matrices, with different numbers of centers and proved that these matrices are invertible and positively defined [3]. Another application of

great interest based on the RBFs was developed by Kansa and Sarra, to obtain approximate solutions for systems of differential equations [4–6]. In the present work, however, the motivation is the creation of an approximate solution of a sample set of data in the form of multiquadric functions.

There are essentially two parameters that greatly affect the accuracy of the method: the location and number of centers and the choice of a parameter called form factor. As far as the location of centers within the domain is concerned, according to Hardy [1, 7], the best option is a placement only in existing data points, since these are the origin of the quadric surface, a zone whose inclination is also dependent on the value of the form factor, which affects the smoothness of the surface. As far as the form factor is concerned, its choice has a huge impact on the convergence and accuracy of the method. Over the years, there have been several attempts by various authors to numerically, or experimentally, calculate the optimal form factor, but to this day there is still no consensus-building method. The three most accepted and most simplistic methods were introduced by Hardy [1], Franke [2], and Fasshauer [8], whose value depends essentially on the number and location of the centers. There are, however, other more complex methods for its determination, whose objective is the minimization of the interpolation error, being that the most widely accepted was introduced by Rippa [9]. A smaller form factor typically corresponds to a flatter surface with more abrupt contours, whereas a larger form factor is characterized by smoother surface evolutions. There are typically five parameters that dictate whether the approximation function requires a greater or lesser form factor: (1) the known sampling points evolution, (2) the RBF used, (3) the domain, (4) the spacing between the sampling points, and (5) the location and number of center points [10]. More recent studies have attempted to find the optimal form factor for each quadric surface, depending on their local properties, so that a distinct form factor for each center applied in the approximation would exist [11–14]. Another solution, but with the same objective, is to use a high form factor (which goes against Hardy's ideas) and constant for all the quadrics used and then find a way to correct the bad conditioning of the interpolation matrix, making efforts to lower their conditioning number through computational algorithms [6, 15–17]. There are also studies using optimization methods, which aim to find the best constant factor to reach any previously defined goal [18–21]. The main conclusion of recent studies regarding the increase of the form factor in the multiquadric functions is that, as the approximation function generates flatter and softer curves, it becomes also more insensitive to the Euclidean distance between the center point and any known sampling point, and the elements of the interpolation matrix become almost equal, which makes the matrix ill-conditioned or even singular in the limit situation. Other recent studies have proposed automatic curve fitting methodologies based on RBF [22] and adaptive methods for eigenvalue problems [23], initial value problems [24], and noise reduction problems [25], among other applications.

Another recent application of multiquadric functions is associated with surrogate models, an engineering method used to estimate the value of any parameter of interest that cannot be directly measured. Most engineering works, before being implemented on a real scale, go through a phase of experimentation and/or simulation to evaluate if the objectives that they propose are feasible, with the advantage of reducing associated costs if any failure is detected. From the aeronautical point of view, in order to find the optimum airfoil to apply to an aircraft wing, so as to optimize a given flight condition or even a set of flight conditions, an engineer must carry out computational and/or experimental simulations to understand the effect of the shape variation of the airfoil (e.g., chord, camber, thickness, and material) which sometimes requires several days or weeks of waiting. To overcome this barrier, surrogate models [26, 27] are used to replicate (or approximate) the behavior of the simulation in the best possible way, but with much less computational cost. This method aims at generating a model surface based on a limited number of simulator responses.

2. Problem Definition

Within the context of aircraft design optimization methodologies and computational codes, one of the most important aspects is obviously to compute accurate aerodynamic coefficients for the lifting surfaces for any given flight condition at the lowest possible cost. In low fidelity analyses, the high cost of computational fluid dynamics makes this choice poorly competitive with respect to simpler methods: the most common solutions being the Lifting Line Theory (LLT), the Vortex Lattice Method (VLM), and the 3D Panel Method. Nonlinear formulations of LLT and VLM require knowledge of the airfoil aerodynamic coefficient curves at various sections across the span.

For using such methods, it is fundamental to have good estimates for the airfoil lift, drag, and pitching moment coefficients (C_l, C_d, and C_m). Particularly, in low Reynolds number flows, it turns out to be absolutely crucial to have a high sampling rate of this coefficients as the optimizations are run because every time there is an airspeed change, the respective impact on the aerodynamic coefficients not only is hard to forecast but also exhibits nonlinear behavior in some situations. If the aerodynamic coefficients are not accurate enough, all the aircraft optimization codes will be mistakenly biased which will result in poor optimization results.

In a standard airfoil, it is right to consider that its aerodynamic coefficients are functions of the Reynolds number (Re) and the angle-of-attack (α). Provided there is a fair number of Re and α combinations for which the aerodynamic coefficients are known, it is straightforward and not costly to implement a multilinear interpolation strategy which does not impact performance while providing good results by using pairs of consecutive data points. In the scope of this paper, this approach, which approximates a multivariable continuous function given by a discrete number of data points by a product of linear functions, is called piecewise linear multivariable interpolation [28].

However, if the goal is the assessment of a continuous morphing flap solution, the same approach could become more costly as the number of variables would obviously increase. The challenge is thus to develop an approximate method, using multiquadric functions, which enable a fast and accurate computation of the aforementioned aerodynamic coefficients as a direct function of Reynolds number (Re), angle-of-attack (α), flap chord ratio (c_{flap}), and flap deflection (δ_{flap}) and then compare its accuracy and computation time with the multivariable interpolation approach and with direct analysis by XFOIL [29]. For a given set of airfoil operational low-speed conditions, which may comprise airspeed, altitude, air turbulence level, and surface roughness, those parameters were considered the most relevant of all affecting the aerodynamic coefficients of a given fixed airfoil [30].

3. Multiquadric Approximation

3.1. Method Formulation. Multiquadric surfaces can be represented generically by the series:

$$\sum_{i=1}^{n} c_i \left[q\left(x_i, y_i, x, y\right) \right] = z. \tag{1}$$

In this equation, z is a function of the input variables x and y, resulting from the sum of all the quadric surfaces q involved. The vertical axis of symmetry of each quadric surface is located at a specific point called the center point (or just center) of generic coordinates (x_i, y_i), which are associated with an unknown coefficient of the function c_i, which must be calculated in such a way as to describe the shape and the algebraic sign of the given surface.

There are certain particular examples of multiquadric surfaces, including the summation of a series of circular hyperboloids in two sheets, probably the most common, which is mathematically represented by

$$\sum_{i=1}^{n} c_i \left[\left(x_i - x\right)^2 + \left(y_i - y\right)^2 + \sigma^2 \right]^{(1/2)} = z. \tag{2}$$

Another typical example is a multiquadric surface defined by the summation of a series of circular paraboloids, which is mathematically represented by

$$\sum_{i=1}^{n} \left[c_i \left(x_i - x\right)^2 + \left(y_i - y\right)^2 + \sigma^2 \right] = z, \tag{3}$$

where the parameter σ represents the form factor of the specific quadric surface, a constant defined by the user and with great impact on the accuracy of the approximation.

After applying equation (2), one can observe in Figure 1 a representation of a quadric surface centered at point $(x, y) = (5, 5)$, in a domain with $x, y \in [0, 10]$ and with the form factor equal to zero and one, respectively. When the form factor is equal to zero, a conical surface is generated, whereas with a form factor greater than zero, the surface softens, and its center region exhibits the shape of a curve (with a continuous derivative function) instead of a vertex.

The multiquadric approximation function h as a function of only one vector x, in the xz-plane, is presented next to more succinctly illustrate the solution method. Later, the method is extended to more variables, since this does not modify the methodology, but only modifies the size of the problem. Thus, the multiquadric approximation function is defined by

$$\widetilde{h}(x) = \sum_{i=1}^{n} c_i \cdot \varnothing_i(x) + c_0. \tag{4}$$

The sum consists of $(n + 1)$ elements, associated to the n centers used to generate the approximation function, which also corresponds to $(n + 1)$ coefficients c_i that must be calculated according to the sample data to be approximated. In this notation, $\varnothing_i(x)$ is a multiquadric function and is defined as

$$\varnothing_i(x) = \sqrt{\sigma^2 + \left\| x - x_i \right\|^2}. \tag{5}$$

After calculating this parameter, for the selected centers and form factor, the coefficients c_i are obtained. This is the true challenge behind the multiquadric approximation, whose main objective is to minimize the difference between the approximation value (h) and the real value (h):

$$f(c) = \sum_{i=1}^{n} \left(\widetilde{h}^{(i)} - h^{(i)} \right)^2, \tag{6}$$

where c is the vector of the function coefficients. In matrix form, the problem can be written as

$$\begin{bmatrix} 1 & \varnothing_1\left(x^{(1)}\right) & \dots & \varnothing_n\left(x^{(1)}\right) \\ 1 & \varnothing_1\left(x^{(2)}\right) & \dots & \varnothing_n\left(x^{(2)}\right) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \varnothing_1\left(x^{(m)}\right) & \dots & \varnothing_n\left(x^{(m)}\right) \end{bmatrix} \cdot \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} h^{(1)} \\ h^{(2)} \\ \vdots \\ h^{(m)} \end{bmatrix}, \tag{7}$$

or, alternatively, in a more condensed notation,

$$[M]_{(m,n+1)} \times [c]_{(n+1,1)} = [h]_{(m,1)}. \tag{8}$$

The interpolation matrix M has m rows, associated with the m known sampling points, so m is the total number of points. It also has $(n + 1)$ columns, the first one being totally filled by ones, where n represents the total number of centers used to represent the function. Vector c represents the $(n + 1)$ unknown coefficients of the function, which will be calculated using the vector of the absolute terms, denoted by vector h of dimension m. Therefore, $\widetilde{h}(x)$ and $h(x)$ are, respectively, the approximation function and the actual function (or set of discrete data) which are required to be approximated, respectively.

If matrix M were square (with equal number of rows and columns), the calculation of the function coefficients would only require a simple matrix inversion; however, for this situation, it is necessary to find the values of c_i that minimize the following function:

$$J(c) = \frac{1}{2} \left\| [M][c] - [h] \right\|^2, \tag{9}$$

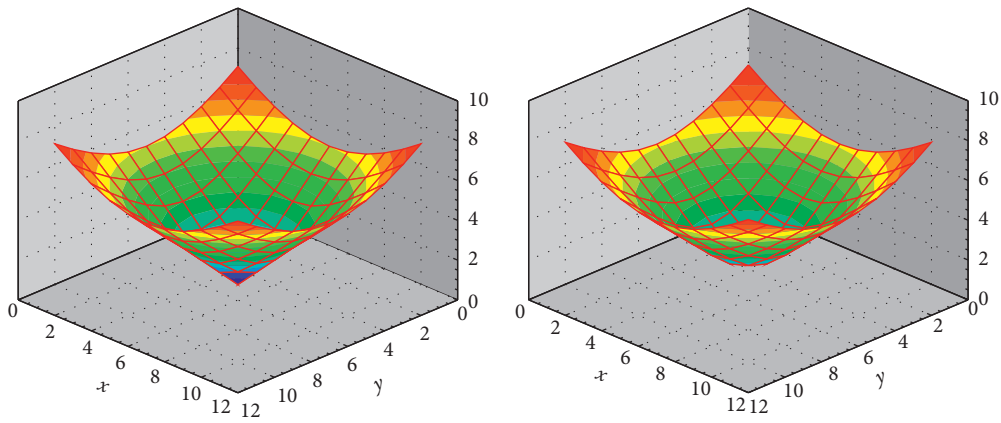

Figure 1: Multiquadric surface representation. (a) Form factor = 0. (b) Form factor = 1.

where $\|\cdot\|$ represents the Euclidean norm. The solution of the previous equation is obtained through the pseudoinverse of the matrix M as follows:

$$[c] = \left([M]^T [M]\right)^{-1} [M]^T [h], \tag{10}$$

where $[c]$ is the vector c in matrix representation.

