Gónzalez-Rodríguez, P., Bayona, V., Moscoso, M., & Kindelan, M.. (2015). Laurent series based RBF-FD method to avoid ill-conditioning. *Engineering Analysis with Boundary Elements*, 52, pp. 24–31

DOI:10.1016/j.enganabound.2014.10.018

Laurent series based RBF-FD method to avoid ill-conditioning

Pedro Gonzalez-Rodriguez, Victor Bayona, Miguel Moscoso, Manuel Kindelan *

Universidad Carlos III de Madrid, Avenida de la Universidad 30, 28911 Leganés, Spain

A B S T R A C T

We propose a new approach to avoid the inherent ill condition in the computation of RBF FD weights, which is due to the fact that the RBF interpolation matrix is nearly singular. The new approach is based on the semi analytical computation of the Laurent series of the inverse of the RBF interpolation matrix. Once the Laurent series is obtained, it can be used to compute the RBF FD weights of any differential operator exactly without extra cost. The proposed method also provides analytical formulas for the RBF FD weights in terms of the parameters involved in the problem. These formulas can be used to derive the exact dependence of the truncation error in the approximation of any differential operator of a given function. Furthermore, from the analysis presented here one can derive the values of the parameters involved in the problem for which the RBF interpolation matrix becomes ill conditioned and, hence, for which the weights cannot be obtained numerically.

Keywords:
Radial basis functions
Finite differences

1. Introduction

The radial basis functions finite difference method (RBF FD) is essentially a generalization of the classical finite difference (FD) method to scattered node layouts [24 26]. In its most general form, the FD method consists of approximating a derivative of a function at a given point, based on a weighted linear combination of the value of the function at some surrounding nodes. However, if one allows the nodes of the stencil to be placed freely, so that a good discretization of the physical domain is obtained, then the problem of computing the weights of scattered FD formulas is not well defined [23]. For instance, what mixed terms should be included in the multivariate polynomial interpolant to the nodes is not defined [27]. This and other ambiguities are resolved if RBF interpolation is used to generate the weights in the FD formula instead.

In fact, the weights of scattered FD formulas can be obtained as the RBF FD weights in the limit of increasingly flat basis functions. As a result, this limit has been thoroughly studied by many researchers during the last few years. In the case of RBF interpolation, Fornberg et al. [7,10] showed that if a limiting interpolant exists, it must be a (multivariate) polynomial. From this result they concluded that RBF's can be a *tool for generalizing, to irregular grids and domains, the classical spectral methods*. The existence of a limiting interpolant implies that the underlying interpolation problem is well behaved, even though the RBF interpolation matrix might be severely ill conditioned. It should be pointed out, however, that in certain exceptional situations (5×5 cartesian node layout, for instance) the interpolant diverges in this limit [7,10].

In the case of RBF FD methods, since the RBF interpolants usually converge to polynomial interpolants in the limit of increasingly flat radial basis functions, all classical 1 D and many higher dimensional FD formulas can be recovered as limiting cases of the corresponding RBF FD formulas [7,19,22,27]. However, trying to compute the FD weights using RBF interpolation is made difficult by the fact that the interpolation matrix becomes ill conditioned in the flat basis limit. Thus, the underlying problem is well posed (the weights of the RBF FD formula are well defined numbers) but the most straightforward computational procedure (RBF direct) is ill conditioned.

The flatness of the radial basis functions is controlled by a shape parameter ϵ. Flatness increases as ϵ decreases and, at the same time, the accuracy of the RBF FD formulas usually increases. In fact, as ϵ goes to zero, the accuracy approaches a constant corresponding to the accuracy of standard finite differences. Thus, the values of ϵ where accuracy is high often lie in the region of ill conditioning. As a result, a great amount of work has been devoted to develop numerical algorithms to avoid ill conditioning of the RBF method in the limit $\epsilon \to 0$. The first successful approach [11] was based in considering the parameter ϵ as a complex number, and evaluating the interpolant using a Contour Padé algorithm. This approach is also used in the RBF RA method [15]. The main limitation of this method is that it is only applicable to relatively small node sets. A different approach was later proposed in [13,14]. It is based on changing the basis to one that is better suited for numerics, while staying in the same approximation space. The QR decomposition forms an integral part in this change of basis, hence the name of the method (RBF QR). It has been used to compute the weights of RBF FD stencils [20] and to compute RBF FD

* Corresponding author. Fax: +34 91 624 91 29.
E-mail addresses: pgonzale@ing.uc3m.es (P. Gonzalez-Rodriguez), vbayona@ing.uc3m.es (V. Bayona), moscoso@math.uc3m.es (M. Moscoso), kinde@ing.uc3m.es (M. Kindelan).

approximations to the Poisson's equation [6]. A somewhat similar alternative (RBF GA) was recently proposed in [15]. It is also based on a change of basis, but requires no truncation of infinite sums. A drawback of this method is that it is limited to Gaussian RBFs. The low computational cost (compared to RBF QR) and simple implemen tation makes it a useful alternative for computing RBF FD stencils, though. Other attempts to circumvent the problem of ill conditioning have been based on using extended precision arithmetic [17], or using truncated singular value decomposition [8,21].

The ill conditioning in the computation of the RBF FD weights is due to the fact that the RBF interpolation matrix becomes singular when $\epsilon = 0$. The objective of this paper is to propose a novel technique to avoid the ill conditioning of the problem. This technique is based on the analytical computation of the Laurent series of the inverse of the RBF interpolation matrix. The Laurent series is an infinite sum of matrices multiplied by powers of the parameter $\delta = (\epsilon h)^2$, where h is a characteristic inter nodal distance. The matrix coefficients in this series are computed using a recursive algorithm based on the analytical method proposed by Avrachenkov and Lasserre for comput ing inverses of nearly singular matrices [2]. The algorithm is imple mented in Matlab using floating point arithmetic. The details of the implementation are described in [16]. Our method is semi analytical because the Laurent series is derived from analytical formulas, but the implementation described in [16] involves a quite significant amount of numerical computations. Once the Laurent series of the inverse of the RBF interpolation matrix has been obtained (as function of δ), it is straightforward to compute the RBF FD weights for any differential operator by just multiplying term by term the Laurent series by the Taylor series of the differential operator. Hence, with this method, the problem of ill conditioning in the computation of the RBF FD weights is completely eliminated.

The recursive algorithm described in [16] is an efficient algo rithm compared to the computation of the inverse of a matrix using a symbolic language such as Mathematica. However, the approach proposed here to compute the RBF weights is not competitive with other existing algorithms (Contour Padé,RBF QR, RBF RA, RBF GA methods,…). If one is interested in the weights of RBF FD formulas for a particular value of δ, these algorithms are more efficient and can handle larger node layouts.

