

On the role of polynomials in RBF-FD approximations:

III. Behavior near domain boundaries

Víctor Bayona *

Departamento de Matemáticas, Universidad Carlos III de Madrid,
28911 Leganés, Madrid, Spain

Natasha Flyer †

Analytics and Integrated Machine Learning,
National Center for Atmospheric Research, Boulder, CO 80305, USA

Bengt Fornberg ‡

Department of Applied Mathematics,
University of Colorado, Boulder, CO 80309, USA

November 28, 2018

Abstract

Radial basis function generated finite difference (RBF-FD) approximations generalize grid-based regular finite differences to scattered node sets. These become particularly effective when they are based on polyharmonic splines (PHS) augmented with multi-variate polynomials (PHS+poly). One key feature is that high orders of accuracy can be achieved without having to choose an optimal shape parameter and without having to deal with issues related to numerical ill-conditioning. The strengths of this approach were previously shown to be especially striking for approximations near domain boundaries, where the stencils become highly one-sided. Due to the polynomial Runge phenomenon, regular FD approximations of high accuracy will in such cases have very large weights well into the domain. The inclusion of PHS-type RBFs in the process of generating weights makes it possible to avoid this adverse effect. With that as motivation, this study aims at gaining a better understanding of the behavior of PHS+poly generated RBF-FD approximations near boundaries, illustrating it in 1-D, 2-D and 3-D.

Keywords: Radial basis functions, RBF, RBF-FD, cubic polyharmonic splines, PHS, Runge's phenomenon.

* Email: victor.bayona.revilla@gmail.com

† Email: flyer@ucar.edu

‡ Email: fornberg@colorado.edu

1 Introduction

RBF-FD (radial basis function generated finite difference) approximations generalize regular FD approximations from grids to scattered node layouts, vastly improving the geometric flexibility, including ease of carrying out local refinement as may be needed due the character of PDE solutions [9, 10]. Some recent works [1, 3, 6, 7] have found that the combination of PHS-type RBFs with high degree polynomials (PHS+poly) offers a fresh alternative for creating RBF-FD formulas. One key feature of this approach is that high orders of accuracy can be achieved without the need of selecting a shape parameter or the issues related to numerical ill-conditioning. Moreover, it was noted in [3] that it offers a major advantage over classical FD when approximating a PDE near a domain boundary. Classical one-sided FD approximations will, for high orders of accuracy, feature weights that are large and oscillatory well into the domain. With PHS+poly based RBF-FD, this can easily be avoided. For instance, this feature was utilized in [3] for solving elliptic PDEs without needing any special boundary treatment (such as “ghost nodes”), either for accuracy or for obtaining linear systems that are well suited for iterative solvers; and also in [13] for numerical quadrature. Weights for classical 1-D FD stencils vs. for PHS+poly based RBF-FD stencils were tabulated and illustrated graphically in [3], but no explanations were then available for why these boundary improvements occurred. The mechanism underlying this feature is formally explained in [2]. In that work it was found that RBF interpolation augmented with polynomials (RBF+poly) approximations can be written as pure RBF approximations plus a correction term, which accounts for the polynomial reproduction of the combined RBF+poly interpolant. This correction mixes RBFs and polynomials, and its magnitude decreases for increasingly large stencil sizes. If the RBFs display a non-oscillatory behavior near the boundary (such as is the case low degree PHS), this is inherited by the RBF+poly interpolant. Supplementing the formal explanations in [2], the present study focuses on heuristic perspectives and numerical demonstrations about the nature of the interpolant near boundaries.

Section 2 begins by recalling the formulas for PHS+poly interpolants and for the RBF-FD weights these give rise to. This is followed by the key empirical observations from [2] regarding PHS+poly approximations near stencil boundaries. Section 3 provides two approaches towards understanding these previously mostly empirical observations. Section 3.1 gives some heuristic arguments and Section 3.2 analyzes the differences between pure RBF and RBF+poly interpolants. Sections 4 and 5 provide a variety of illustrations, showing that the results in 1-D carry over essentially unchanged to 2-D and to 3-D, respectively. Finally, the conclusions are summarized in Section 6.

2 Some background observations

2.1 RBF-FD discretizations

The combination of polyharmonic splines (PHS) with polynomials has a quite long history, as summarized in the Introductions of [3, 7], as well as in the monographs [5, 9]. The early literature concerned global RBF approximations, in which a single interpolant of the form

$$s(\underline{x}) = \sum_{k=1}^N \lambda_k \phi(\|\underline{x} - \underline{x}_k\|) + \sum_{k=1}^{\binom{l+d}{l}} \beta_k p_k(\underline{x}) \quad (1)$$

with matching constraints

$$\sum_{k=1}^N \lambda_k p_j(x_k) = 0, \quad j = 1, 2, 3, \dots, \binom{l+d}{l} \quad (2)$$

was applied over all the N node locations in the full domain of interest. Here, $\|\cdot\|$ denotes the standard Euclidean norm, d is the number of space dimensions and $p_k(\underline{x})$ are all the independent multivariate polynomials up through total degree l . The PHS class of radial functions is defined by $\phi(r) = r^m$ if m is odd and $\phi(r) = r^m \log r$ if m is even.

For cases that are relevant in applications, the total number of nodes (N -values) will be at least in the thousands. The single polynomial in (1) can then assist in providing non-singularity when solving for the interpolation coefficients λ_k and β_k , but will otherwise play no significant role in the actual approximation. With the RBFs doing this, their smoothness will determine the accuracy of the interpolation.