It can be shown that it is possible to compute the approximation function $\widetilde{h}(x)$ in equation (4), once matrix M and the function coefficients associated with the selected centers are calculated. If two independent variables x and y are used in the data, equation (4) is slightly modified converting the hyperboles into circular hyperboloids with the following mathematical formulation:

$$\widetilde{h}(x, y) = \sum_{i=1}^{n} c_i \cdot \sqrt{\sigma^2 + (x - x_1)^2 + (y - y_1)^2} + c_0. \tag{11}$$

However, in order to obtain a good approximation, it is necessary to understand the influence of the placement of the centers and the form factor that composes the function. Depending on the user, the concept of good approximation may have different meanings, whereas some applications emphasize the speed of the function generation (and low processing time to get the approximation values), and others focus more on accuracy in order to obtain an interpolation error below a certain value. In this way, it is necessary to understand the effects behind the parameters that make up the approximation function so that they can be adjusted according to specific needs and objectives.

3.2. Center Points' Placement. It is neither practical nor realistic to manually perform the distribution of the centers over the domain under study, especially when the number of sampling points or the number of independent variables increases. Therefore, a method that automatically calculates the centers regardless of the domain and/or number of variables is needed. It is known from the outset that the greater the number of centers used, the more complex the multiquadric function becomes and, consequently, the greater the computational effort required to generate the function. As such, the user must stipulate a compromise between having greater accuracy (and greater computation time) or faster processing time (with less computational effort and with the subsequent decreased accuracy).

If the function is defined in a two-dimensional space, or in a higher dimension, it is not enough to identify the number of centers that establish the function, since these can be distributed in countless ways along the domain. Therefore, whenever it is necessary to identify the number of centers applied, taking for example a two-dimensional space, the number of centers will be selected in a line parallel to the x-axis and y-axis so that the total number of centers making up the multiquadric function will always be the product of both values, since the automatic method of collocation guarantees symmetry in its placement. In a two-dimensional space, the variables used follow the notation (x, y) for any set of discrete data or Re and α when applied to the analysis of wing airfoils, whereas with four independent variables, the notation will change to (x, y, z, t) or (Re, α, c_{flap}, δ_{flap}) when applied to flapped wing airfoils. Variables x, y, z, and t are associated with Re, α (in degrees), and percentage of c_{flap} and δ_{flap} (in degrees), respectively.

In order to denote the number of centers that make up the function, the following notation will be used in the two-dimensional space, taking as an example C(5,4) when placing five centers in a line parallel to the x-axis and four centers in a line parallel to the y-axis, shown in Figure 2(b) (using the automatic collocation method). With respect to the number of sample points used to approximate the values of the aerodynamic coefficients of the wing airfoils, the notation P(21,21) is used as shown in Figure 2(a). Thus, one can describe exactly how the centers and sample points are distributed throughout the domain. Note that the cases under study the known points associated with the aerodynamic coefficients are also symmetrically distributed (which allows the use of the notation P(x,y)), although it is possible to have the known points in any distribution within the domain defined for the purpose. However, since the

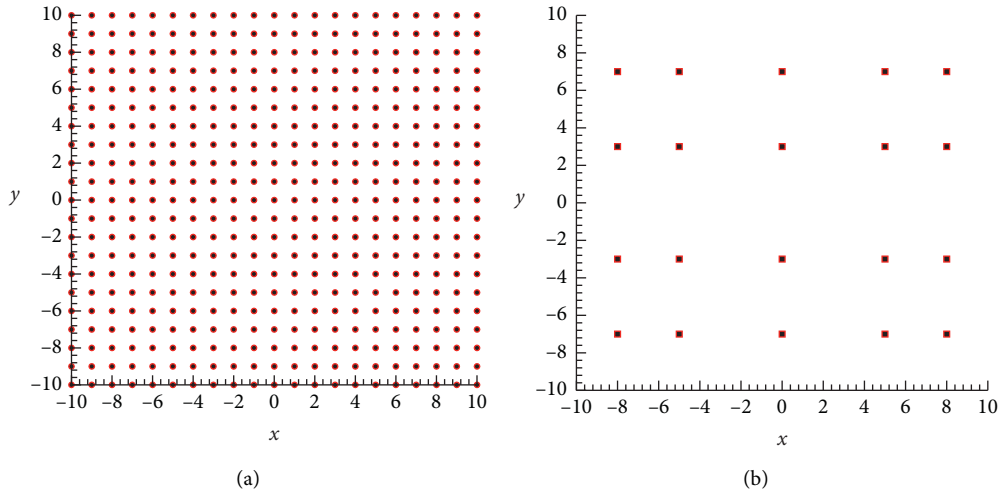

Figure 2: Example of the automatic collocation of center points throughout the domain. (a) Sample points. (b) Automatic center collocation method.

aerodynamic coefficients are obtained from the XFOIL program and because of program interface limitations, the sample data will always be symmetrically distributed.

In one of his more recent works, Hardy states that the best and simplest solution is to place the centers exactly in the same positions where the sample points are, since each center point generates a focus of inclination in that position, which is not beneficial to occur in arbitrary domain positions [4]. However, placing a center at each known data point increases the processing time required, since the interpolation matrices take on a larger dimension, so this alternative was adapted, and a new collocation method was implemented. This method places the points symmetrically along the domain assuring that each center is also a known point of the function, a strategy advised by several authors. This strategy, in addition to placing centers symmetrically distributed throughout the domain and at known data points, distributes the centers as uniformly as possible so that one can avoid having large ranges of points without any centers compared to other areas of the domain. When the number of centers is also equal to the number of points, the upper limiting case for the number of used centers means that each known point will also be a center of the function.

As an example, a random two-dimensional domain, with the sample points $x, y \in [-10, 10]$ and with a unit step, which makes up for a total of 441 sample points, or in the notation P(21,21), as shown in Figure 2(a), which shows all available points to generate the approximation function is defined. The automatic distribution of the centers applying the implemented automatic placement method is also shown in Figure 2(b), with five centers in a line parallel to the x-axis and four centers in a line parallel to the y-axis, which makes a total of 20 centers, or in the notation C(5,4). An increase in the number of centers may not always benefit the approximation function, since increasing its quantity also means changing, in most cases, the location of all other centers in order to readjust them in a new configuration. Thus, by changing the location of the centers, the method can sometimes place them in less appropriate places (depending on the sample set), increasing its quantity and thus slightly worsening the function interpolation error. It is, however, desirable to adopt this configuration alternatively to one in which the addition of centers does not adjust the location of the previous centers because in this way better results can be obtained with low amounts of centers, since these will always be almost uniformly distributed over the domain and each readjustment of the number of centers will also readjust all the configuration adopted in its placement.

3.3. Form Factor Calculation Using FFSQP. The idea of implementing this optimization method came as a corollary due to the typical evolution of the curves of the average relative error versus the form factor, in order to find, in an automated way, without user intervention, the best possible form factor for the combination of predefined centers, that is, the smallest percentage average relative error (or abbreviated by REL.P) is obtained. This method aims to be applied to convex functions to find the value for which its derivative is (approximately) zero in the vicinity of the initial point under study. Despite not being possible to guarantee that this point is in fact an absolute maximum or minimum of the function, it will always be at least a local maximum or minimum in the study region. Considering the case study, a function minimization is required.

FFSQP 3.7 (*FORTRAN Feasible Sequential Quadratic Programming-Version 3.7*) is a series of subroutines that aim to minimize the solution of one or more defined objective functions (with the possibility of none at all) subject to predefined constraints. It is up to the user to provide an initial estimate of the solution from which to start the application of the method, preferably in the vicinity where the method will predictably find a plausible solution compliant with the restrictions imposed and that minimizes the objective. If this initial estimate does not respect the constraints imposed, a plausible estimate is generated from which the

iterative process starts. It is also up to the user to define the subroutines encompassing the objective function(s) and the constraint(s) applied to the problem. This process solves differential equations based on the approximation of derivatives through forward finite differences, having its formulation based on the Taylor series of the derivative function. This optimization method calculates the derivatives of a given objective function (which in this case is the REL.P as a function of the form factor) and autonomously determines the point at which the derivative is (approximately) zero, which corresponds to a local minimum of the function [31].

In the course of the study, the parameters shown in Table 1 were applied in order to guarantee, as much as possible, a fast and accurate approximation. These parameters have been subjected to several sensitivity tests to understand its influence.

There are mainly two parameters that influence the value of convergence of the method: the step size and the initial value where the process starts. As for the step size, it affects not only the speed of convergence but also has an impact on the quality of the solution. Having a high step size leads to a larger iteration step, which may allow the desired solution to be reached more quickly, but it can also be a factor that leads the solution to zones of unstable solution, exceeding the desired local minimum. The initial form factor can also have an impact on the solution since one may already be in the vicinity of another local minimum (when the evolution of the function is nonmonotonic), which may be beneficial or harmful, depending on the convex shape acquired by the defined objective function. It was decided to define the step size equal to 0.1E-6 and a form factor whose initial value equals zero.

Another important aspect that mainly affects the convergence time of the method is the required accuracy of the solution, since the greater the precision, the greater the number of iterations required for the method to converge. Therefore, an accuracy of 0.001% was adopted for the mean relative error. By reducing accuracy, it is possible to achieve similar (but less accurate) solutions with lower convergence time; thus, it is up to the user to adjust it to each problem.

3.4. Normalization of the Problem. The implemented algorithm is made to work regardless of the dataset. This includes datasets with different dimensions, number of points, and variables' magnitude. Therefore, a normalization to all points belonging to the data set in a range of [−1,1] for all input variables is applied, to be able to always have a relative perception of the zone of the domain that the user is interested in, without investigating the limits of the domain of the variable, always having the guarantee that it is within the defined limits. This approach aims to standardize the datasets, regardless of their size. It is known that the value of the form factor is naturally related to the set of sample points and to the absolute distance between them, so this approach also tries to standardize, as far as possible, the range of form factor values that enables a good approximation to the function. Since typically the computational methods have difficulty in operating with values of great magnitude, it is also intended with this procedure to reduce the absolute value of the large scales.

Table 1: Parameters applied to the FFSQP 3.7 method.

FFSQP 3.7 (FORTRAN feasible sequential quadratic programming-Version 3.7)	
Step size	0.1E-6
Required precision (%)	0.001
Initial form factor	0
Maximum number of iterations	100
Estimating function gradients by	Forward finite differences

In the same way that the data points are normalized, the particular centers along the domain will also be normalized, in order to keep the solution consistent. In Figure 3(b), the centers automatically placed before normalization for a domain with P(19,7), where $x \in [0, 18]$ and $y \in [0, 6]$, are represented, with a uniform step between any two consecutive points equal to one for both variables. Figure 3(c) features the distribution of the centers after their normalization with C(11,5). The dataset used before normalization is represented in Figure 3(a). It was decided to apply the representation in a two-dimensional space since it facilitates the understanding of the problem, although this method is extended to any dimensional space.

4. Piecewise Linear Multivariable Interpolation

A more conventional approach to estimate the value of a function between two known points is the linear interpolation method. If the function under analysis is linear, then the linearly interpolated value will be its exact value. If the array has more than two dimensions (multivariable), the value sought will be at a point within the interior of the corresponding polytope, with the number of mathematical operations to be performed depending on the number of problem dimensions. The quality of the interpolated solution depends not only on the number of scattered points for which the function is known but also on the function behavior. The more linear behavior it exhibits, the better will be the fitting between the actual function and the interpolated values.

Before generalizing the multilinear interpolation approach, let us look at the simplest case of a linear interpolation. Let the superscript (i) denote the variable dimension and subscript (j) denote the point's reference number, (x_j^i). Figure 4 depicts a typical scheme with two points, for which the function is known $(f(x_0^1))$ and $(f(x_1^1))$, and the point in between these two, for which the linear interpolation is required $(f(x_2^1))$.