Our semi analytical approach offers other advantages. For instance, a very relevant and useful byproduct of the algorithm is the order of the singularity of the inverse (the exponent of the leading order of the Laurent series). In this way, it is possible to predict exactly how the condition number of the inverse approaches infinity as δ approaches zero. We finally stress that this approach leads to analytical formulas for the weights in terms of h and ϵ that can be used to derive exact formulas for the local truncation error in terms of h, ϵ, and the value of the function and its derivatives at the node. These formulas can then be used to derive an optimal value of the shape parameter following the approach described in [3 5].

The paper is organized as follows. In Section 2 we describe the RBF FD method to approximate differential operators. In Section 3 we describe how the Laurent series of the inverse can be used to accurately compute the weights of RBF FD formulas. In Section 4 we present results in 2D and 3D using both equispaced and non equispaced nodes and we discuss how the Laurent series of the inverse can be used to compute the optimal value of the shape parameter. Finally, Section 5 contains the main conclusions.

2. RBF-FD formulation

Consider a stencil consisting of n scattered nodes $\mathbf{x}_1, \ldots, \mathbf{x}_n$, and a differential operator $\mathcal{L}$. For a given node, say $\mathbf{x}_1$, the objective is to approximate $\mathcal{L}u(\mathbf{x}_1)$ as a linear combination of the values of u at the n scattered nodes, so that

$$\mathcal{L}u(\mathbf{x}_1) \approx \sum_{i}^{n}{}_{1} \alpha_i u(\mathbf{x}_i). \tag{1}$$

To determine the weighting coefficients α_i, a set of base functions $\phi_i(\mathbf{x})$, $i = 1, \ldots, n$, are required. In that base,

$$\mathcal{L}\phi_j(\mathbf{x}_1) = \sum_{i}^{n}{}_{1} \alpha_i \phi_j(\mathbf{x}_i), \quad j = 1, 2, \ldots, n. \tag{2}$$

This is a system of n linear equations with n unknowns whose solution yields the weighting coefficients α_i. The coefficients α_i depend on the distances from $\mathbf{x}_1$ to the other nodes in the stencil, and on the shape parameter ϵ that controls the flatness of the base functions. In matrix form, Eqs. (2) can be written as

$$A\boldsymbol{\alpha} = \mathbf{f}, \tag{3}$$

where the matrix A is the local RBF interpolation matrix whose elements are

$$A_{i,j} = \phi_j(\mathbf{x}_i). \tag{4}$$

The elements of the vector $\mathbf{f}$ in (3) are the result of applying the differential operator $\mathcal{L}$ to the RBF basis functions, so that

$$f_j = \mathcal{L}\phi_j(\mathbf{x}_1).$$

In the following, we will use the multiquadrics

$$\phi_i(\mathbf{x}) = \sqrt{1 + \epsilon^2 \| \mathbf{x} \quad \mathbf{x}_i \|_2^2}$$

as RBFs. Note that as ϵ decreases the multiquadrics becomes increasingly flat.

For multiquadrics, the elements of the matrix A can be written in terms of a characteristic inter nodal distance, h, as

$$A_{i,j} = \sqrt{1 + (\epsilon h)^2 \left(\frac{\| \mathbf{x}_i \quad \mathbf{x}_j \|_2}{h} \right)^2}.$$

Thus, for a given stencil, the matrix A depends on the locations of the nodes in the stencil and on the parameter $\delta = (\epsilon h)^2$. Note that the matrix A is singular for $\epsilon = 0$.

3. RBF-FD weights

To compute the RBF FD weights one has to solve the linear system (3). This system can be solved numerically, in general. However, if the shape parameter is small the system becomes ill conditioned and the errors in the numerical solution are very large. To overcome this problem we propose a new method which is based on the solution of (3) using the Laurent series of the inverse of the interpolation matrix A.

Let us assume that the locations of the nodes are given. Then, the RBF interpolation matrix depends only on the parameter $\delta = (\epsilon h)^2$. Consider the Taylor series expansion

$$A(\delta) = \sum_{k}^{\infty}{}_{0} \delta^k A_k \tag{5}$$

of the RBF interpolation matrix. For $\delta = 0$, the matrix $A(\delta = 0) = A_0$ is singular (all its elements are equal to one). However, it is not singular for $\delta > 0$. We seek to compute the Laurent series of the inverse of the RBF FD matrix $A(\delta)$ for $\delta > 0$. Thus, we have to compute the matrix coefficients H_k in

$$A^{-1}(\delta) = \sum_{k}^{\infty}{}_{-p} \delta^k H_k, \tag{6}$$

where p is the order of the singularity of the inverse. In the following, we use the algorithm described in [16] to compute the matrices H_k in terms of the matrices A_k in (5).

Let us start by the simplest example of an equispaced, three node stencil ($[x_1 \quad h, x_1, x_1+h]$) to approximate the first and second order derivatives in 1D. In this case, the matrix $A(\delta)$ and the first terms of its Taylor series are given by,

$$A(\delta)=\begin{bmatrix} 1 & \sqrt{1+\delta} & \sqrt{1+4\delta} \\ \sqrt{1+\delta} & 1 & \sqrt{1+\delta} \\ \sqrt{1+4\delta} & \sqrt{1+\delta} & 1 \end{bmatrix} \approx \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} + \delta \begin{bmatrix} 0 & 1/2 & 2 \\ 1/2 & 0 & 1/2 \\ 2 & 1/2 & 0 \end{bmatrix} + \delta^2 \begin{bmatrix} 0 & 1/8 & 2 \\ 1/8 & 0 & 1/8 \\ 2 & 1/8 & 0 \end{bmatrix} + \cdots \tag{7}$$

Clearly, to leading order ($\delta = 0$) the matrix A is singular. The first few terms of the Laurent series of $A^{-1}(\delta)$ are

$$A^{-1}(\delta) \approx \frac{1}{\delta^2}\begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 1/2 & 1 & 1/2 \\ 1/4 & 1/2 & 1/4 \end{bmatrix} + \frac{1}{\delta}\begin{bmatrix} 3/4 & 5/4 & 1/4 \\ 5/4 & 3 & 5/4 \\ 1/4 & 5/4 & 3/4 \end{bmatrix} + \begin{bmatrix} 0 & 1/16 & 1/2 \\ 1/16 & 0 & 1/16 \\ 1/2 & 1/16 & 0 \end{bmatrix} + \cdots \tag{8}$$

Notice that the order of the singularity is $p=2$ and, therefore, as $\delta \to 0$ the inverse approaches infinity as $\delta^{-2} = (\epsilon h)^{-4}$. In (8), only the matrices H_k with $k \leq 0$ are given.