In the context of RBF-FD, each of the N nodes in turn form the “center” of a separate RBF-FD stencil containing $n \ll N$ nodes, typically chosen as the closest ones to the “center node”. Applying (1) and (2) separately over the n nodes within each of these N stencils, there will be a total of N different polynomials. Applied only locally, these polynomials become well suited to dominate the approximations within each stencil. Under refinement (i.e. with n fixed while N is increased), the polynomials alone determine the order of accuracy (c.f. [2, 3, 7]), while the RBFs assist towards non-singularity. In this regard, the roles of the RBFs and the polynomials have been reversed compared to the global RBF case.

Given a linear operator L , we calculate for each RBF-FD stencil its weights w_k , $k = 1, 2, \dots, n$. In the special case of 2-D and including up through linear polynomials, the weights are found by solving

$$\begin{bmatrix} & 1 & x_1 & y_1 \\ A & \vdots & \vdots & \vdots \\ & 1 & x_n & y_n \\ - & - & + & - & - & - \\ 1 & \cdots & 1 & | & & \\ x_1 & \cdots & x_n & | & 0 \\ y_1 & \cdots & y_n & | & & \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_n \\ - \\ \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{bmatrix} = \begin{bmatrix} L\phi(\|\underline{x} - \underline{x}_1\|)|_{\underline{x}=\underline{x}_c} \\ \vdots \\ L\phi(\|\underline{x} - \underline{x}_n\|)|_{\underline{x}=\underline{x}_c} \\ - \\ L 1|_{\underline{x}=\underline{x}_c} \\ L x|_{\underline{x}=\underline{x}_c} \\ L y|_{\underline{x}=\underline{x}_c} \end{bmatrix}. \quad (3)$$

A derivation is given in [9], Section 5.1.4. In more concise form, linear systems of this type can be written as

$$\begin{bmatrix} A & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} \underline{w} \\ \underline{\gamma} \end{bmatrix} = \begin{bmatrix} L\phi(\underline{x}) \\ L\underline{p}(\underline{x}) \end{bmatrix}. \quad (4)$$

Key to much of the following discussion is the fact that exactly the same linear systems arise also in the different context of seeking the solution (via Lagrange multipliers) to the minimization problem

$$\min_{\underline{w}} J(\underline{w}) = \frac{1}{2} \underline{w}^T A \underline{w} - \underline{w}^T L \underline{\phi}(\underline{x}) \quad \text{subject to} \quad P^T \underline{w} = L \underline{p}(\underline{x}). \quad (5)$$

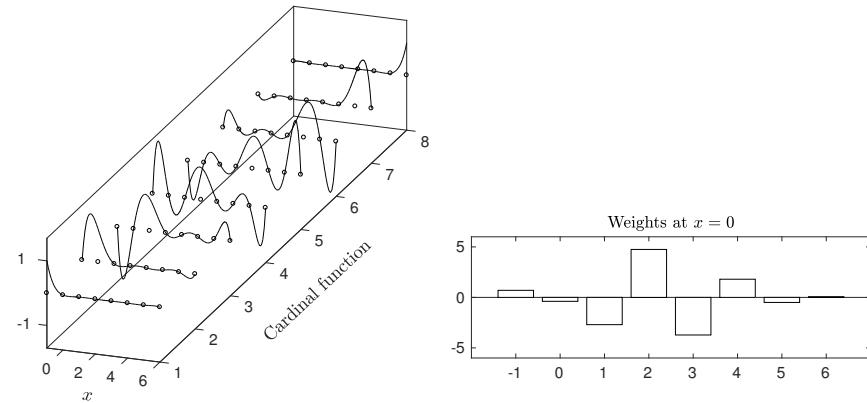
With a PHS-type radial function and the given constraint on the \underline{w} -vector, the quadratic form $\underline{w}^T A \underline{w}$ becomes positive definite, ensuring that $J(\underline{w})$ has a unique minimum. As a consequence, (4) is generally well conditioned (since \underline{w} solves a minimization problem with more variables than constraints).

2.2 Key previous observations

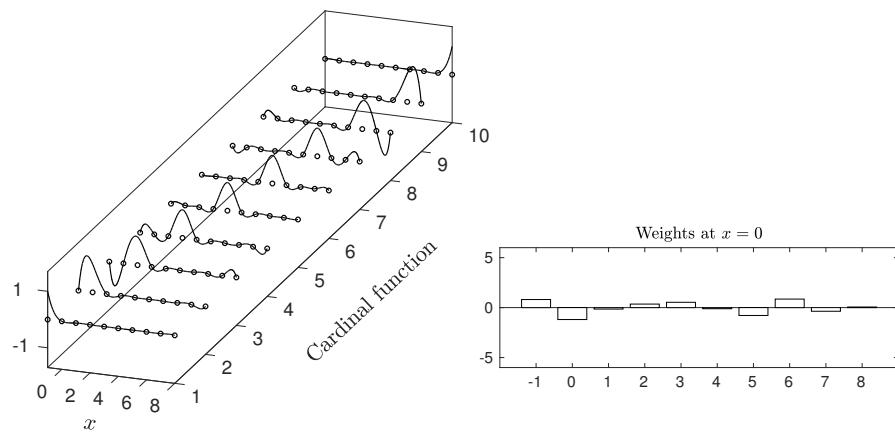
It was found in [3] that RBF+poly approximations can overcome the edge oscillations by simply increasing the stencil size for a fixed polynomial degree. Figure 5 in [3] (displayed here as Figure 1) shows the 1-D cardinal functions using cubic PHS augmented with 7th degree polynomials over the stencil $x = -1, 0, 1, 2, \dots, n-2$, for $n = 8, 10$ and 16 , together with the RBF-FD weights for approximating the second order derivative (which are equal to the second derivative of the successive cardinal functions) at $x = 0$. The following two cases are distinguished as a function of the stencil size:

- **Pure polynomial case:** For $n = 8$, the RBF-FD weights coincide with standard FD weights. Observe that the largest weights are in the middle of the stencil, well away from the evaluation point $x = 0$. Table 1 lists the traditional FD weights for increasing degrees $l = n - 1$ of the polynomials used in deriving them. As l is increased, the weights near the stencil center grows exponentially fast in magnitude, while the ones in the region of interest (i.e. mainly the leftmost three columns) become less significant. This version is unacceptable in the context of solving an elliptic equation due to stability issues.
- **Case with RBFs present:** Figure 1 also shows the result of keeping the l -value fixed while increasing the stencil size from $n = 8, 10$ and 16 . In this case, the RBFs will then come to play an increasing role in the process of determining the stencil weights while the formal accuracy order is maintained, as determined by the degree $l = 7$ polynomials. We recognize the top row in Table 2 as row 6 in Table 1 - corresponding to using polynomials only. The following rows show that, when including increasingly many RBFs (while maintaining the same l value), the pattern of weights quickly reverts itself to the ideal case of the weights at the leftmost nodes dominating over all others. By somewhat resembling the pattern $[1, -2, 1]$ at the three leftmost nodes, $x = -1, 0, 1$, and being smaller everywhere else, they offer similar stability features (in the context of solving elliptic PDEs) as the second-order $n = 3$ classical finite difference approximation, yet at a much higher accuracy.

(a) PHS r^3 + poly cardinal functions; $n = 8$ and $l = 7$.



(b) PHS r^3 + poly cardinal functions; $n = 10$ and $l = 7$.



(c) PHS r^3 + poly cardinal functions; $n = 16$ and $l = 7$.

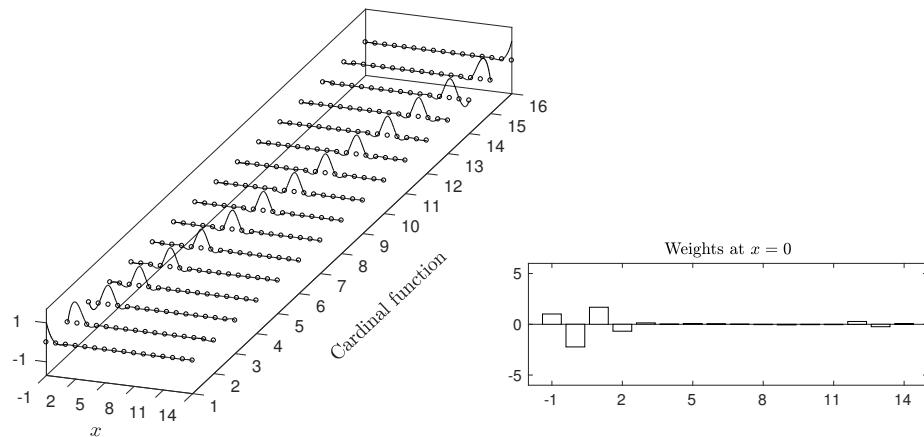


Figure 1: Cardinal functions and associated weights for approximating the second derivative at $x = 0$ when using polynomials of degree $l = 7$ together with cubic PHS, and increasing number of stencil nodes n .

Table 1: Standard FD weights (rows) approximating $u''(x)$ at $x = 0$ over the points $x = -1, 0, 1, 2, \dots, n-2$ (columns) for increasing degrees $l = 2$ (first row), $l = 3$ (second row), \dots , $l = 9$ (last row) of polynomials and stencil sizes $n = l + 1$.

$x =$	-1	0	1	2	3	4	5	6	7	8
	1.0000	-2.0000	1.0000							
	1.0000	-2.0000	1.0000	0						
	0.9167	-1.6667	0.5000	0.3333	-0.0833					
	0.8333	-1.2500	-0.3333	1.1667	-0.5000	0.0833				
	0.7611	-0.8167	-1.4167	2.6111	-1.5833	0.5167	-0.0722			
	0.7000	-0.3889	-2.7000	4.7500	-3.7222	1.8000	-0.5000	0.0611		
	0.6482	0.0254	-4.1500	7.6500	-7.3472	4.7000	-1.9500	0.4754	-0.0518	
	0.6040	0.4236	-5.7429	11.3667	-12.9222	10.2750	-5.6667	2.0683	-0.4500	0.0442

6

Table 2: Cubic PHS RBF-FD weights (rows) augmented with polynomials up to 7th degree, approximating $u''(x)$ at $x = 0$ over the points $x = -1, 0, 1, 2, \dots, n-2$ (columns) for increasing stencil sizes $n = 8, 9, \dots, 16$ (rows).