Equation (12) shows the linear interpolation used for computing $f^C(x_2^1)$:

$$f^c\left(x_2^1\right) = f\left(x_1^1\right) + \frac{\left[f\left(x_1^1\right) - f\left(x_0^1\right)\right]\left(x_2^1 - x_0^1\right)}{\left(x_1^1 - x_0^1\right)}. \quad (12)$$

If the goal were to obtain a bilinear interpolation, the problem to solve would be something like the scheme shown

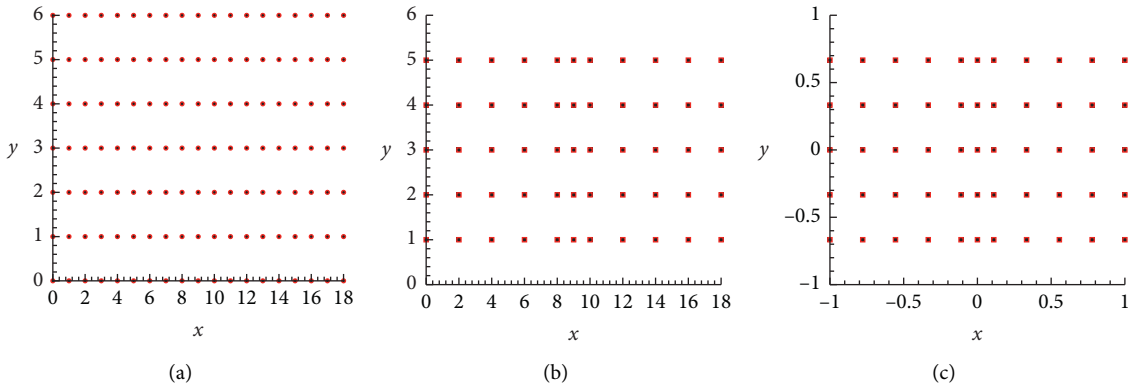

Figure 3: Example of normalization. (a) Dataset P(19,7), (b) C(11,5) before normalization, and (c) C(11,5) after normalization.

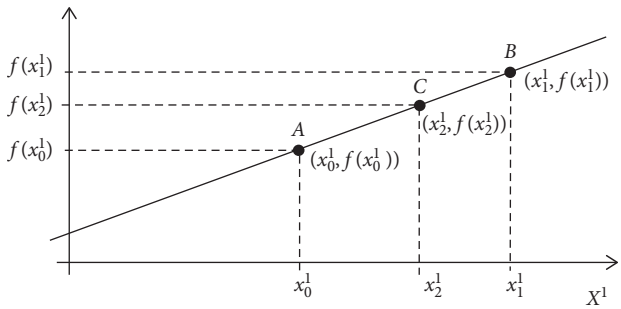

Figure 4: Linear interpolation.

in Figure 5. The function value is known at the four points A, B, C, and D, $(f(x_0^1, x_1^2))$, $(f(x_1^1, x_1^2))$, $(f(x_0^1, x_0^2))$, and $(f(x_1^1, x_0^2))$, respectively, and the bilinear interpolation goal is to obtain an approximate value for the same function at point E, $(f(x_2^1, x_2^2))$.

Contrary to the approach adopted for the linear interpolation, the bilinear interpolation is divided into two steps, each of which representing a linear interpolation. The first step consists of computing the function value at points E_0 and E_1, each being obtained via a linear interpolation:

$$f^{E_0}\left(x_0^1, x_2^2\right) = f^C\left(x_0^1, x_0^2\right) + \frac{\left[f^A\left(x_0^1, x_1^2\right) - f^C\left(x_0^1, x_0^2\right)\right]\left(x_2^2 - x_0^2\right)}{\left(x_1^2 - x_0^2\right)},$$

$$f^{E_1}\left(x_1^1, x_2^2\right) = f^D\left(x_1^1, x_0^2\right) + \frac{\left[f^B\left(x_1^1, x_1^2\right) - f^D\left(x_1^1, x_0^2\right)\right]\left(x_2^2 - x_0^2\right)}{\left(x_1^2 - x_0^2\right)}. \tag{13}$$

The second step is to use these two new points to estimate the function value at point E, in the same way as it is done in the linear interpolation:

$$f^E\left(x_2^1, x_2^2\right) = f^{E_0}\left(x_0^1, x_2^2\right) + \frac{\left[f^{E1}\left(x_1^1, x_2^2\right) - f^{E_0}\left(x_0^1, x_2^2\right)\right]\left(x_2^1 - x_0^1\right)}{\left(x_1^1 - x_0^1\right)}. \tag{14}$$

As the number of variables increases, the graphical representation becomes harder to understand and it is therefore important to generalize what has been stated for linear and bilinear interpolation to multivariable interpolation, $f(x_2^1, x_2^2, x_2^3, \ldots, x_2^n)$.

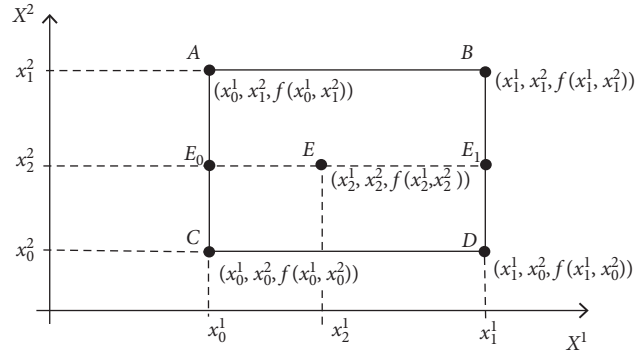

Figure 5: Bilinear interpolation.

The first step is to determine the coordinates of the known points which are closer to the point $(x_2^1, x_2^2, x_2^3, \ldots, x_2^n)$. These points define the vertices of a multidimensional polytope surrounding the point at which the function aims to be interpolated:

$$\begin{cases} \left(x_0^1, x_0^2, \ldots, x_0^n\right) \\ \left(x_0^1, x_0^2, \ldots, x_1^n\right) \\ \left(x_0^1, x_1^2, \ldots, x_1^n\right) \\ \left(x_1^1, x_1^2, \ldots, x_1^n\right) \end{cases} \tag{15}$$

Following a similar approach to the one presented for the linear and bilinear interpolation and beginning by interpolating variable x_2^1,

$$\begin{cases} f\left(x_0^1, x_0^2, \ldots, x_0^n\right) = f\left(x_0^1, x_0^2, \ldots, x_0^n\right) + \left[f\left(x_0^1, x_0^2, \ldots, x_0^n\right) - f\left(x_0^1, x_0^2, \ldots, x_0^n\right)\right] \dfrac{\left(x_2^1 - x_0^1\right)}{\left(x_1^1 - x_0^1\right)} \\ f\left(x_2^1, x_0^2, \ldots, x_1^n\right) = f\left(x_0^1, x_0^2, \ldots, x_1^n\right) + \left[f\left(x_1^1, x_0^2, \ldots, x_1^n\right) - f\left(x_0^1, x_0^2, \ldots, x_1^n\right)\right] \dfrac{\left(x_2^1 - x_0^1\right)}{\left(x_1^1 - x_0^1\right)} \\ f\left(x_2^1, x_1^2, \ldots, x_0^n\right) = f\left(x_0^1, x_1^2, \ldots, x_0^n\right) + \left[f\left(x_1^1, x_1^2, \ldots, x_0^n\right) - f\left(x_0^1, x_1^2, \ldots, x_0^n\right)\right] \dfrac{\left(x_2^1 - x_0^1\right)}{\left(x_1^1 - x_0^1\right)} \\ f\left(x_2^1, x_1^2, \ldots, x_1^n\right) = f\left(x_0^1, x_1^2, \ldots, x_1^n\right) + \left[f\left(x_1^1, x_1^2, \ldots, x_1^n\right) - f\left(x_0^1, x_1^2, \ldots, x_1^n\right)\right] \dfrac{\left(x_2^1 - x_0^1\right)}{\left(x_1^1 - x_0^1\right)} \end{cases} \tag{16}$$

Interpolating variable x_2^2,

$$\begin{cases} f\left(x_2^1, x_2^2, \ldots, x_0^n\right) = f\left(x_0^1, x_0^2, \ldots, x_1^n\right) + \left[f\left(x_1^1, x_0^2, \ldots, x_0^n\right) - f\left(x_0^1, x_0^2, \ldots, x_0^n\right)\right] \dfrac{\left(x_2^2 - x_0^2\right)}{\left(x_1^2 - x_0^2\right)} \\ f\left(x_2^1, x_2^2, \ldots, x_1^n\right) = f\left(x_2^1, x_0^2, \ldots, x_1^n\right) + \left[f\left(x_2^1, x_1^2, \ldots, x_1^n\right) - f\left(x_2^1, x_1^2, \ldots, x_1^n\right)\right] \dfrac{\left(x_2^2 - x_0^2\right)}{\left(x_1^2 - x_0^2\right)} \end{cases} \tag{17}$$

Interpolating variable x_2^n,

$$f\left(x_2^1, x_2^2, \ldots, x_0^n\right) = f\left(x_2^1, x_2^2, \ldots, x_1^n\right) + \left[f\left(x_2^1, x_2^2, \ldots, x_1^n\right) - f\left(x_2^1, x_2^2, \ldots, x_0^n\right)\right] \frac{\left(x_2^n - x_0^n\right)}{\left(x_1^n - x_0^n\right)}. \tag{18}$$

The number of vertices of the aforementioned multidimensional polytope is equal to (2^n), with (n) being the number of problem dimensions, and the number of linear interpolations to be performed is thus equal to $(\sum_{i=1}^{n} 2^i - 1)$.

This methodology can be applied to a set of data, representing a multidimensional non-linear function, by fitting linear functions inside the polytope made up by consecutive data points. The result is a nonlinear global representation of the data by local piecewise linear multivariable interpolation.

5. Results

5.1. Two Independent Variables. This section begins by approximating the aerodynamic coefficients of airfoils as a function of Re and α through multiquadric functions. Four different approaches are applied to analyze the behavior and applicability of the available approximation functions in Sections 5.1.2–5.1.5.

5.1.1. Airfoils and Domain Used Throughout the Study. Two different airfoils: UBI_03_016 and DAE-21, represented in Figures 6 and 7, respectively, are used throughout the study to approximate their aerodynamic coefficients (lift, drag, and pitching moment). The data points applied were obtained from XFOIL: P5,51 for each dataset associated with any given aerodynamic coefficient. A set of 255 sample points are used to define each function that represents a particular aerodynamic coefficient with $\mathrm{Re} \in [75000, 675000]$ and $\alpha \in [-5, 20]$, which results in a step of 150,000 and 1 degree, respectively. Since the goal is to define the coefficients through a multiquadric function, Re and α are treated as independent variables associated with

Figure 6: UBI-03_106 airfoil.

Figure 7: DAE-21 airfoil.

the function, while the associated coefficient is the output value.

5.1.2. Influence of the Number of Centers and of the Form Factor. To understand the behavior of the multiquadric approximations with increasing number of centers, the variation of the REL.P as a function of the number of centers is represented for a constant form factor equal to zero, as shown in Figures 8(a)–8(c) for the UBI_03_016 airfoil and Figures 8(d)–8(f) for the DAE-21 airfoil.

The first important aspect to highlight is the greater difficulty in approximating the data corresponding to the drag coefficient, a trend that is not limited to these airfoils. Irregularity and nonstandard behavior mean that more centers are required to allow reasonable approximations with multiquadric functions. Considering the other two aerodynamic coefficients, it is possible to note some similarities in the approach behavior depending on the number of centers applied when comparing the two wing airfoils. It should be noted that the goal is, however, to optimize the value of the form factor as far as possible to minimize the associated errors.

To have a perception of the value of the form factor from which an unstable behavior occurs in the solution, resulting from the ill-conditioning associated with the interpolation matrix, the evolution of the REL.P with a growing form factor is represented in Figure 9 for all possible centers combinations of the UBI_03_016 airfoil lift coefficient.