The Laurent series (6) can also be derived using symbolic languages such as Mathematica or Maple. However, the computational complexity is much larger if one uses symbolic computation [1]. In fact, we have not been able to calculate the Laurent series of the perturbed matrix for stencils larger than 13 nodes in a reasonable time. These limitations disappear if the Laurent series is computed with the algorithm described in [16], which computational complexity grows algebraically with the number of nodes ($\propto n^3$), but exponentially with the order of the singularity. In fact, the computational complexity of the proposed algorithm is $O(2n^3B)$, where

$$B = 2\sum_{j}^{2p-1}\sum_{i}^{j+1}\frac{j!(2i \quad 1)!}{(j \quad i+1)!((i \quad 1)!)^3}\Bigg)$$

and p is the order of the singularity. If we plug the numbers in the formula, the numerical complexity grows as

$p=1:\quad B=14$
$p=2:\quad B=598$
$p=3:\quad B=18694$
$p=4:\quad B=544254$
$p=5:\quad B=15287758$
$p=6:\quad B=420038854$
$p=7:\quad B=11368586038$
$p=8:\quad B=304362660958$

Using the proposed method, the Laurent series can be computed in seconds for singularities up to order five, which corresponds to fairly large stencils in 2D, and even larger in 3D. For instance, the computational time on an Intel Core i7 950 @3.07 GHz processor, for 9, 11 and 13 equispaced nodes in 2D are 5.7, 5.9 and 7.5 s respectively. With a symbolic language (Mathematica) the corresponding times are 4.8, 38.5 and 29134 s respectively. For stencils with more than 13 nodes it is not possible to obtain the solution. Timing results for other techniques to avoid ill conditioning can be found in Fig. 8 of Ref. [15].

Once the inverse of the RBF FD matrix is known, the weights of the RBF FD formula are given by $\boldsymbol{\alpha} = A^{-1}, \mathbf{f}$. The components of the vector $\mathbf{f}$ are the result of applying the differential operator to each of the RBF basis functions, and evaluating the resulting functions at $\mathbf{x}_1$. Vector $\mathbf{f}$ can also approximated by its Taylor series

$$\mathbf{f} = \sum_{k}^{\infty}{}_{q} \delta^k \mathbf{f}_k,$$

where q is the order of the leading term in the expansion of $\mathbf{f}$. For instance, in the case of the RBF FD formula to approximate the first derivative,

$$\mathbf{f} = \frac{\delta}{h}\begin{bmatrix} (1+\delta)^{-1/2} \\ 0 \\ (1+\delta)^{-1/2} \end{bmatrix} \approx \frac{\delta}{h}\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} + \frac{\delta^2}{h}\begin{bmatrix} 1/2 \\ 0 \\ 1/2 \end{bmatrix} + \frac{\delta^3}{h}\begin{bmatrix} 3/8 \\ 0 \\ 3/8 \end{bmatrix} + \cdots \tag{9}$$

and, therefore, $q=1$. Multiplying this series by the Laurent series for $A^{-1}(\delta)$ (8) we obtain the RBF FD weights

$$\boldsymbol{\alpha} = A^{-1}\mathbf{f} = \frac{1}{h}\begin{bmatrix} 1/2 \\ 0 \\ 1/2 \end{bmatrix} + \frac{\delta}{h}\begin{bmatrix} 1/4 \\ 0 \\ 1/4 \end{bmatrix} + \frac{\delta^2}{h}\begin{bmatrix} 9/16 \\ 0 \\ 9/16 \end{bmatrix} + \cdots \tag{10}$$

which are equal to those computed in [3]. The leading order approximation to the RBF FD weights are the standard finite difference weights. Notice that $H_{-2}\mathbf{f}_1 = H_{-2}\mathbf{f}_2 = 0$. Thus, the leading order approximation is given by $H_{-1}\mathbf{f}_1$ The order δ correction to the weights in (10) is computed from $H_{-2}\mathbf{f}_3 + H_{-1}\mathbf{f}_2 + H_0\mathbf{f}_1$, and the order δ^2 correction from $H_{-2}\mathbf{f}_4 + H_{-1}\mathbf{f}_3 + H_0\mathbf{f}_2 + H_1\mathbf{f}_1$. Thus, to compute the weights to order δ^2 it is necessary to compute the Laurent series for $A^{-1}(\delta)$ up to order one only, and the Taylor series for $\mathbf{f}$ up to order four.

Fig. 1 shows the error in the approximation of the first derivative of function $u(x) = \exp(\quad x^2)$ (black) with the RBF FD formula for three nodes. There is a very good agreement between the error using the RBF FD weights computed by direct numerical solution of the linear system (3) (solid line) and the error using the Taylor series for the weights (10) (dashed line). Notice that for small values of δ, the matrix A becomes ill conditioned and the weights cannot be computed numerically. Notice also, that there is an *optimal* value $\delta \approx 4 \times 10^{-4}$ for which the error is minimum. This optimal value is located in the region where the matrix A is well conditioned and, therefore, the weights can be computed numerically without problem. However, in the case of the function $u(x) = \exp(\quad 1.5x^2)$ (red), the optimal value of δ lies in the region of

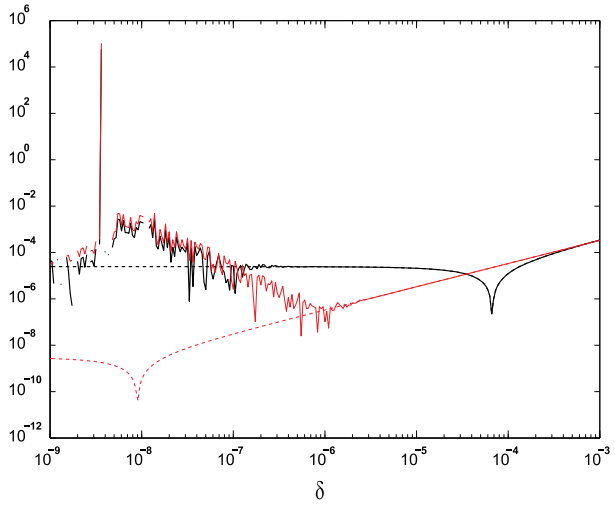

Fig. 1. Error in approximation of first derivative with the RBF-FD formula for three nodes. Solid line: numerically computed weights. Dashed line: weights computed from Taylor series (10). Black: function $u(x) \quad \exp(-x^2), x \quad 1, h \quad 0.01$. Red: function $u(x) \quad \exp(-1.5x^2), x \quad 1, h \quad 0.01$. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

ill conditioning and, therefore, the weights can only be computed from the Taylor series (10).