$x =$	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	0.7000	-0.3889	-2.7000	4.7500	-3.7222	1.8000	-0.5000	0.0611								
	0.7591	-0.8617	-1.0450	1.4400	0.4152	-1.5100	1.1550	-0.4117	0.0591							
	0.8081	-1.1947	-0.1461	0.3524	0.5318	-0.1117	-0.7871	0.8534	-0.3654	0.0592						
	0.8513	-1.4552	0.4419	-0.1590	0.4558	0.0910	-0.1708	-0.4492	0.6677	-0.3332	0.0597					
	0.8901	-1.6661	0.8493	-0.4110	0.3399	0.1458	-0.0158	-0.1595	-0.2643	0.5394	-0.3079	0.0602				
	0.9252	-1.8416	1.1460	-0.5448	0.2554	0.1197	0.0713	-0.0712	-0.1199	-0.1605	0.4468	-0.2870	0.0606			
	0.9572	-1.9901	1.3696	-0.6178	0.1977	0.0817	0.0913	0.0023	-0.0761	-0.0866	-0.0976	0.3766	-0.2688	0.0607		
	0.9866	-2.1177	1.5426	-0.6582	0.1606	0.0453	0.0889	0.0318	-0.0211	-0.0708	-0.0591	-0.0589	0.3220	-0.2528	0.0606	
	1.0137	-2.2283	1.6794	-0.6803	0.1376	0.0154	0.0772	0.0427	0.0073	-0.0323	-0.0599	-0.0386	-0.0343	0.2785	-0.2383	0.0603

3 Approximations near boundaries in 1-D

In order to gain insights into the key previous observations of PHS+poly near boundaries in 1-D, in this section we use two distinct ways to split the interpolant $s(\underline{x})$ into two parts. In Section 3.1, we consider the weights set $\{\lambda_k\}$ and $\{\beta_k\}$ from (1), while in Section 3.2 (following [2]) we instead split $s(\underline{x})$ into a pure RBF interpolant and a ‘correction’ term that includes both RBFs and polynomials.

3.1 Heuristic arguments

3.1.1 Character of one-sided PHS+poly weight sets

Figure 2 illustrates again the weights at $x = -1, 0, 1, 2, \dots$ when approximating $u''(0)$, first with (a) regular FD formulas of increasing orders, and with (b) the order of accuracy ‘locked in’ at 6, but with RBF-FD stencils (using $\phi(r) = r^3$) extending still further into the domain. While one-sided FD approximations feature weights that grow exponentially fast with the order of accuracy, also including PHS-type RBFs makes the weights revert back towards the ideal case (near diagonal dominance, and nearly vanishing away from the boundary).

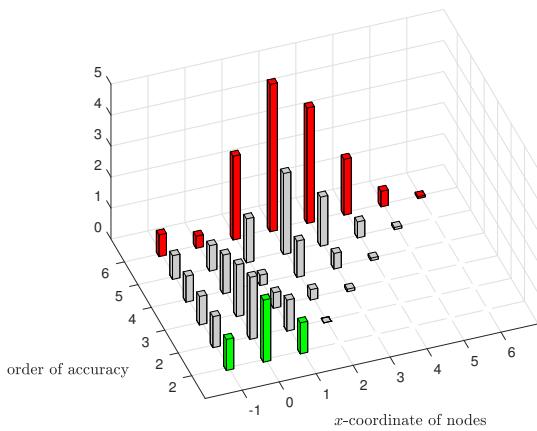
This shown behavior becomes plausible if one first considers the weight set $[1, -2, 1, 0, 0, \dots, 0]$. This is exact for $u''(0)$ in cases of the functions $1, x, x^2, x^3$, while we for accuracy of order $l + 1$ need it to be exact also for x^4, x^5, \dots, x^l . From the minimization property given in (5), the question becomes how this weight set can be adjusted to accommodate these further powers of x while staying as small as possible. Since all these further test functions grow rapidly as one moves out to the right, that will therefore be the location where the ‘corrections’ can be achieved with the smallest adjustments in the weights. The resulting stencils will thus be close to $[1, -2, 1]$ at their left, and with only small ‘corrections’ far to the right (decreasing in size as the stencil width increases), as seen in Figure 2 (b) and Table 2.

3.1.2 Character of PHS+poly cardinal functions

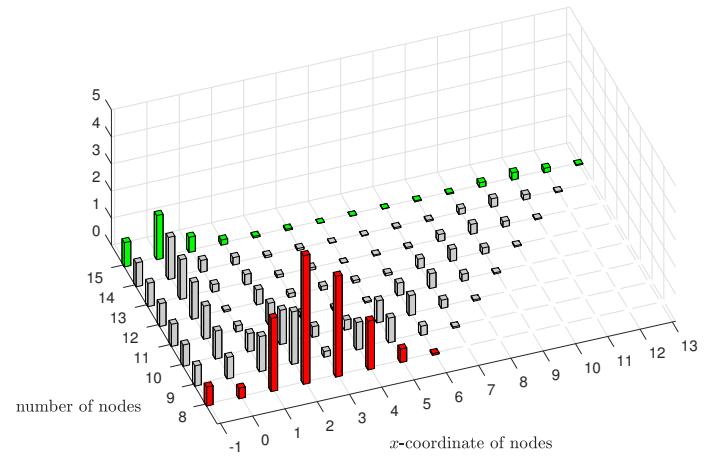
While the argument above in Section 3.1.1 quickly explains the key features of the RBF-FD weights w_k , $k = 1, 2, \dots, 16$, seen in Figure 1, it gives additional insights to arrive at the same result also via considering the coefficients λ_k and β_k in (1) (for RBF interpolation). In this case, the interpolant is a combination of RBFs and polynomials of degree l . Neither alone interpolates the data - but their sum does.