5.1.3. Form Factor Optimization Using FFSQP 3.7. To demonstrate the benefits of applying this optimization method, a random set of centers' combinations, associated with the multiquadric function that approximates the lift coefficient (since the trend is the same for the other aerodynamic coefficients), is selected, and then, the method is applied to find the form factor that minimizes the percentage relative error. The results obtained are shown in Tables 2–4. This method always converges to the local minimum of the objective function defined in the vicinity of the chosen starting point as the form factor is equal to zero. Since this approach requires many iterations to ensure convergence of the optimized form factor, the time required to fully generate the function, with all associated function coefficients calculated and saved (or generation time with abbreviation GT), to be used to obtain approximate solutions of the aerodynamic coefficients is presented. As a way of comparison, the solution obtained by maintaining a form factor equal to zero for all existing centers is also presented.

To verify the benefits of the application of the FFSQP method, Figure 10 presents the evolution of the REL.P versus the form factor for the two airfoils under analysis and for all center combinations. Note that the observed unstable zone occurs due to poor conditioning of the interpolation matrices, a situation already detected by Hardy and many other researchers when increasing the form factor too much. The more centers applied the sooner instability begins, that is, for smaller form factors.

The FFSQP method makes it possible to achieve the local minimum of the function with the cost of increasing the required processing time, since several iterations are required until the method converges to the desired final solution. It is up to the user to assess the benefits of having greater precision at the cost of increased generation time. Note that once the function is generated it can provide infinite approximate solutions of the aerodynamic coefficients, depending on the input data entered, since the function needs to be generated only once.

5.1.4. Accuracy of the Approximation Methods. Since the problem has a finite set of data points dispersed over a predefined domain, there is no continuity between any two consecutive points, so it is up to the applied approximation function to estimate its shape. In most situations, it is not enough to obtain a low interpolation error, that is, a low error in the known data points, and it is also necessary to have a realistic evolution between any two consecutive points. To assess the quality of the approximation resulting from the dataset, the concentration of the data points increased within the previously defined domain, again using XFOIL to get P(13,251), or 3263 sampling points, with a uniform step of 50000 and 0.1 for Re and α, respectively.

By knowing the exact value (in some locations) of the neighboring data points used to generate the functions, it is possible to compare those with the values obtained using the approximations methods and find out whether the methods provide or not a good approximation. In Figure 11, the location of the 3263 data points (inserted inside the domain under study) is represented in yellow, while the location of the data points used to define the multiquadric functions and multivariable interpolation is represented in black. This analysis refers to the DAE-21 airfoil shown in Figure 7.

While the set of known points has increased, the points to generate the functions have not. Therefore, it is required to change the adopted nomenclature. When analyzing the mean relative error for all points represented in yellow in Figure 11, parameter REL.P.A (percentage average relative error increased) is used instead, since REL.P only contemplates the data points used to generate the approximation functions.

Looking at the results obtained and only considering the multiquadric functions with form factors equal to 0, 0.1 and 0.15, it was observed that the methods of piecewise linear multivariable interpolation and multiquadric functions provided very similar approximations. While the piecewise

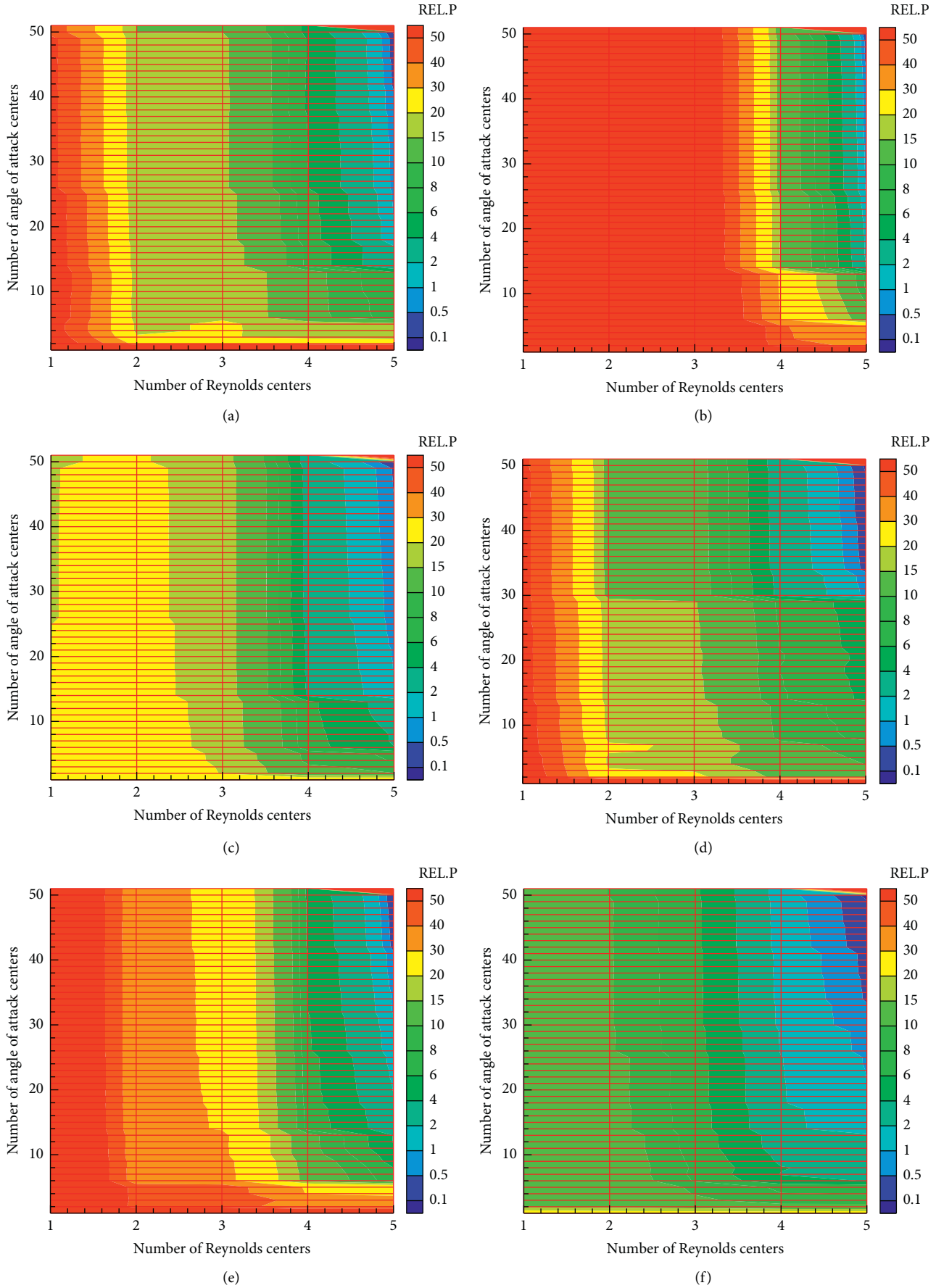

Figure 8: REL.P versus centers' combination C(Re,α). (a) Lift coefficient for UBI_03_016 airfoil, (b) drag coefficient for UBI_03_016 airfoil, (c) pitching moment coefficient for UBI_03_016 airfoil, (d) lift coefficient for DAE-21 airfoil, (e) drag coefficient for DAE-21 airfoil, and (f) pitching moment coefficient for DAE-21 airfoil.

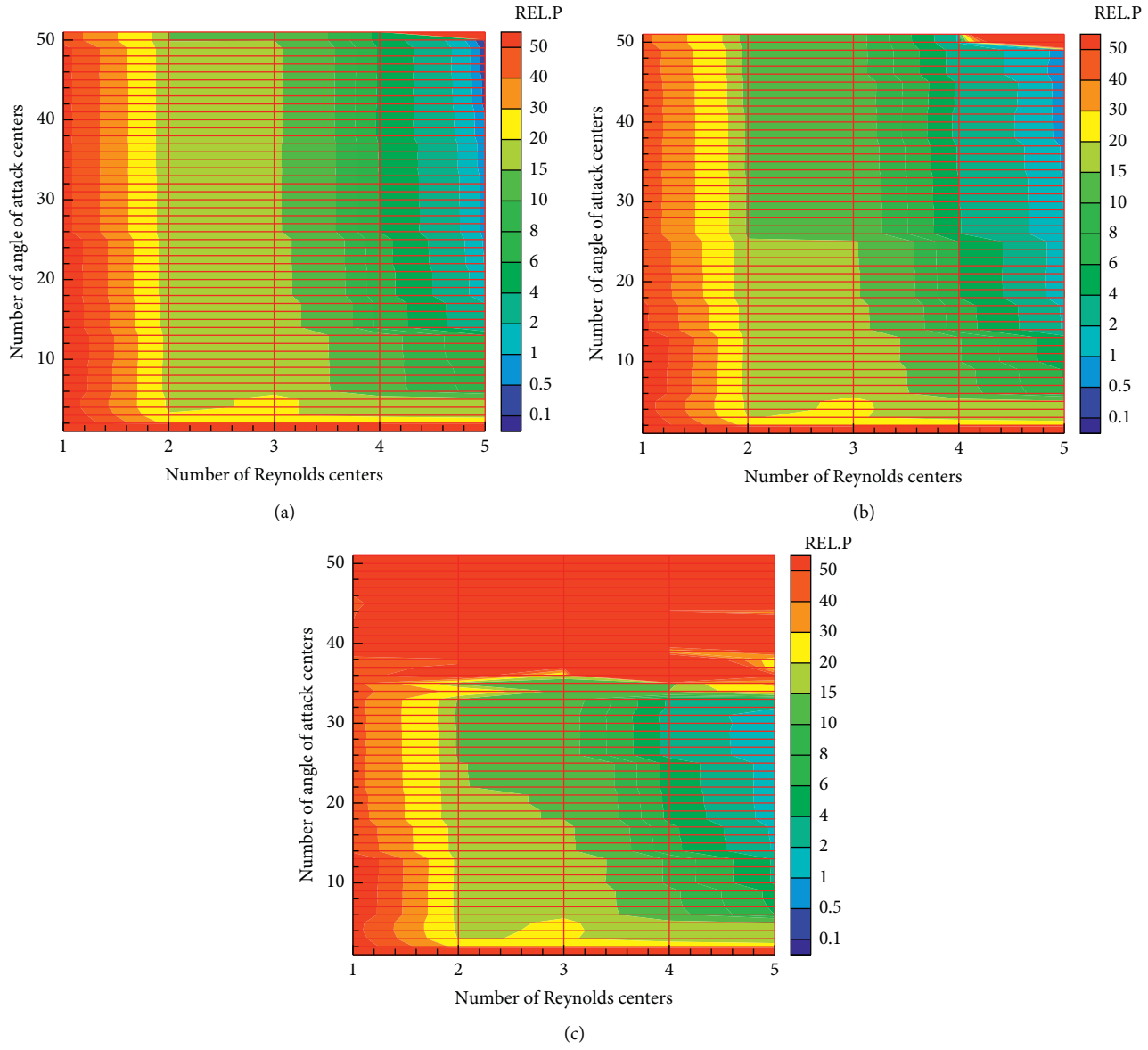

Figure 9: Form factor increase: REL.P versus centers combination C(Re,α) for UBI_03_016 airfoil. (a) Form factor = 0. (b) Form factor = 0.1. (c) Form factor = 0.15.

Table 2: Results obtained through the FFSQP method (1).

Centers' combination C(5,5)			
Airfoil		UBI_03_016	DAE-21
FFSQP 3.7	Form factor	1.45	0.28
	REL.P	4.40%	6.36%
	GT [s]	4.125	1.09
Form factor = 0	REL.P	16.51%	8.91%
	GT [s]	0.0156	0.0156

linear multivariable interpolation approach yielded a 4.9%, 16.1%, and 1.9% REL.P.A, the multiquadric functions generated minimum values around 5.7%, 15.5%, and 3.5% for the lift, drag, and pitching moment coefficients, respectively.