Similarly, the Taylor series for the second derivative in 1D is

$$\mathbf{f}=\frac{\delta}{h^2}\begin{bmatrix}(1+\delta)^{-3/2}\\ 1\\ (1+\delta)^{-3/2}\end{bmatrix}\approx\frac{\delta}{h^2}\begin{bmatrix}1\\1\\1\end{bmatrix}+\frac{\delta^2}{h^2}\begin{bmatrix}3/2\\0\\3/2\end{bmatrix}+\frac{\delta^3}{h^2}\begin{bmatrix}15/8\\0\\15/8\end{bmatrix}+\cdots, \tag{11}$$

so that the corresponding RBF FD weights are

$$\boldsymbol{\alpha}=A^{-1}\mathbf{f}=\frac{1}{h^2}\begin{bmatrix}1\\2\\1\end{bmatrix}+\frac{\delta}{h^2}\begin{bmatrix}1\\2\\1\end{bmatrix}+\frac{\delta^2}{h^2}\begin{bmatrix}\frac{195}{64}\\ \frac{219}{32}\\ \frac{195}{64}\end{bmatrix}+\cdots.$$

If the number of nodes in the stencil increases, the inverse $A^{-1}(\delta)$ approaches the singularity faster. For instance, with a stencil of four nodes ($[x_1 \quad h, x_1, x_1+h, x_1+2h]$), we obtain

$$A^{-1}(\delta)\approx\frac{1}{\delta^3}\begin{bmatrix}\frac{1}{36}&\frac{1}{12}&\frac{1}{12}&\frac{1}{36}\\ \frac{1}{12}&\frac{1}{4}&\frac{1}{4}&\frac{1}{12}\\ \frac{1}{12}&\frac{1}{4}&\frac{1}{4}&\frac{1}{12}\\ \frac{1}{36}&\frac{1}{12}&\frac{1}{12}&\frac{1}{36}\end{bmatrix}+\frac{1}{\delta^2}\begin{bmatrix}\frac{1}{3}&\frac{19}{24}&\frac{2}{3}&\frac{5}{24}\\ \frac{19}{24}&2&\frac{15}{8}&\frac{2}{3}\\ \frac{2}{3}&\frac{15}{8}&2&\frac{19}{24}\\ \frac{5}{24}&\frac{2}{3}&\frac{19}{24}&\frac{1}{3}\end{bmatrix}+\cdots$$

We observe that, in 1D, the order of the singularity of the inverse of the RBF FD matrix corresponding to a stencil of n nodes is δ^{1-n}. We also point out that the leading order term in the Taylor series of the weights, i.e. $\boldsymbol{\alpha}(\delta=0)$, gives the weights for standard finite differences in all the cases.

The same procedure can be used to derive the Laurent series of $A^{-1}(\delta)$ and the corresponding Taylor series for the weights of stencils in 2D and 3D. Some relevant results using equispaced and non equispaced nodes are presented in Section 4.

3.1. Condition number

The ill conditioning of system (3) is determined by the condition number of matrix A: $N_{\text{cond}}=\|A\|\,\|A^{-1}\|$. This number can be estimated from the leading order term of the Laurent series using the fact that $\|A_0\|=n$ and that in 1D the order of the singularity is $p=n-1$. Thus, $N_{\text{cond}}\approx n\|H_{1-n}\|\delta^{1-n}$. Fig. 2 shows with a dashed line, the dependence for n=2, 3 and 4 equispaced nodes in 1D of the condition number of the inverse matrix $A^{-1}(\delta)$ with δ. With a solid line we show the dependence estimated from the leading order term of the Laurent series, $N_{\text{cond}}\approx n\|H_{1-n}\|\delta^{1-n}$. Notice that for small values of δ both curves are undistinguishable until the matrix becomes ill conditioned. Similarly in 2D and 3D, if the singularity of the inverse is of order p and the size of A is n then $N_{\text{cond}}\approx n\|H_{-p}\|\delta^{-p}$. If we consider that the matrix becomes ill conditioned when the condition number is greater than 10^{16}, then the minimum value of δ that can be used to avoid ill conditioning is simply given by

$$\delta_{\min}=\left(\frac{n\|H_{-p}\|}{10^{16}}\right)^{1/p}. \tag{12}$$

These critical values are shown with a dashed vertical line in Fig. 2. In fact, since $\|H_{-p}\|$ is of order one or smaller, a simple rule to find an effective value of δ is simply to take $\delta\approx(10^{-16}n)^{1/p}$.

3.2. Local truncation error

The Taylor series solution for the RBF FD weights can also be used to derive approximate formulas for the local truncation error in approximating derivatives of a given function. These formulas depend on the values of the parameters δ and h, and on the values of the function and its derivatives at the node. They can be very useful in order to locate the optimal value of the shape parameter that minimizes the error.

As an example, consider the Laplacian of the function,

$$u(\mathbf{x})=\exp\left[-\left(x-\frac{1}{4}\right)^2-\left(y-\frac{1}{2}\right)^2\right]\cos(2\pi y)\sin(\pi x), \tag{13}$$

which was used by Wright and Fornberg [27], and consider the six node stencil $[(0,0),(-1.17h,0.72h),(0.82h,1.21h),(0.4h,-0.5h),(1.16h,0.28h),(0,1.19h)]$ which is shown in the inset of Fig. 3. The error in approximating the Laplacian at $\mathbf{x}_1=(0,0)$ with the RBF FD method using the weights computed numerically by solving system (3) is shown as a function of δ in the Figure (solid line). The Laurent series for the inverse of the interpolation matrix can be used to derive the Taylor series for the weights of the Laplacian using the same procedure described

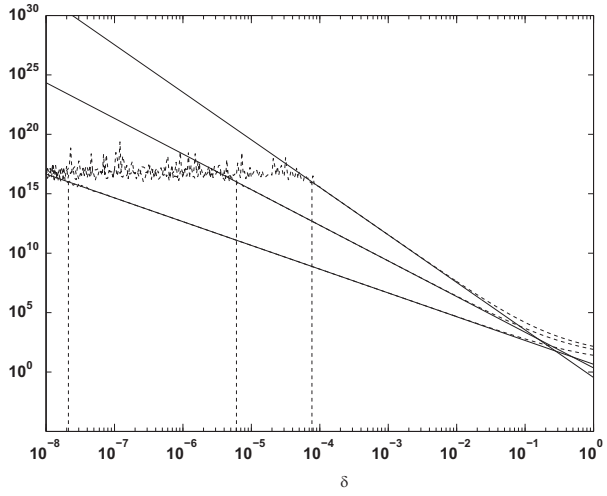

Fig. 2. Solid line: condition number estimated by $n\|H_{1-n}\|\delta^{1-n}$, as a function of δ. Dashed line: Condition number of inverse matrix A^{-1} as a function of δ. The three curves correspond to $n=3, 4, 5$ equispaced nodes. The minimum values of δ (12) are shown with vertical dashed lines.

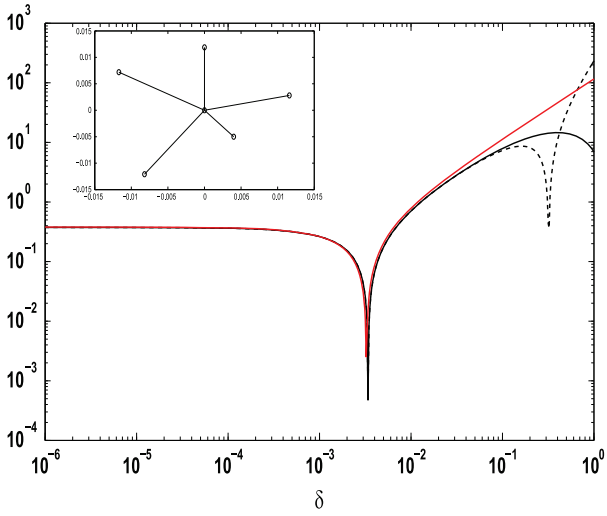

Fig. 3. Error in approximating the Laplacian of function (13) at $\mathbf{x}_1=(0,0)$ with $n=6$ non-equispaced RBF-FD formula. Solid line: numerically computed RBF-FD weights ($h=0.01$). Dashed line: weights computed with Taylor series (14). Red line: Eq. (15). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

in Section 3. It results in

$$\boldsymbol{\alpha} = A^{-1}\mathbf{f} = \frac{1}{h^2}\begin{pmatrix} 3.1233 \\ 0.5527 \\ 0.4986 \\ 1.0407 \\ 0.5511 \\ 0.4802 \end{pmatrix} + \frac{\delta}{h^2}\begin{pmatrix} 2.8168 \\ 0.4415 \\ 0.9759 \\ 0.0201 \\ 0.6916 \\ 0.6876 \end{pmatrix} + \frac{\delta^2}{h^2}\begin{pmatrix} 0.2521 \\ 0.3870 \\ 3.9009 \\ 7.0213 \\ 4.1827 \\ 0.0713 \end{pmatrix} \tag{14}$$