Figure 1 illustrated how differentiation weights can be read off from the corresponding cardinal functions. As the front and back curves in its top subplot showed, polynomials alone provide reasonable cardinal functions if the cardinal point is at or near one of the end points of the interval, but they behave very badly if the cardinal point is located towards the interior. PHS (say $\phi(r) = r^3$) alone will, on the other hand, provide excellent cardinal functions throughout the domain interior, as seen in the left column in Figure 4. In the present case, they then become cubic splines. For the last cardinal function display of Figure 1, equation (1) becomes

$$s(x) = \sum_{k=1}^{16} \lambda_k \phi(\|x - x_k\|) + \sum_{k=0}^7 \beta_k x^k,$$



(a) Lines 1-6 of Table 1



(b) Lines 1-8 of Table 2

Figure 2: Magnitude of the weights in the leading lines of Tables 1 and 2, respectively. The back row in part (a) is identical to the front row in part (b).

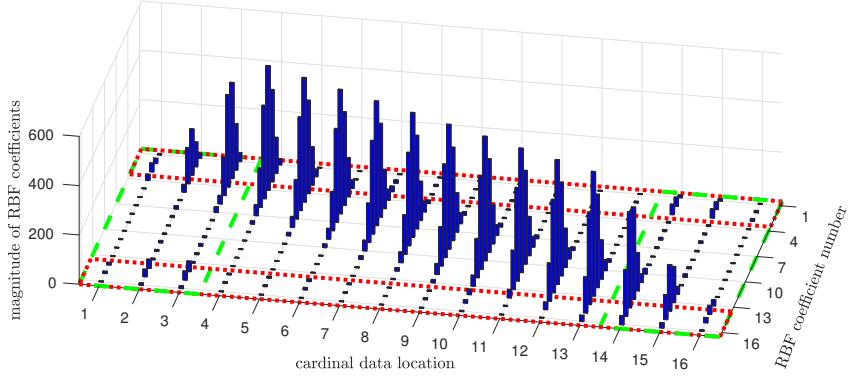
with the constraints

$$\sum_{i=1}^{16} \lambda_i x_i = 0, \quad k = 0, 1, 2, \dots, 7.$$

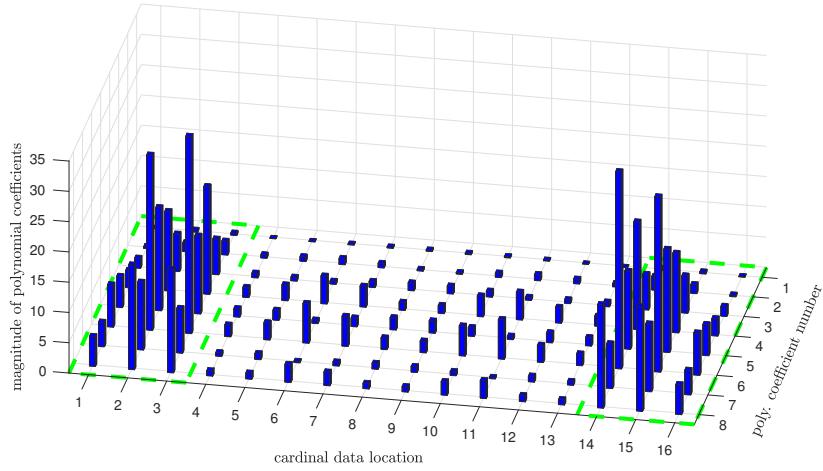
Nothing fundamental changes if we let the x -interval be $[-1, 1]$ instead of $[-1, 14]$. The last two constraints will then roughly become $\lambda_1(-1)^k + \lambda_{16}(+1)^k = 0$, with $k = 6, 7$, i.e. $\lambda_1 + \lambda_{16} \approx 0$ and $\lambda_1 - \lambda_{16} \approx 0$, together telling that $\lambda_1 \approx \lambda_{16} \approx 0$. The $k = 4, 5$ constraints similarly suggest that $\{\lambda_2, \lambda_{15}\}$ and (to a lesser extent) $\{\lambda_3, \lambda_{14}\}$ will also be smaller than the remaining ones. The constraint equations will consequently leave the more central RBF coefficients less constrained. The approximation throughout the interior can thus be expected to still be ‘spline-like’. This observation is independent of the data we interpolate, and will thus hold for each one of the cardinal functions.

Figure 3 shows two arrays (in subplots (a) and (b), respectively). In each of these, the 16 columns correspond in turn to the 16 cardinal functions, as seen in Figure 1. Part (a) shows (in magnitude) the RBF coefficients $\lambda_1, \lambda_2, \dots, \lambda_{16}$ and Part (b) similarly the polynomial coefficients. The red dashed rectangles in Part (a) indeed show reduced values for the lowest and highest indexed RBF coefficients. This suppresses the RBF contributions to the first and last cardinal functions (with coefficients inside the green dashed rectangles). The green rectangles in Part (b) of the figure show how the polynomials then ‘step in’ (to ensure the cardinal data requirements are met). These polynomials can quite well handle these first and last few cardinal functions, but will (due to the general nature of high degree polynomials) also feature some oscillations at the opposite end of the interval. Hence, the RBFs will, although with small coefficients, have to do some corrections for this. We see this as somewhat elevated coefficients in the top right and bottom left in Part (a). This in turn explains the slight ‘ridge’ visible along the diagonal in Figure 2 (b) (with the rest of that figure a consequence of the excellent spline-dominated cardinal functions).