For certain multiquadric functions with certain centers combinations (keeping a constant form factor equal to zero), the absolute error (ABS) across the domain between the multiquadric and the multivariable interpolation functions was compared. In order to identify the regions of the domain

Table 3: Results obtained through the FFSQP method (2).

Centers' combination C(5,15)			
Airfoil		UBI_03_016	DAE-21
FFSQP 3.7	Form factor	0.185	0
	REL.P	1.823%	2.22%
	GT [s]	4.63	0.375
Form factor = 0	REL.P	2.04%	2.22%
	GT [s]	0.047	0.047

Table 4: Results obtained through the FFSQP method (3).

Centers' combination C(5,30)			
Airfoil		UBI_03_016	DAE-21
FFSQP 3.7	Form factor	0	0.007
	REL.P	0.73%	1.28%
	GT [s]	4.86	23.19
Form factor = 0	REL.P	0.73%	2.03%
	GT [s]	0.094	0.094

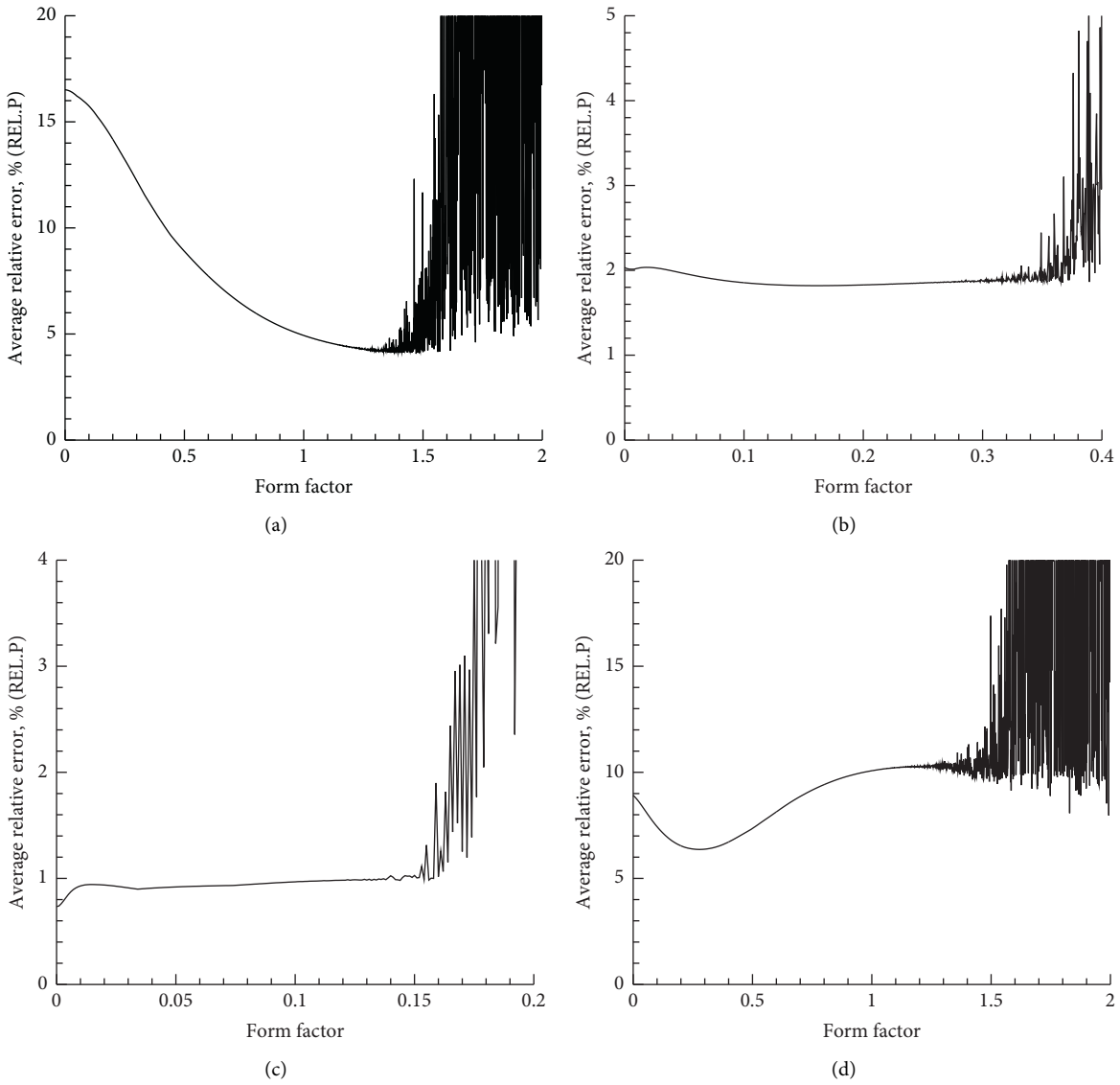

Figure 10: Continued.

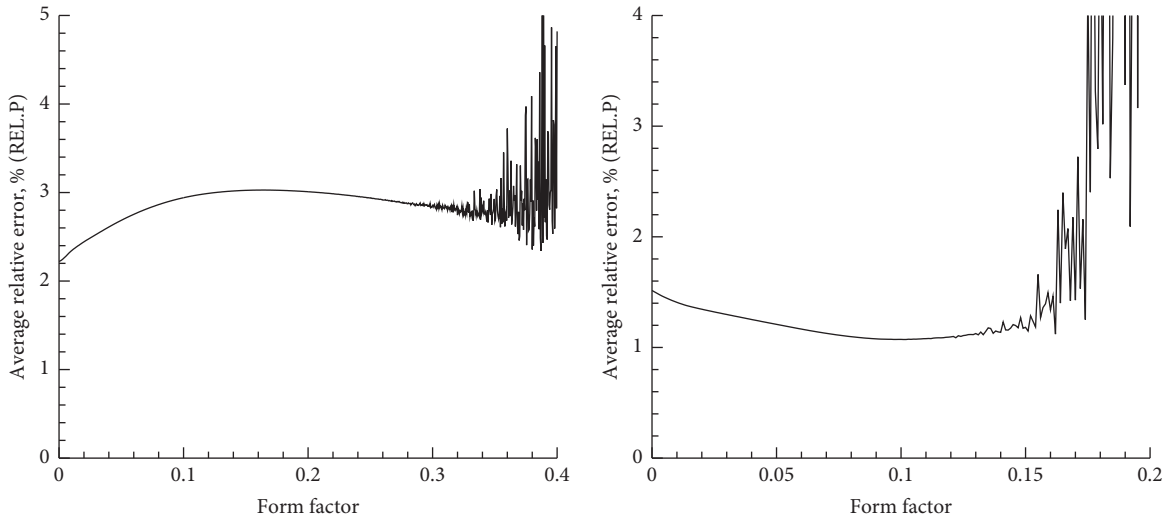

Figure 10: REL.P versus form factor. (a) UBI_03_016-C(5,5), (b) UBI_03_016-C(5,15), (c) UBI_03_016-C(5,30), (d) DAE-21-C(5,5), (e) DAE-21-C(5,15), and (f) DAE-21-C(5,30).

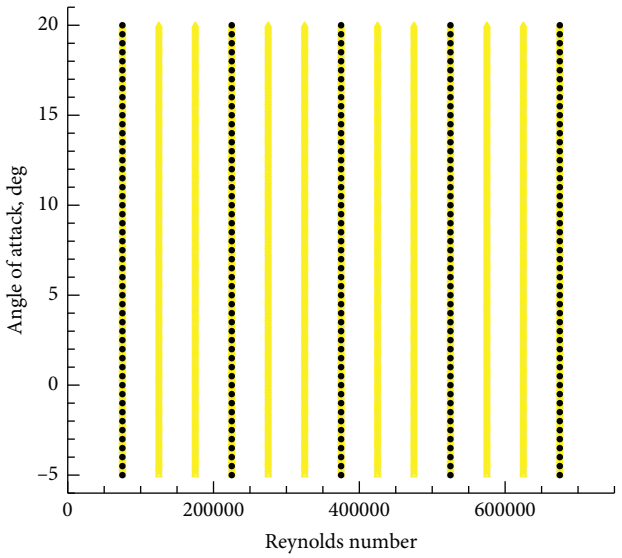

Figure 11: Representation of the known data points.

where one of the errors prevailed, the difference between the modulus of the absolute error of the multiquadric function and the modulus of the absolute error of the multivariable interpolation function (ABSD) was calculated, and the value obtained through the predefined domain is represented in Figure 12. When the value is positive (hot colors), it implies that the absolute error of the multiquadric function is higher, whereas when the value is negative (cold colors), it means that the absolute error of the piecewise linear multivariable interpolation function is now superior. Figures 12(a)–12(c) are related to the lift coefficient, while Figures 12(d)–12(f) correspond to the drag coefficient, and finally, Figures 12(g)–12(i) are associated with the pitching moment coefficient approximation functions.

There is a clear dominance of the piecewise linear multivariable interpolation function compared to the generated multiquadric functions, since the latter exhibits larger areas of reddish colors that reflect the larger approximation errors obtained with the multiquadric functions. The only zone of the domain where the multiquadric functions have some dominance corresponds to the range of low Re between [75000, 225000], where the need to increase the concentration of points to define the function was identified, since in this zone the largest absolute errors occur in both functions. The advantage of the multiquadric functions (regarding the accuracy obtained) compared to the piecewise linear multivariable interpolation is clear when there are few sample points in areas that typically require a larger point

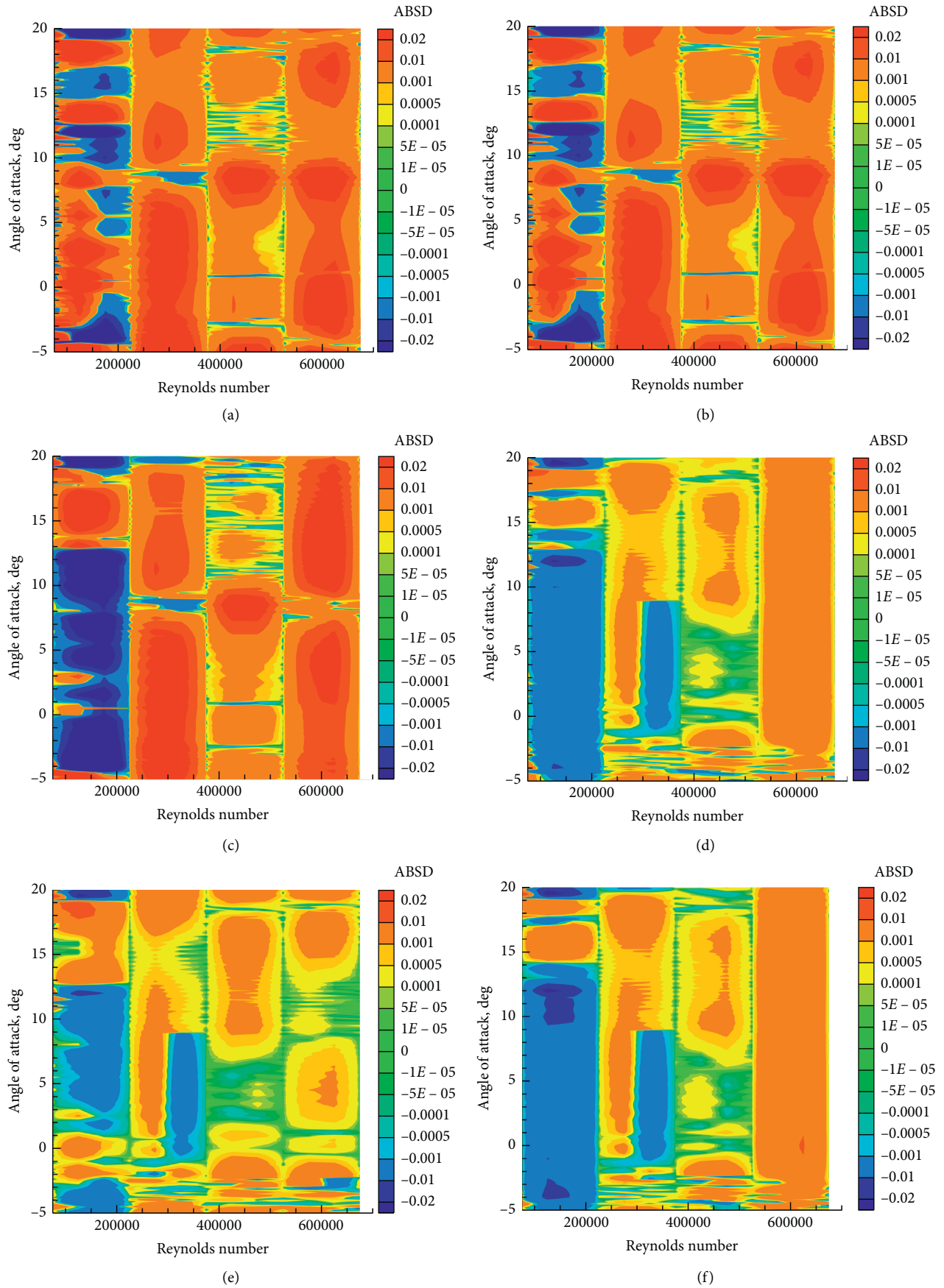

Figure 12: Continued.