The error in approximating the Laplacian with the RBF FD weights approximated by its Taylor series (14) are shown with a dashed line in the Figure. Notice that there is very good agreement with the numerical results obtained using the exact RBF FD weights. Therefore, the Taylor approximation can be used to locate the optimal value of the shape parameter (δ^*) using the procedure described in [3].

In fact, substituting $u(\mathbf{x}_i)$ by its Taylor series in the vicinity of $\mathbf{x}_1$, and the weights $\boldsymbol{\alpha}$ by Eq. (14) results in,

$$\sum_{i=1}^{6} \alpha_i u(\mathbf{x}_i) = u^{(2,0)}(\mathbf{x}_1) + u^{(0,2)}(\mathbf{x}_1) + \frac{\delta}{h}[\ 0.5065u^{(1,0)}(\mathbf{x}_1) + 0.1388u^{(0,1)}(\mathbf{x}_1)]$$
$$+ h[\ 3.88910\ {}^{2}u^{(3,0)}(\mathbf{x}_1) + 0.1317u^{(2,1)}(\mathbf{x}_1)$$
$$+ 0.3898u^{(1,2)}(\mathbf{x}_1) + 2.35910\ {}^{3}u^{(0,3)}(\mathbf{x}_1)] + \cdots$$

where $u^{(m,n)}$ denotes the partial derivative of function u with respect to x, m times and respect to y, n times. Thus, the local truncation error is approximated by,

$$\begin{aligned}\tau_6(\mathbf{x}_1) &= \left| \sum_{i=1}^{6} \alpha_i u(\mathbf{x}_i) \quad \Delta u(\mathbf{x}_1) \right| \\ &= \left| \frac{\delta}{h}[\ 0.5065u^{(1,0)}(\mathbf{x}_1) + 0.1388u^{(0,1)}(\mathbf{x}_1)] \right. \\ &\quad + h[\ 3.88910^{-2}u^{(3,0)}(\mathbf{x}_1) + 0.1317u^{(2,1)}(\mathbf{x}_1) \\ &\quad \left. + 0.3898u^{(1,2)}(\mathbf{x}_1) + 2.35910^{-3}u^{(0,3)}(\mathbf{x}_1)] + \cdots \right|\end{aligned} \tag{15}$$

This approximation to the local truncation error is represented in red in Fig. 3. Notice that there is very good agreement with the actual error (except for large values of δ).

Fig. 4 shows the corresponding results for $h=0.0002$. In this case, the minimum error occurs in the region of ill conditioning

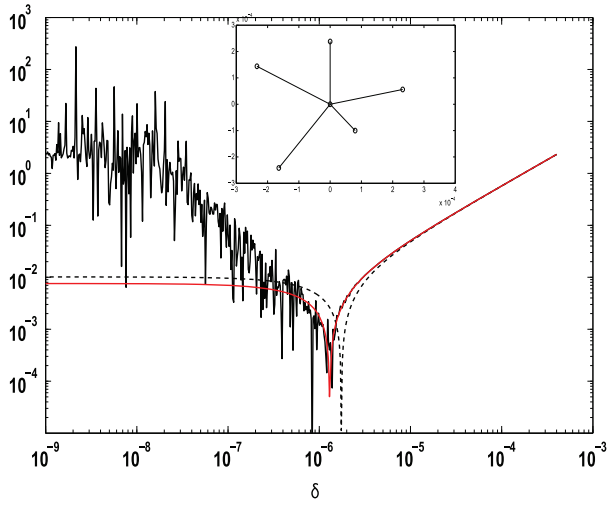

Fig. 4. Error in approximating the Laplacian of function (13) at $\mathbf{x}_1$ $(0,0)$ with n 6 non-equispaced RBF-FD formula. Solid line: numerically computed RBF-FD weights (h 0.0002). Dashed line: weights computed with Taylor series (14). Red line: Eq. (15). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

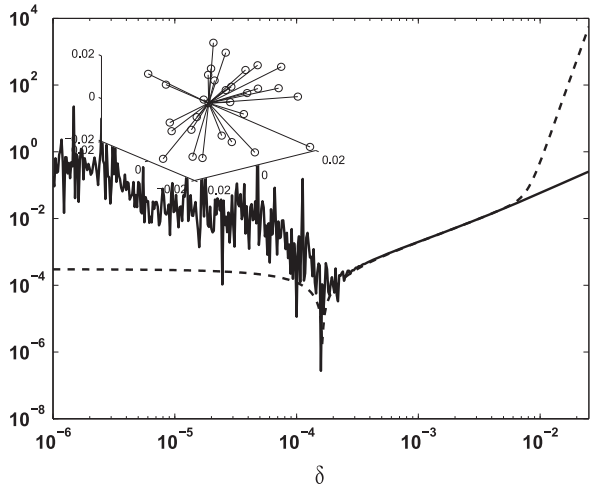

Fig. 5. Error in approximating the Laplacian of function (17) at $\mathbf{x}_1$ $(0,0,0)$ with n 31 random nodes as a function of δ. Solid line: numerically computed RBF-FD weights (h 0.01). Dashed line: weights from Laurent series for the inverse.

where only the Taylor series (14) can be used to compute the weights of the RBF FD formula.

In Figs. 3 and 4 one can clearly notice the existence of an optimal value of the shape parameter for which the local truncation error is zero to leading order. From (15) this optimal value δ^* corresponds to,

$$\frac{\delta^*}{h^2} = \frac{3.88910^{-2}u^{(3,0)} + 0.1317u^{(2,1)} + 0.3898u^{(1,2)} + 2.35910^{-3}u^{(0,3)}}{0.5065u^{(1,0)} + 0.1388u^{(0,1)}} \tag{16}$$

Substituting the partial derivatives for function (13) by their exact values at $\mathbf{x}_1 = (0,0)$ results in $\delta^* = h^2 * 32.5754$ ($\delta^* = 0.0032$ for $h=0.01$ and $\delta^* = 1.303 \times 10^{-6}$ for $h=0.0002$), which agrees with the location of the minimum error in Figs. 3 and 4.