The character of RBF coefficients for cardinal data on an infinite equispaced grid was discussed in



(a) RBF coefficients $\lambda_1, \lambda_2, \dots, \lambda_{16}$ for each of the 16 cardinal functions.



(b) Polynomial coefficients $\beta_0, \beta_1, \dots, \beta_7$ for each of the 16 cardinal functions.

Figure 3: Magnitudes of the coefficients in equation (1) when $s(x)$ in turn is chosen as the cardinal functions displayed in Figure 1. (a) RBF coefficients $\lambda_1, \lambda_2, \dots, \lambda_{16}$. (b) Polynomial coefficients $\beta_0, \beta_1, \dots, \beta_7$.

[11]. In particular, the top line of Table 2 in that reference tells that, for cubic RBFs $\phi(r) = r^3$, the coefficients for cardinal 1-D data decays with the distance from the cardinal point approximately as $(-1)^k \cdot 5.20 \cdot e^{-1.32k}$ when k nodes away from the cardinal one. This fits very well with what we see in the central columns of Figure 3 (a).

In summary, the excellent form of the PHS+poly cardinal functions seen in Figure 1 is caused by the first and last few being mainly of polynomial nature while the remaining ones essentially are spline-based. The presence of the polynomial constraints will nevertheless cause the resulting RBF-FD approximation to feature a formal order of accuracy that corresponds to the polynomial degree.

3.2 Approach based on closed form expressions

3.2.1 RBF+poly cardinal functions and RBF-FD weights

In this section we summarize the results found in [2], which are based on finding closed form expressions for the RBF+poly interpolant satisfying the collocation conditions $s(x_j) = f_j$, $j = 1, \dots, N$, by simply solving the linear system of equations (1) and (2). The reader is refer to this work for further details.

With the same notation ~~than-as~~ in (4), the $\{\lambda_k\}$ and $\{\beta_k\}$ coefficients are, provided that A and P are full rank, equal to

$$\underline{\lambda}^T = \underline{f}^T (I - W P^T) A^{-1} \quad (6)$$

and

$$\underline{\beta}^T = \underline{f}^T W, \quad (7)$$

where $W = A^{-1} P (P^T A^{-1} P)^{-1}$. As we will show in the following sections, this matrix W plays a key role in the approximation, as the columns of the matrix contain the weights \underline{w} that exactly differentiate the augmented multivariate polynomial terms $p_k(\underline{x})$, $k = 1, \dots, \binom{l+d}{l}$.

Therefore, the RBF+poly interpolant (1) can be written as

$$s(\underline{x}) = \underline{f}^T [I - WP^T] \hat{\psi}(\underline{x}) + \underline{f}^T W \underline{p}(\underline{x}), \quad (8)$$

where $\hat{\psi}(\underline{x}) = A^{-1} \underline{\phi}(\underline{x})$ are the pure RBF cardinal functions. From this result, follows the exact expressions for the RBF+poly cardinal functions

$$\underline{\psi}(\underline{x}) = \hat{\psi}(\underline{x}) - W [P^T \hat{\psi}(\underline{x}) - \underline{p}(\underline{x})]. \quad (9)$$

Observe these are equal to the RBF cardinal functions $\hat{\psi}(\underline{x})$ that, by itself, interpolates the data, plus a ‘correction’ term

$$\underline{C}(\underline{x}) = -W [P^T \hat{\psi}(\underline{x}) - \underline{p}(\underline{x})], \quad (10)$$

containing both RBFs and polynomial terms. This is zero at all the collocation points, and accounts for the reproduction of the augmented polynomial terms. Thereby, the exact expressions of the RBF-FD weights (4) approximating a linear differential operator L are

$$\underline{w} = L \underline{\psi}(\underline{x}) = L \hat{\psi}(\underline{x}) + L \underline{C}(\underline{x}). \quad (11)$$

These results will be used in the following sections to gain a better understanding of the behavior of PHS+poly based RBF-FD approximations near boundaries.

3.2.2 Explanation to key previous observations

According to equation (9), RBF approximations combined with polynomials can be split into the contribution of pure RBF cardinal functions $\hat{\psi}(\underline{x})$ and a correction term $\underline{C}(\underline{x})$. For the 1-D case displayed in Figure 1 for cubic PHS augmented with 7th degree polynomials, $\hat{\psi}(x)$ and $\underline{C}(x)$ are shown in Figure (4). Observe cubic PHS cardinal functions (left column) do not feature edge oscillations. These are introduced in the PHS+poly approximation by the polynomial components of the correction term (right column). As the stencil size increases, however, the correction term required to enforce the polynomial reproduction property decreases, and $\underline{\psi}(x)$ converges to $\hat{\psi}(x)$ as shown in [2]. This decrease is the key to many of the observations in this study.