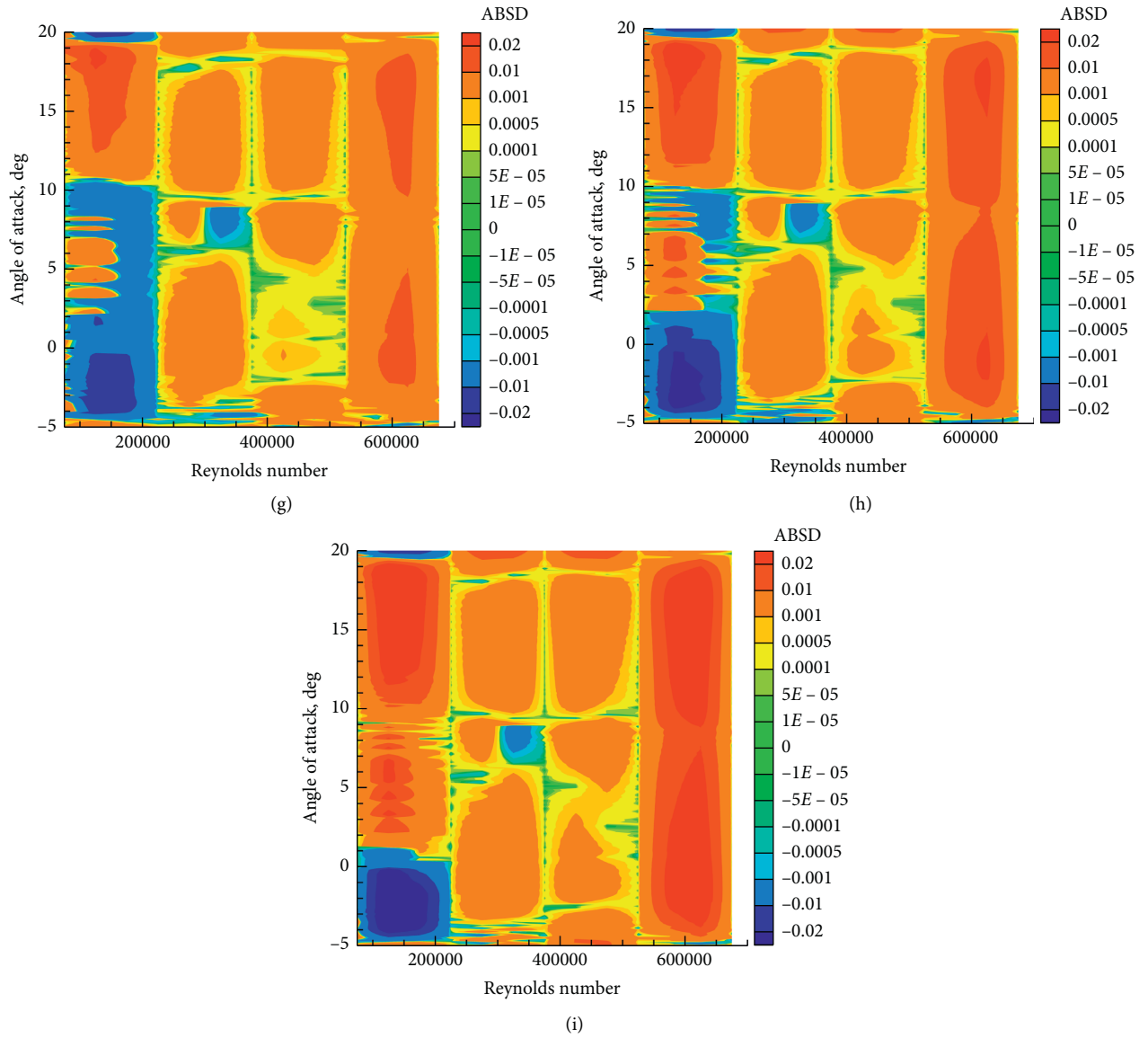

Figure 12: Difference between the ABS of the multiquadric function and the ABS of multivariable interpolation. (a) C_l-C(5,25), (b) C_l-C(5,35), (c) C_l-C(5,45), (d) C_d-C(5,25) (e) C_d-C(5,35), (f) C_d-C(5,45), (g) C_m-C(5,25), (h) C_m-C(5,35), and (i) C_m-C(5,45).

concentration. It is not clear that the increase in the number of centers used to define the multiquadric function improves the approximation along the domain, though they typically improve the approximation at the interpolation points used to define the function.

In order to verify if the results obtained in the previous analysis are in fact a characteristic of the representation of the aerodynamic coefficients of airfoils, the set of points used to define the multiquadric and the piecewise linear multivariable interpolation function (abbreviated by M.I.) was modified in Table 5 (as much as possible), while maintaining the previously defined limits for both Re and α. Taking as an example the situation where P(9,21) and knowing that Re $\in$ [75000 , 675000] and $\alpha \in [-5, 20]$, it is easy to obtain a step for Re and for α equal to 75000 and 1 degree, respectively. The dataset, which is enlarged and shown in yellow in Figure 11, will remain unchanged since these points will be used to establish the quality of the approximation of the functions outside the sample data points applied in the functions.

With the change of the data points used to define the multiquadric and multivariable interpolation functions, as shown in Table 5, a variation of the associated REL.P.A is notorious with the values of both functions always relatively close to each other, with differences of less than 1% (except once when P(5,51) and for the pitching moment coefficient). Thus, regarding the obtained precision, it is not possible to say which of the approximation functions is preferred.

5.1.5. Computation Time. Once the centers' combinations and form factor influence on the accuracy of the multiquadric functions has been analyzed, some combinations

Table 5: Changing the input data P(Re, α).

DAE-21 multiquadric functions versus piecewise linear multivariable interpolation									
	Lift coefficient			Drag coefficient			Pitching moment coefficient		
Data	Multiquadrics		M.I.	Multiquadrics		M.I.	Multiquadrics		M.I.
	C(Re,α)(%)	REL.P.A(%)	REL.P.A(%)	C(Re,α)	REL.P.A(%)	REL.P.A(%)	C(Re,α)	REL.P.A(%)	REL.P.A(%)
P(9,26)	C(9,15)	2.60	2.62	C(9,17)	11.88	11.72	C(9,25)	1.76	1.59
P(5,51)	C(5,44)	5.28	4.89	C(5,30)	15.71	16.07	C(5,13)	3.58	1.88
P(13,26)	C(13,16)	1.75	1.76	C(13,25)	10.19	10.10	C(13,25)	1.19	1.19
P(9,51)	C(9,33)	2.59	2.61	C(9,32)	11.09	11.19	C(9,42)	3.45	3.23

regarding their required processing time will be tested and compared with the piecewise linear multivariable interpolation methods and through a data collection directly from XFOIL.

First of all, it is necessary to define some parameters that aim to assist this analysis. The time for collecting and reading the data used to define the approximation functions will be abbreviated by CT and RT (in seconds), the generation time of the multiquadric functions is defined by GT (this step is not performed by the piecewise linear multivariable interpolation method), and finally the time required to actually obtain solutions of the aerodynamic coefficients (depending on the input data associated with Re and α) will be nominated ST. Regardless of the method applied to generate the multiquadric function, either through the FFSQP optimization method or by inserting a constant form factor at startup, such an approach will only be reflected in the GT. It should be noted that, in the ST phase of the multiquadric functions, only the normalization of the input data and the respective sum of the functions are performed in order to obtain the desired solution, according to the parameters saved or obtained previously during the GT. Regarding XFOIL direct calculation, since it does not need to perform any type of data collection and reading, it is expected that, for a small number of required solutions, this method is the most beneficial in terms of minimizing the time spent, although it is also predictable that it will slow down when several consecutive solutions are required for different input data.

To compare the computation time associated with the three approaches under analysis, the DAE-21 airfoil was selected, with the domain and the set of points previously used (P(5,51)) to define the functions, selecting three combinations of random centers and complemented with a form factor equal to zero, to define the multiquadric functions. The time required for the multiquadric functions, piecewise linear multivariable interpolation function and XFOIL, to perform N calculations in order to obtain N solutions within the domain of interest is calculated to determine which calculation approach is more efficient. Each calculation performed implies obtaining the three aerodynamic coefficients, for which three piecewise linear multiquadric functions and three multivariable interpolations are generated, each associated with a specific aerodynamic coefficient. The data collection and reading times (CT and RT) are performed only once at the beginning of the process for the two approximation methods, while the GT associated to the multiquadric functions is also performed only three times in the initial stage. In order to be able to compare these two approaches, the same number of calculations will always be performed and always with the same random input data. With regards to XFOIL, the time required to generate all N different solutions (with N different Re and always for the same predefined α because of interface limitations) was compared with the time used by the approximation functions above.

The use of XFOIL, besides not guaranteeing the convergence for certain situations, shows a very high calculation time when compared to the proposed alternatives. Although this method does not require initial time to collect and read data, it becomes time consuming for optimization problems where many function calculations are required. It was possible to verify that, for approximately 225 different calculations, the use of both the piecewise linear multivariable interpolation method and the multiquadric defined functions became beneficial when compared to the XFOIL approach.

In Table 6, the REL.P associated to each of the generated multiquadric functions is represented (firstly, for the multiquadric function associated with the lift coefficient, followed by the functions associated with the drag and pitching moment coefficients, respectively), as well as the time of collection of the data (CT) and the generation time of the multiquadric functions (GT). The reading time (RT) is omitted because it is very close to zero, and insignificant when compared to CT, while the calculation time (ST) is shown in Figure 13, for a consecutive set of random inputs. The GT corresponds to the time required to generate the three multiquadric functions, associated with the three aerodynamic coefficients (stage that is not performed by the piecewise linear multivariable interpolation), while CT and RT correspond to the time required to obtain and read all the sample data that will compose the functions, that is, all P(5,51) points for each function, which makes a total of 765 sample points, 255 for each particular function.

5.2. Four Independent Variables

5.2.1. Airfoil and Domain Applied Throughout the Study. The aerodynamic coefficients of airfoils depend on set parameters to be estimated. In wing optimization studies, it is common to analyze different flap sizes and deflections at different design points, so it becomes a slow process to adjust these parameters in XFOIL each time a change is requested. The

Table 6: Results obtained from the approximation functions.