In 3D, we approximate the Laplacian of function

$$u(\mathbf{x}) = \exp\left[\left(x - \frac{1}{4}\right)^2 \left(y - \frac{1}{2}\right)^2 \right]$$
$$\cos(2\pi y)\sin(\pi x)\cos(\pi z/2) \tag{17}$$

Fig. 5 shows the local approximation error in computing the Laplacian of function (17) using the RBF FD method with 31 random nodes. The RBF FD weights are computed numerically (solid line) and with the Laurent series of the inverse (dashed line). For $\delta < 3 \times 10^{-4}$, the RBF FD weights cannot be computed numerically due to ill conditioning. For $\delta \to 0$, the RBF FD weights approach standard finite difference weights, and the local approximation error approaches 3×10^{-4}.

4. Results

4.1. Equispaced nodes

Table 1 shows the leading order of the RBF FD weights for the Laplacian in 2D, *i.e.*, the standard finite difference weights, as a function of the number of nodes in the stencil n. We assume that the nodes are in a square lattice and that a stencil of n nodes uses the n nearest nodes in the lattice. Table 2 shows the first order correction to the RBF FD weights. We also show in the last row of these tables the order p of the singularity of the inverse. The order of the singularity is 2 for $n \in [5,6]$, 3 for $n \in [7,8]$, 4 for $n \in [9,14]$, 5 for $n \in [15,16]$, 6 for $n \in [17,22]$. Similarly in 3D the order of the

Table 1
Leading order RBF-FD weights for Laplacian ($\alpha_0 h^2$).

Node	n 5	n 6	n 7	n 8	n 9	n 10	n 13
(0,0)	−4	−4	−4	−4	−372/77	−372/77	−5
(0,1)	1	1	1	1	109/77	109/77	4/3
(1,0)	1	1	1	1	109/77	109/77	4/3
(0,−1)	1	1	1	1	109/77	109/77	4/3
(−1,0)	1	1	1	1	109/77	109/77	4/3
(1,1)		0	0	0	−16/77	−16/77	0
(1,−1)			0	0	−16/77	−16/77	0
(−1,−1)				0	−16/77	−16/77	0
(−1,1)					−16/77	−16/77	0
(0,2)						0	−1/12
(2,0)							−1/12
(0,−2)							−1/12
(−2,0)							−1/12
p	2	2	3	3	4	4	4

Table 2
First order RBF-FD weights for Laplacian ($\alpha_1 h^2/\delta$).

Node	n 5	n 6	n 7	n 8	n 9	n 10	n 13
(0,0)	−10/3	−10/3	−6	−26/3	−1217/205	−1354/209	−1502/173
(0,1)	5/6	5/6	13/6	13/6	−2717/1095	2147/604	1318/519
(1,0)	5/6	5/6	7/2	29/6	−2717/1095	397/346	1318/519
(0,−1)	5/6	5/6	13/6	29/6	−2717/1095	1380/457	1318/519
(−1,0)	5/6	5/6	5/6	13/6	−2717/1095	397/346	1318/519
(1,1)		0	−4/3	−4/3	−1391/1395	598/8421	92/173
(1,−1)			−4/3	−8/3	−1391/1395	−983/1344	92/173
(−1,−1)				−4/3	−1391/1395	−983/1344	92/173
(−1,1)					−1391/1395	598/8421	92/173
(0,2)						−643/601	−935/1038
(2,0)							−935/1038
(0,−2)							−935/1038
(−2,0)							−935/1038
p	2	2	3	3	4	4	4

singularity of the inverse is 2 for $n \in [7,10]$, 3 for $n \in [11,16]$, 4 for $n \in [17,23]$, 5 for $n \in [24,27]$.

Another relevant property of RBF FD formulas is the order s of the local truncation error (error $= O(h^s)$). For equispaced nodes in 2D, the order is $s=2$ for $n \in [5,12]$, $s=4$ for $n \in [13,28]$, $s=6$ for $n \in [29,52]$. In 3D the order of approximation for the laplacian is $s=2$ for $n \in [7,32]$, $s=4$ for $n \in [33,\ ]$. If only nodes located in the three axes are used, then the order of the approximation is 2 for $n=7$, and 4 for $n=13$.

Similar to what was done in Eq. (15), one can use the weights in Tables 1 and 2 to derive the dependence of the local approximation error on h, δ and the value of the function and its derivatives at $\mathbf{x}_1$. For instance, in the case of a thirteen node stencil the resulting error is

$$\tau_{13}(\mathbf{x}_1) = \frac{h^4}{90}[u^{(6,0)}(\mathbf{x}_1)+u^{(0,6)}(\mathbf{x}_1)] \quad h^2\delta[0.9008u^{(4,0)}(\mathbf{x}_1) \\ 0.5318u^{(2,2)}(\mathbf{x}_1)+0.9008u^{(0,4)}(\mathbf{x}_1)] \\ 4.6214\delta^2[u^{(2,0)}(\mathbf{x}_1)+u^{(0,2)}(\mathbf{x}_1)]+\cdots \quad (18)$$

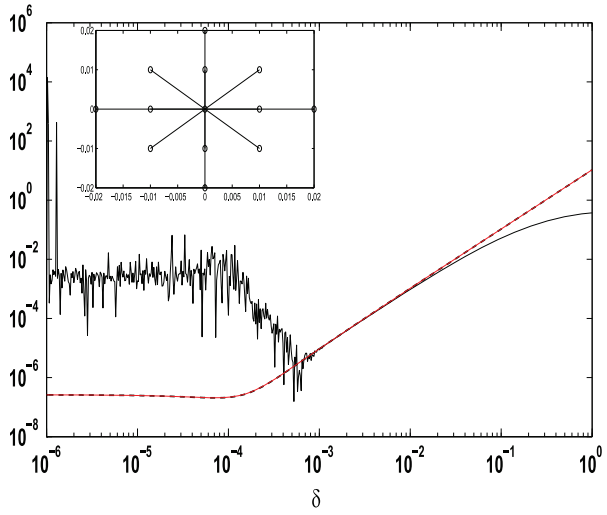

Fig. 6. Error in approximating the Laplacian of function (13) at $\mathbf{x}_1$ (0,0) with n 13 equispaced RBF-FD formula as a function of δ. Solid line: numerically computed RBF-FD weights (h 0.01). Dashed line: weights from Taylor series in Tables 1 and 2. Red line: Eq. (18). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

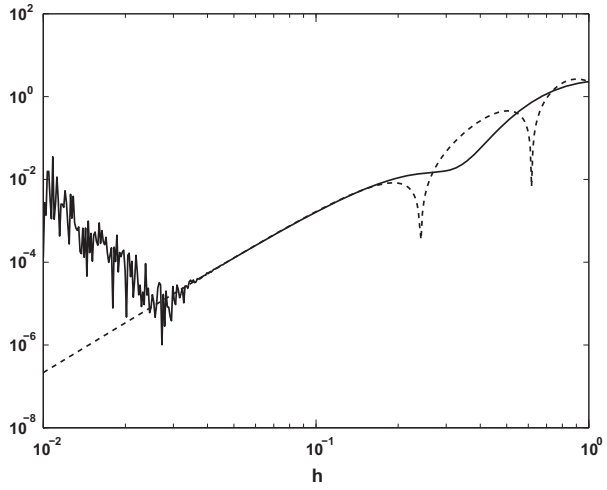

Fig. 7. Error in approximating the Laplacian of function (13) at $\mathbf{x}_1$ (0,0) with n 13 equispaced RBF-FD formula as a function of h. Solid line: numerically computed RBF-FD weights (h 0.01). Dashed line: weights from Taylor series in Tables 1 and 2.