The contribution to the RBF-FD weights approximating the second derivative at $x = 0$ are also displayed in the figure, illustrating the stability recovery (elliptic PDEs) for increasingly large stencil sizes. Observe the weights for cubic PHS resembles somehow the pattern $[1, -2, 1]$ centered at the evaluation point $x = 0$. But these are not exact for polynomials and require the action of the correction term. For $n = 8$, the correction term features large values at the stencil center as in standard FD. However, its magnitude decreases as a function of n , leading for $n = 16$ to exact RBF-FD weights (for polynomials up to 7th degree) with a pattern $[1, -2, 1]$ inherited from cubic PHS.

The stability of an interpolation method can be visualized by the Lebesgue function, defined as

$$\Lambda_n(\underline{x}) = \sum_{i=1}^n |\psi_i(\underline{x})|. \quad (12)$$

Typically, the Lebesgue function of a method free of edge anomalies will feature low amplitude oscillations. Figure 5 displays the Lebesgue functions for r^3 (left) and r^7 (right) augmented with all polynomial terms up to 7th degree, as a function of the stencil size n (y -axis) for $n = 8, 10$ and 16 . For comparison purposes, the Lebesgue functions for PHS approximations without polynomial augmentation are also displayed in the figure.

In contrast to r^7 -type RBFs, cubic PHS approximations do not feature edge oscillations. This is related to the fact that cubic spline with “natural” edge conditions minimize $\int(s''(x))^2 dx$ over all possible interpolants, as pointed out in [12]. However, when combined with polynomials, the polynomial nature of the correction term introduces edge oscillations. For $n = 8$, the approximation is totally polynomial (as the degree satisfy $l = n - 1$) and both $r^3+\text{poly}$ and $r^7+\text{poly}$ feature equivalently large edge oscillations. As the stencil size increases, the correction term required to make the approximation exact for the augmented degrees decrease (as shown in Figure 4), and the Lebesgue functions recover the PHS dominated edge behavior. Specially interesting is the case $r^3+\text{poly}$ since the approximation can reproduce polynomials up to the augmented degree l (in this case $l = 7$) without the adverse Runge Phenomena effects.

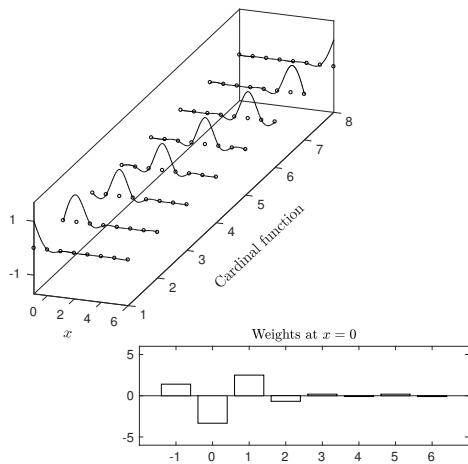
3.2.3 Accuracy near edges

The behavior of global RBF approximations near boundaries was analyzed in [8], comparing the performance of different strategies to handle the boundary in both 1-D and 2-D. The strategies considered included adding polynomial terms to a cubic PHS interpolant. Through the numerical approximation of $f(x) = \arctan(5(x + 1/2))$ using 11 equispaced nodes over $[-1, 1]$, the authors showed that, if the function is not well resolved, increasing the polynomial degree damages the

cubic PHS cardinal functions

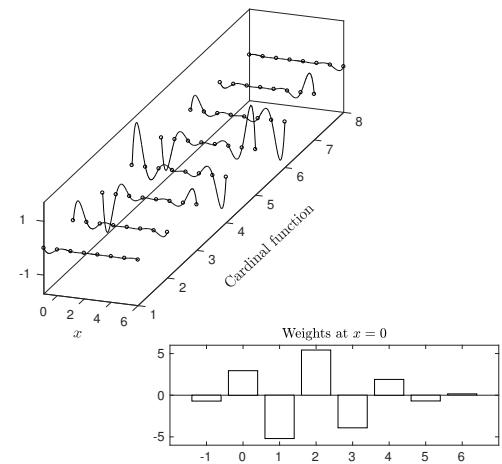
$$\hat{\psi}(x)$$

$$n = 8$$

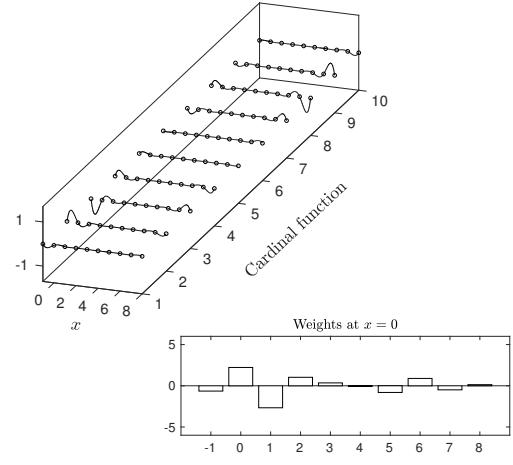
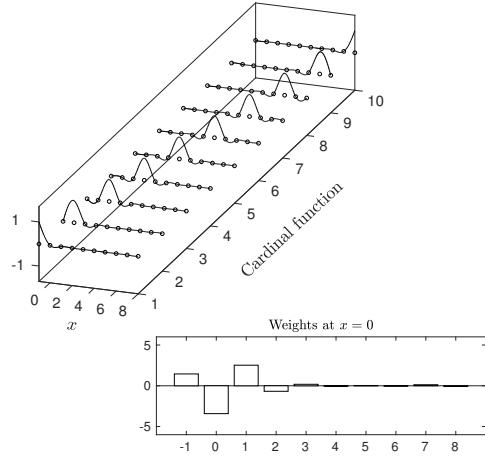