DAE-21 airfoil-CL/CD/CM-form factor = 0			
Function	REL.P	CT (s)	GT (s)
Function MQ C(5,5)	8.9%/24.2%/7.7%	25.28	0.047
Function MQ C(5,25)	1.9%/1.1%/0.9%	25.28	0.141
Function MQ C(5,35)	0.4%/0.7%/0.4%	25.28	0.281
Multivariable interpolation	~0%	25.28	Not applicable

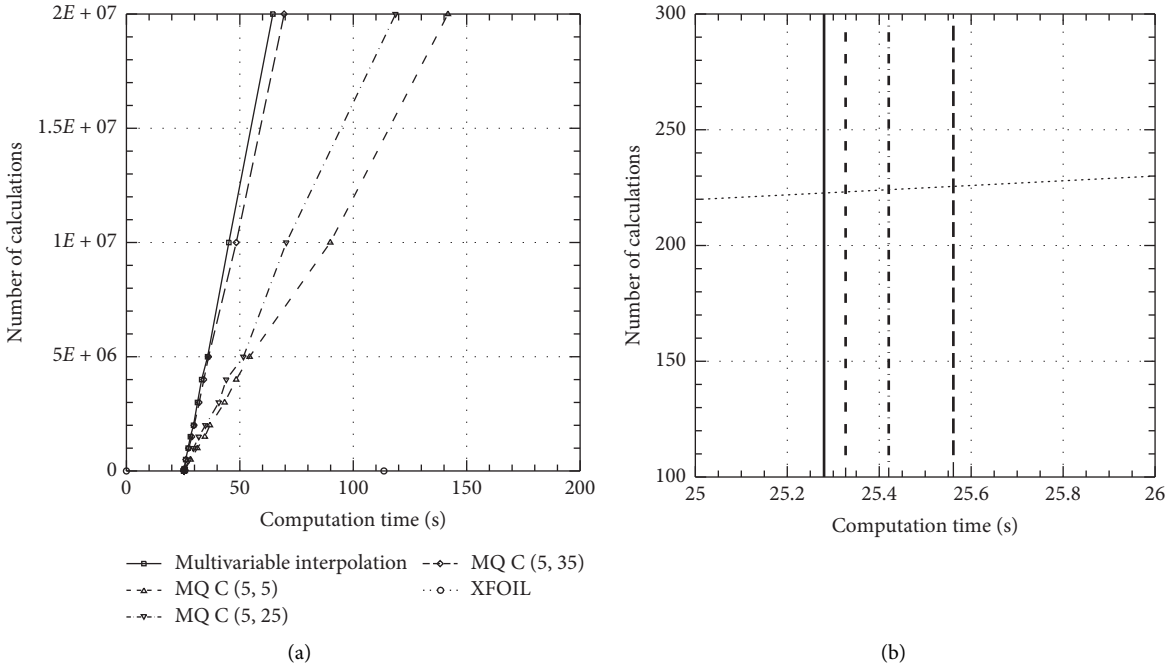

Figure 13: Number of calculations versus computation time with two variables for all studied methods. (a) Overall view. (b) Detailed view.

implemented alternative is the creation of continuous functions whose solution (value of the aerodynamic coefficients) is dependent on the following input variables for the whole domain:

$$x, y, z, t = (\text{Reynolds number, angle of attack, flap chord ratio, flap deflection}). \tag{19}$$

The dataset P(5,51,5,5) was selected and applied to the UBI_03_016 airfoil to define the multiquadric and multivariable interpolation functions within the following limits:

$$\begin{aligned} \text{Reynolds number} \equiv \text{Re} &\in [75000, 675000], \\ \text{Angle of attack}\,[^\circ] \equiv \alpha &\in [-5, 20], \\ \text{Flap chord ratio}\,[\%] \equiv c_{\text{Flap}} &\in [20, 40], \\ \text{Flap deflection}\,[^\circ] \equiv \delta_{\text{Flap}} &\in [0, 10]. \end{aligned} \tag{20}$$

Thus, increments of 150,000, 0.5, 5, and 2.5 for Re, α, c_{flap}, and δ_{flap}, respectively, were used to define the known data points in the given intervals.

5.2.2. Number of Centers and Influence of Form Factor. In order to represent the influence of the number of centers in the REL.P parameter of the given multiquadric function and taking into account the obtained and analyzed data, only the situation where C(Re,α,5,5) with a constant form factor equal to zero was represented. Once again and similar to the two-dimensional analysis, the increase in the number of centers is associated with a decrease in REL.P of the multiquadric function. It is also clear a greater difficulty in representing the drag coefficient compared to the other two other aerodynamic coefficients as seen in Figure 14.

5.2.3. Computation Time. The time required to perform N calculations using multiquadric functions, multivariable interpolation, and XFOIL was calculated to obtain the value of the three aerodynamic coefficients. Again, the input data corresponding to Re, α, c_{flap}, and δ_{flap} were randomly selected for each performed calculation using the approximation methods. However, when using XFOIL, due to interface limitations the

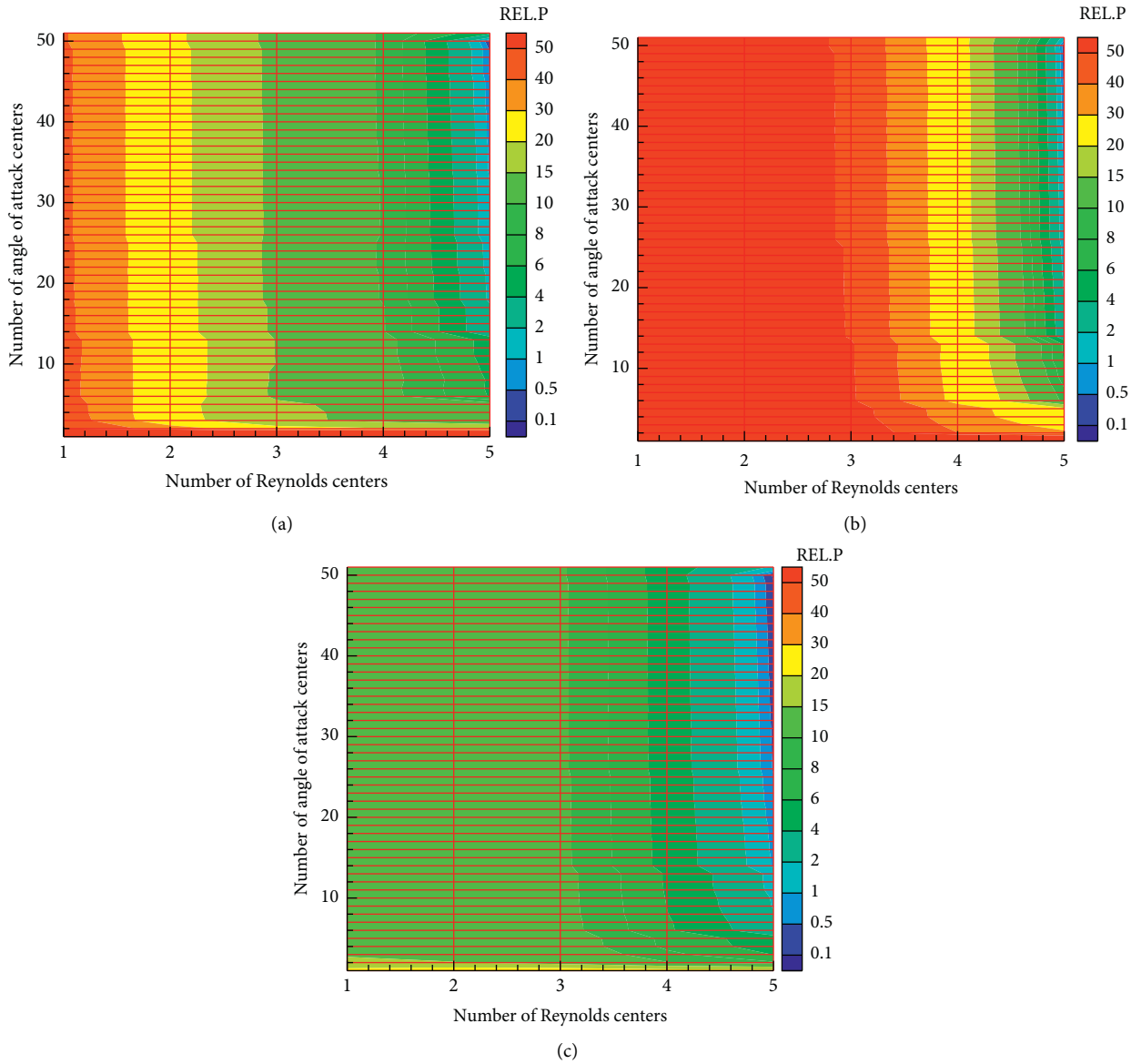

Figure 14: REL.P versus centers' combination C(Re,α,5,5). (a) Lift coefficient, (b) drag coefficient, and (c) pitching moment coefficient.

values of α, c_{flap}, and δ_{flap} were kept constant while different Re were assigned within the bounds of its domain.

Since the set of sampling points used to define each function increased to 6375 points, which makes up a total of 19125 sample points for the three approximation functions associated with the three different aerodynamic coefficients, the time required to collect all the points using the XFOIL (CT) also increased, as well as the time for reading the data (RT). Since this problem has a higher space dimension and more data points, the size of the support matrices needed to create the multiquadric functions increases, which requires more time to perform all the operations needed to obtain the function coefficients, which in turn implies an increase in GT. Once again, both CT and RT are only performed once to obtain and read the 19,125 sample points, both in multivariable interpolation and in multiquadric functions. The latter has also to complete GT relative to the time required to generate the three approximation functions, which is performed only once before the start of the ST phase. The results are shown in Table 7. Like the two-dimensional situation, multivariable interpolation offers a faster calculation time even when compared to multiquadric functions with only 225 of the 6375 possible centers (MQ C(3,3,5,5)). Once again, the same number of centers was always applied to define the three multiquadric functions associated with the three aerodynamic coefficients. The greater difficulty in representing the evolution of the drag coefficient also became clear, since its function requires a larger number of centers compared to the remaining two existing aerodynamic coefficients.

In this situation, since CT and RT are higher and more input data exists, the use of the approximation methods only becomes beneficial after approximately 6000 calculations, as can be seen from Figure 15(b).

Table 7: Results for the four-variable case obtained from the approximation functions.

UBI_03_016 airfoil-CL/CD/CM				
Function	REL.P	CT [s]	RT [s]	GT [s]
Function MQ C(3,3,5,5)	16.1%/55.9%/12.2%	646.4	1.797	7.359
Function MQ C(5,5,5,5)	11.3%/21.2%/4.8%	646.4	1.797	57.93
Function MQ C(5,10,5,5)	5.4%/6.6%/1.9%	646.4	1.797	237.93
Multivariable interpolation	~0%	646.4	1.797	Not applicable

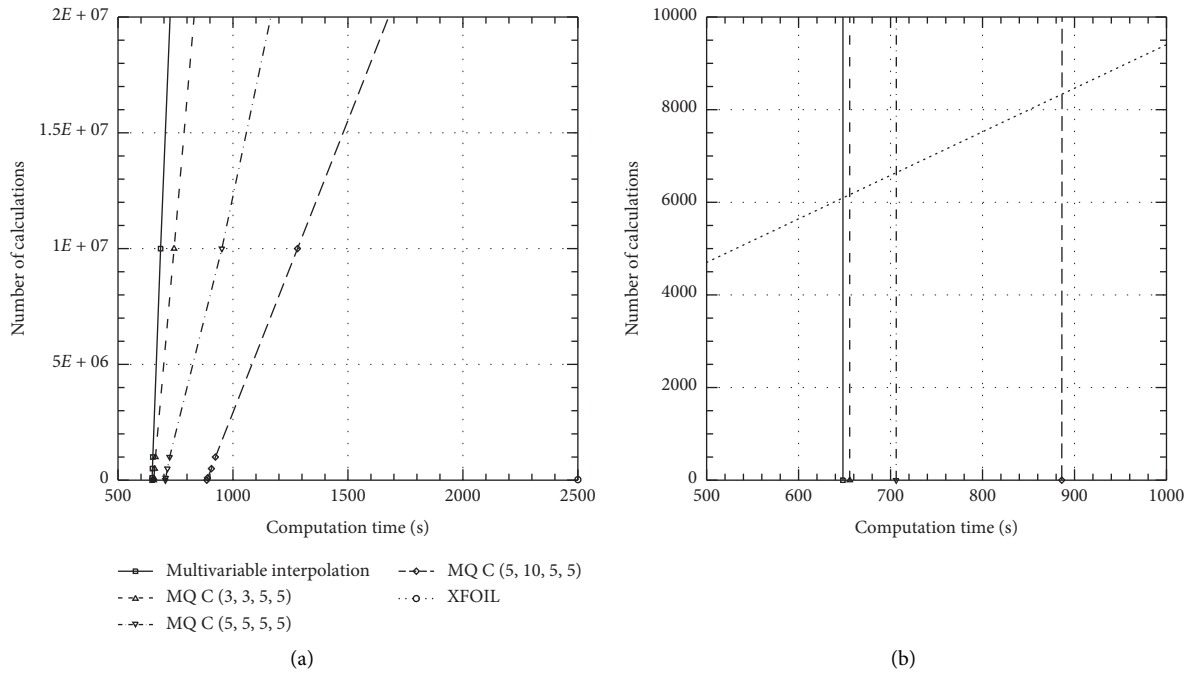

Figure 15: Number of calculations versus computation time with four variables for all studied methods. (a) Overall view. (b) Detailed view.