In Ref. [3] the coefficients in this error formula were numerically computed by choosing appropriate polynomial functions and fitting the results to a power dependence with h and δ. The coefficients obtained in that way were 1/90, 0.93, 0.5 and 4.4, respectively, which are in pretty good agreement with the exact values shown in (18). Fig. 6 shows the local approximation error with the RBF FD weights computed numerically (solid line) and with the RBF FD computed from the three first terms of its Taylor series shown in Tables 1 and 2 (dashed line). We also show the error computed from Eq. (18). Notice that for $\delta < 10^{-3}$ the RBF FD weights cannot be computed numerically due to ill conditioning. Fig. 7 shows the error as a function of h. Notice that for $n=13$ the approximation error is $O(h^4)$ since all the terms is Eq. (18) are of order h^4.

4.2. Non equispaced nodes

For non equispaced nodes the order of the singularity of the inverse is 2 for $n \in [5,6]$, 3 for $n \in [7,10]$, 4 for $n \in [11,15]$, 5 for

$n \in [16, 21]$. These results can be easily inferred from the power dependence with ϵ of the eigenvalues of the A matrix, that were numerically computed in [12]. In 3D the order of the singularity of the inverse is 2 for $n \in [6, 10]$, 3 for $n \in [11, 20]$, 4 for $n \in [21, 35]$. The order of the local truncation error with respect to h in 2D is $s=1$ for $n \in [5, 9]$, $s=2$ for $n \in [10, 14]$, $s=3$ for $n \in [15, 20]$. The order of the local truncation error in 3D is $s=0$ for $n \in [6, 9]$, $s=1$ for $n \in [10, 19]$, $s=2$ for $n \in [20, 35]$.

The order of the singularity is related to the degree of the polynomial used to interpolate data in the stencil. For instance, in 2D with a polynomial of degree 2 one has to satisfy 6 conditions (coefficients of $1, x, y, x^2, xy, y^2$), and therefore with up to six node stencils a second degree polynomial is used which implies a singularity of order 2. A polynomial of degree 3 (10 conditions) allows interpolation in up to ten node stencils and, therefore, the order of the singularity is 3. Similarly, a second degree polynomial in 3D (ten conditions) implies singularity of order two for up to ten node stencils.

The order of the local truncation error is also related to the number of conditions that can be satisfied. For instance, in 2D with a nine node stencil it is not possible to satisfy all 10 conditions for interpolation with a second degree polynomial and, therefore, the order of the local truncation error is one, but with ten nodes the error becomes of order two since all ten conditions are satisfied. In general, the upper limit for the number of nodes corresponding to a certain singularity order is the lower limit for the order of the local truncation error.

Random nodes are rarely used in practical applications. In fact, there are different techniques which are used to create near uniform node distributions with irregular domains [9]. For instance, *Minimum Energy* (ME) nodes correspond to the minimum of the potential energy for electrostatic repulsion of scattered point charges. Thus, the nodes try to reach an hexagonal lattice distribution in which all the nodes are equidistant thereby minimizing the potential energy of the system.

Tables 3 and 4 show the leading order and first order correction to the RBF FD weights for the Laplacian using nodes distributed in

Table 3
Leading order RBF-FD weights for Laplacian ($\alpha_0 h^2$) with hexagonal lattice.

Node	n 7	n 9	n 11	n 13	n 15	n 17	n 19
(0,0)	−4	−4.541	−4.969	−5.079	−5.024	−5.333	−5.563
(1,0)	2/3	1.027	0.842	0.889	0.874	1	1.076
(0.5,0.866)	2/3	2/3	0.924	1.047	1.024	1	1.076
(−0.5,0.866)	2/3	2/3	0.924	0.846	0.825	1	1.076
(−1,0)	2/3	1.027	0.842	0.889	0.874	1	1.076
(−0.5,−0.866)	2/3	2/3	0.924	1.047	1.024	1	1.076
(0.5,−0.866)	2/3	2/3	0.924	0.846	0.825	1	1.076
(2,0)		−0.090	−0.076	−0.053	−0.061	0	0.038
(−2,0)		−0.090	−0.076	−0.053	−0.061	0	0.038
(0,1.732)			−0.128	−0.123	−0.093	−0.111	−0.187
(0,−1.732)			−0.128	−0.123	−0.093	−0.111	−0.187
(1.5,0.866)				−0.067	−0.028	−0.111	−0.187
(−1.5,−0.866)				−0.067	−0.028	−0.111	−0.187
(1,1.732)					−0.028	0	0.038
(−1,−1.732)					−0.028	0	0.038
(1.5,−0.866)						−0.111	−0.187
(−1.5,0.866)						−0.111	−0.187
(−1,1.732)							0.038
(1,−1.732)							0.038
p	3	4	4	4	5	5	6

Table 4
First order RBF-FD weights for Laplacian ($\alpha_1 h^2/\delta$) with hexagonal lattice.

node	n 7	n 9	n 11	n 13	n 15	n 17	n 19
(0,0)	−10/3	−7.894	−9.305	−9.335	−9.289	−9.746	−10.39
(1,0)	5/9	3.416	2.353	2.333	2.253	2.436	2.838
(0.5,0.866)	5/9	0.796	2.037	2.474	2.476	2.436	2.838
(−0.5,0.866)	5/9	0.796	2.037	1.798	1.647	2.436	2.838
(−1,0)	5/9	3.416	2.353	2.333	2.253	2.436	2.838
(−0.5,−0.866)	5/9	0.796	2.037	2.474	2.476	2.436	2.838
(0.5, −0.866)	5/9	0.796	2.037	1.798	1.647	2.436	2.838
(2,0)		−1.060	−0.836	−0.606	−0.684	0	0.480
(−2,0)		−1.060	−0.836	−0.606	−0.684	0	0.480
(0,1.732)			−0.940	−0.906	−0.625	−0.812	−1.586
(0,−1.732)			−0.940	−0.906	−0.625	−0.812	−1.586
(1.5,0.866)				−0.426	−0.085	−0.812	−1.586
(−1.5,−0.866)				−0.426	−0.085	−0.812	−1.586
(1,1.732)					−0.336	0	0.480
(−1,−1.732)					−0.336	0	0.480
(1.5,−0.866)						−0.812	−1.586
(−1.5,0.866)						−0.812	−1.586
(−1,1.732)							0.480
(1,−1.732)							0.480
p	3	4	4	4	5	5	6

an hexagonal lattice. The last row of the Tables shows the order of the singularity of the inverse. Notice that the symmetries of the node distribution result in an increase in the order of the singularity of the inverse with respect to the order of the singularity for random nodes. For instance, in the case of 19 nodes, the order of the singularity is six for hexagonally distributed nodes, and three for random nodes. Also the order of the local truncation error with respect to h increases: $s=2$ for $n \in [7, 15]$.