If the user is only interested in calculating a single aerodynamic coefficient, the sampling points associated with that same function are read and only a single function associated with that specific aerodynamic coefficient is generated, thus requiring less computing time. However, XFOIL always calculates all three aerodynamic coefficients for a given set of input data, regardless of the user's settings. Another feature that favors the use of the approximation functions over XFOIL direct analysis is that the latter sometimes does not guarantee convergence for certain input data, unlike the approximation functions which are defined for the whole domain.

5.3. Computation Time Comparison. For the two-dimensional case, the slopes of linear regressions in Figure 13 produced 486,618 calculations/s and 442,234 calculations/s for the multivariable interpolation method and the multiquadric function C(5,5), respectively, whereas for four independent variables, 247,537 calculations/s and 116,439 calculations/s were obtained from Figure 15(a) for the multivariable interpolation and for the multiquadric function C(3,3,5,5), respectively. In the two-dimensional case, only 25 of the 255 available centers (corresponding to about 9.8% of the available centers) were used, while with four independent variables 225 of the 6375 available centers (corresponding to approximately 3.53% of the available centers) were used. The ratio of the slope of the linear regression of the multivariable interpolation method to the linear regression of the multiquadric function is 1.1 for the two-dimensional case and 2.13 with four independent variables, which means that even with a reduced percentage of centers used to define the multiquadric function (which in turn worsens the quality of the generated approximation), the difference between the slope associated with the calculation time of both approaches increases in benefit of the multivariable interpolation, when increasing the dimensional space of the problem.

6. Conclusions

This work studied the application of multiquadric functions to represent the aerodynamic lift, drag, and pitching moment coefficients of airfoils. The method can approximate a given dataset of any spatial dimension, automatically calculating the location of the center points (as many as required) and also calculating the form factor according to the chosen method. It was concluded that, with the increase of

center points, it was possible to obtain better approximations, with only a few exceptions. It was also realized the enormous influence of the form factor, which also plays a relevant role in the quality of the approximation, since it is directly responsible for the ill-conditioning of the interpolation matrix used in the generation of the multiquadric function. The method using FFSQP was able, in most situations, to converge to the best possible form factor value (with lower associated REL.P), although this requires that the user specifies the number of centers beforehand, without the guarantee of obtaining the best solution for this combination, since function REL.P versus form factor may have a local minimum in the starting point neighborhood.

To assess the precision of the multiquadric approximation, it was compared with the piecewise linear multivariable interpolation method, since this is a method widely applied in the approximate representation of the aerodynamic coefficients. Considering the quality of the approximation at points outside the applied dataset, but within the selected domain, it was concluded that, by varying the set of points used to define the functions, there was no clear advantage of any of the approaches. This analysis was performed in a two-dimensional space, but similar results are expected to be obtained for larger dimensions.

The processing time required to perform a certain number of calculations (to obtain the value of the three aerodynamic coefficients) based on a random set of input data was evaluated. For the two-dimensional case, it was observed that the use of the approximation functions is preferred over XFOIL after about 240 calculations, whereas for the four-dimensional case, the break-even point increases to approximately 6000 calculations. Regarding the computation time, the multivariable interpolation approach has shown to be faster compared to multiquadric functions. The increase in the spatial dimension of the problem caused the difference in ST between the multiquadric functions and the multivariable interpolation approach to increase, in favor of the latter, even with a reduced percentage of centers applied to develop the multiquadric function (which in turn worsens the quality of the generated approximation).

Multiquadric functions are still experiencing increased widespread application across many areas of science and engineering. In order to improve the process of defining the multiquadric functions with increased precision, some open issues still need to be addressed: new techniques for placement of center, application of a variable form factor for each center to which it is applied, methods of direct calculation of the form factor, methods that aim to lower the conditioning number of the interpolation matrices with the gradual increase of the form factor, and understanding the best techniques to define the domain and the set of points to apply in the problem.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This work was supported by the Universidade da Beira Interior.

References

[1] R. L. Hardy, "Multiquadric equations of topography and other irregular surfaces," *Journal of Geophysical Research*, vol. 76, no. 8, pp. 1905–1915, 1971.

[2] R. Franke, "Scattered data interpolation: tests of some methods," *Mathematics of Computation*, vol. 38, no. 157, pp. 181–200, 1982.

[3] C. A. Micchelli, "Interpolation of scattered data: distance matrices and conditionally positive definite functions," *Constructive Approximation*, vol. 2, no. 1, pp. 11–22, 1986.

[4] E. J. Kansa, "Multiquadrics – a scattered data approximation scheme with applications to computational fluid-dynamics," *Computers and Mathematics with Applications*, vol. 19, no. 8-9, pp. 127–145, 1990.

[5] E. J. Kansa and Y. C. Hon, "Circumventing the ill-conditioning problem with multiquadric radial basis functions: applications to elliptic partial differential equations," *Computers & Mathematics with Applications*, vol. 39, no. 7-8, pp. 123–137, 2000.

[6] S. A. Sarra, "Integrated multiquadric radial basis function approximation methods," *Computers & Mathematics with Applications*, vol. 51, no. 8, pp. 1283–1296, 2006.

[7] R. L. Hardy, "Theory and applications of the multiquadric-biharmonic method 20 years of discovery 1968-1988," *Computers & Mathematics with Applications*, vol. 19, no. 8-9, pp. 163–208, 1990.

[8] G. E. Fasshauer, *Meshfree Approximation Methods with MATLAB*, vol. 6, World Scientific Co, Pte. Ltd., Singapore, 2007.

[9] 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.

[10] M. Mongillo, "Choosing basis functions and shape parameters for radial basis function methods," *SIAM Undergraduate Research Online*, vol. 4, 2011.

[11] M. Ranjbar, "A new variable shape parameter strategy for Gaussian radial basis function approximation methods," *Mathematics and Computer Science Series*, vol. 42, no. 2, pp. 260–272, 2015.

[12] R. E. Carlson and T. A. Foley, "The parameter R2 in multiquadric interpolation," *Computers & Mathematics with Applications*, vol. 21, no. 9, pp. 29–42, 1991.

[13] T. A. Foley, "Near optimal parameter selection for multiquadric interpolation," *Journal of Applied Science and Computations*, vol. 1, pp. 54–69, 1994.

[14] E. J. Kansa and R. E. Carlson, "Improved accuracy of multiquadric interpolation using variable shape parameters,"

Computers & Mathematics with Applications, vol. 24, no. 12, pp. 99–120, 1992.

[15] R. K. Beatson, J. B. Cherrie, and C. T. Mouat, "Fast fitting of radial basis functions: methods based on preconditioned GMRES iteration," *Advances in Computational Mathematics*, vol. 11, no. 2-3, pp. 253–270, 1999.

[16] 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.

[17] V. Bayona, M. Moscoso, and M. Kindelan, "Optimal constant shape parameter for multiquadric based RBF-FD method," *Journal of Computational Physics*, vol. 230, no. 19, pp. 7384–7399, 2011.

[18] L. Machado, R. M. S. De Oliveira, W. C. B. Souza, R. C. F. Araújo, and M. E. L. Tostes, "An automatic methodology for obtaining optimum shape factors for the radial point interpolation method," *Journal of Microwaves, Optoelectronics and Electromagnetic Applications*, vol. 10, no. 2, pp. 389–401, 2011.

[19] J. Biazar and M. Hosami, "An algorithm for the shape parameter in radial basis functions interpolation," in *International Conference on Advances in Applied Mathematics and Mathematical Physics*, Turkey, August 2014.

[20] Y. C. Hon, R. Schaback, and X. Zhou, "An adaptive greedy algorithm for solving large RBF collocation problems," *Numerical Algorithms*, vol. 32, no. 1, pp. 13–25, 2003.

[21] L.-T. Luh, "The choice of the shape parameter-a friendly approach," *Engineering Analysis with Boundary Elements*, vol. 98, pp. 103–109, 2019.

[22] G. Trejo-Caballero, H. Rostro-Gonzalez, C. H. Garcia-Capulin, O. G. Ibarra-Manzano, J. G. Avina-Cervantes, and C. Torres-Huitzil, "Automatic curve fitting based on radial basis functions and a hierarchical genetic algorithm," *Mathematical Problems in Engineering*, vol. 2015, Article ID 731207, 14 pages, 2015.

[23] S. A. Sivaram and K. J. Vinoy, "Inverse multiquadric radial basis functions in eigenvalue analysis of a circular waveguide using radial point interpolation method," *IEEE Microwave and Wireless Components Letters*, vol. 30, no. 6, pp. 537–540, 2020.

[24] J. Gu and J.-H. Jung, "Adaptive Gaussian radial basis function methods for initial value problems: construction and comparison with adaptive multiquadric radial basis function methods," *Journal of Computational and Applied Mathematics*, vol. 381, 2021.

[25] M. A. Khan, A. B. Altamimi, Z. H. Khan et al., "Multiquadric radial basis function approximation scheme for solution of total variation based multiplicative noise removal model," *CMES-computer Modeling in Engineering & Sciences*, vol. 126, no. 1, pp. 55–88, 2021.

[26] K. Amouzgar and N. Strömberg, "Radial basis functions as surrogate models with a priori bias in comparison with a posteriori bias," *Structural and Multidisciplinary Optimization*, vol. 55, no. 4, pp. 1453–1469, 2017.

[27] K. Amouzgar, S. Bandaru, and A. H. C. Ng, "Radial basis functions with a priori bias as surrogate models:A comparative study," *Engineering Applications of Artificial Intelligence*, vol. 71, pp. 28–44, 2018.

[28] E. Meijering, "A chronology of interpolation: from ancient astronomy to modern signal and image processing," *Proceedings of the IEEE*, vol. 90, no. 3, pp. 319–342, 2002.

[29] M. Drela, "XFOIL: an analysis and design system for low Reynolds number airfoils," in *Low Reynolds Number Aerodynamics*, T. J. Mueller, Ed., Lecture Notes in Engineering, Springer-Verlag, New York, NY, USA, 1989.

[30] J. D. Anderson, *Fundamentals of Aerodynamics*, 5th edition, McGraw-ill, New York, NY, USA, 2011.

[31] J. L. Zhou, A. L. Tits, and C. T. Laurence, *User's Guide for FFSQP Version 3.7: A FORTRAN Code for Solving Constrained Nonlinear (Minimax) Optimization Problems, Generating Iterates Satisfying All Inequality and Linear Constraints*, Electrical Engineering Department and Institute for Systems Research, University of Maryland, College Park, MD, USA, 1997.