5. Conclusions

In this paper we show how the Laurent series for the inverse of the interpolation matrix can be effectively used to compute the weights of RBF FD formulas avoiding problems of ill conditioning. This technique is based on the computation of the Laurent series of the inverse of the RBF interpolation matrix in powers of the parameter $\delta = (\epsilon h)^2$, where h is the characteristic inter nodal distance and ϵ is the shape parameter. The matrix coefficients of this Laurent series are computed using a recursive algorithm described in [16] which is based on the method proposed by Avrachenkov and Lasserre [2]. The Laurent series is then used to derive the Taylor series for the weights in powers of δ. In this way, it is possible to compute the RBF FD weights for any differential operator exactly, without any rounding errors due to ill conditioning. In particular, the leading order term in the Taylor series corresponds to the standard finite difference weights.

The exponent of the leading order term of the Laurent series (the order of the singularity of the inverse) describes how the inverse approaches infinity as δ approaches zero. This result can be used to predict when the matrix becomes ill conditioned so that the weights cannot be computed directly by solving the corre sponding linear system numerically.

Finally, it should be pointed out that this method leads to analytical formulas for the weights in terms of h and δ that can be used to derive the exact dependence of the local truncation error in terms of these parameters, and the value of the function and its derivatives at the node. These formulas can then be used to derive an optimal value of the shape parameter for a given function and stencil.

The computational cost of the algorithm to obtain the Laurent series of the inverse, grows exponentially with the order of the singularity. As a result, the proposed technique is only feasible in a reasonable time for singularities of order seven or less, that is, 36 nonequispaced nodes or 24 equispaced nodes in 2D. It should be pointed out that in some applications RBF stencil sizes in the range 30 80 nodes have been common and these sizes are beyond the capabilities of the method. In those cases, other available algo rithms (Contour Padé, RBF QR, RBF GA, etc.) could be used.

The order of the singularity for a specified number of nodes decreases with increasing number of dimensions. Therefore, the proposed method is specially well suited for the solution of PDEs in 3D or higher dimensions with stencil sizes such that the order of the singularity is five or less.

Acknowledgments

We thank the reviewers for their useful comments and sugges tions. This work has been supported by Spanish MICINN Grants FIS2010 18473, FIS2013 41802 R, CSD2010 00011.

References

[1] Avrachenkov KE, Haviv M, Howlett PG. Inversion of analytic matrix functions that are singular at the origin. SIAM J Matrix Anal Appl 2001;22:1175–89.
[2] Avrachenkov KE, Lasserre JB. Analytic perturbation of generalized inverses. Linear Algebra Appl 2013;438:1793–813.
[3] Bayona V, Moscoso M, Carretero M, Kindelan M. RBF-FD formulas and convergence properties. J Comput Phys 2010;229:8281–95.
[4] Bayona V, Moscoso M, Kindelan M. Optimal constant shape parameter for multiquadric based RBF-FD method. J Comput Phys 2011;230:7384–99.
[5] Bayona V, Moscoso M, Kindelan M. Optimal variable shape parameter for multiquadric based RBF-FD method. J Comput Phys 2012;231:2466–81.
[6] Davydov O, Oanh DT. On the optimal shape parameter for Gaussian radial basis function finite difference approximation of the Poisson equation. Comput Math Appl 2011;62:2143–61.
[7] Driscoll TA, Fornberg B. Interpolation in the limit of increasingly flat radial basis functions. Comput Math Appl 2002;43:413–22.
[8] Emdadi A, Kansa EJ, Ali Libre N, Rahimian M, Shekarchi M. Stable pde solution methods for large multiquadric shape parameters. Comput Model Eng Sci 2008;25:23–42.
[9] Flyer N, Lehto E. Rotational transport on a sphere: local node refinement with radial basis functions. J Comput Phys 2010;229:1954–69.
[10] Fornberg B, Wright GB, Larsson E. Some observations regarding interpolants in the limit of flat radial basis functions. Comput Math Appl 2004;47:37–55.
[11] Fornberg B, Wright GB. Stable computation of multiquadric interpolants for all values of the shape parameter. Comput Math Appl 2004;48:853–67.
[12] Fornberg B, Zuev J. The Runge phenomenon and spatially variable shape parameters in RBF interpolation. Comput Math Appl 2007;54:379–98.
[13] Fornberg B, Piret C. A stable algorithm for flat radial basis functions on a sphere. SIAM J Sci Comput 2008;30:60–80.
[14] Fornberg B, Larsson E, Flyer N. Stable computations with Gaussian radial basis functions. SIAM J Sci Comput 2011;33:869–92.
[15] Fornberg B, Lehto E, Powell C. Stable calculation of Gaussian-based RBF-FD stencils. Comput Math Appl 2013;65:627–37.
[16] Gonzalez-Rodriguez P, Moscoso M, Kindelan M. Laurent expansion of the inverse of perturbed, singular matrices, submitted for publication; 2014.
[17] Huang C-S, Lee C-F, Cheng AH-D. Error estimate, optimal shape factor, and high precision computation of multiquadric collocation method. Eng Anal Bound Elem 2007;31:614–23.
[19] Larsson E, Fornberg B. Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions. Comput Math Appl 2005;49:103–30.
[20] Larsson E, Lehto E, Heryudono E, Fornberg B. Stable computation of differ- entiation matrices and scattered node stencils based on Gaussian radial basis functions, 2013;35:A2096–A2119.
[21] Ali Libre N, Emdadi A, Kansa EJ, Rahimian M, Shekarchi M. A stabilized rbf collocation scheme for neumann type boundary value problems. Comput Model Eng Sci 2008;24:61–80.
[22] Schaback R. Multivariate interpolation by polynomials and radial basis functions. Constr Approx 2005;21:293–317.
[23] Schönauer W, Adolph T. How we solve PDEs. J Comput Appl Math 2001;131:473–92.
[24] Shu C, Ding H, Yeo KS. Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier-Stokes equations. Comput Methods Appl Mech Eng 2003;192:941–54.
[25] Tolstykh AI, Shirobokov DA. On using radial basis functions in a "finite difference mode" with applications to elasticity problems. Comput Mech 2003;33:68–79.
[26] Wright GB. Radial basis function interpolation: numerical and analytical developments [Ph.D. thesis]. University of Colorado, Boulder; 2003.
[27] Wright GB, Fornberg B. Scattered node compact finite difference-type for- mulas generated from radial basis functions. J Comput Phys 2006;212:99–123.