Correction term

$$\underline{C}(x)$$



$$n = 10$$



$$n = 16$$

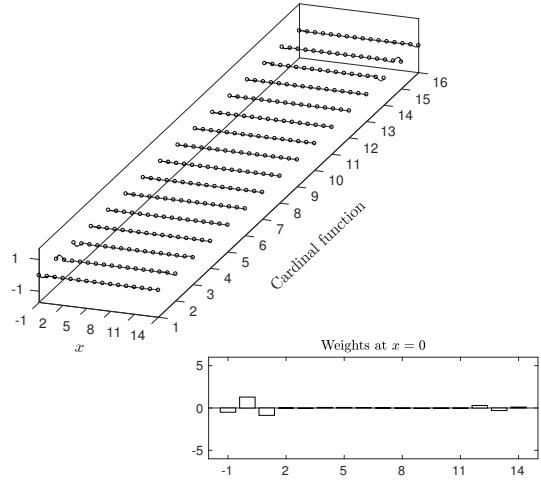
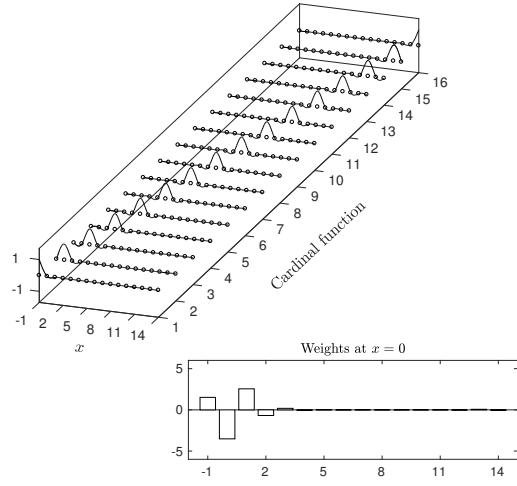


Figure 4: PHS+poly cardinal functions from Figure 1, split according to equation (9) into pure PHS cardinal functions $\hat{\psi}(x)$ (left column) and correction term $\underline{C}(x)$ (right column). The weights in the right column of the subplots, when added to the ones in the left column, leads to the RBF-FD weights displayed in Figure 1, ensuring the specified order of accuracy.

Lebesgue functions $\Lambda_n(x)$

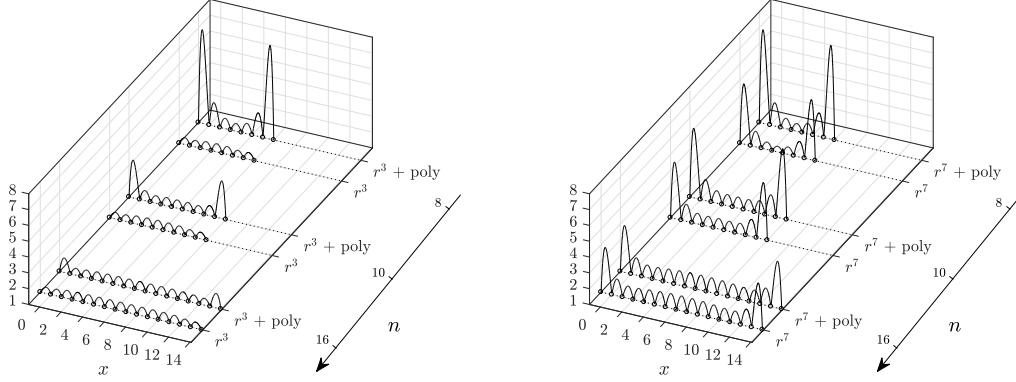


Figure 5: Lebesgue functions for PHS r^3 (left) and r^7 (right) combined with polynomials up to 7th degree using $n = 8, 10$ and 16 (y -axis). The Lesbesgue functions for approximations based only on PHS r^3 and r^7 are also displayed in the figure for comparison purposes. Dotted line represents optimal value equals one.

accuracy. By contrast, for highly resolved smooth functions this approach can significantly reduce boundary errors, as proven in 1-D and 2-D cases.

With that as motivation, in this section we have repeated a similar one dimensional numerical experiment. This is, the numerical approximation of $f(x) = -\arctan(5R(x + 1/2))$ over the interval $[-1, 1]$ using 16 equispaced nodes, and have varied the smoothness of the function through the parameter R . Figure 6 shows the approximation function (left column) as a function of R (rows) for $R = 1, 0.3, 0.1$ and $n = 16$. Note how the function becomes smoother as we zoom in. For each of the values considered, the middle column shows the approximation error when using PHS r^3 (solid),

$$\hat{s}(x) - f(x) = \sum_{k=1}^n f(x_k) \hat{\psi}_k(x) - f(x), \quad (13)$$

as well as the correction term (10) required to make PHS+poly exact up to 7th degree polynomials (dashed),

$$\sum_{k=1}^n f(x_k) C_k(x) = -f^T W \left[P^T \hat{\psi}(x) - \underline{p}(x) \right]. \quad (14)$$

The right column shows the approximation error when using cubic PHS combined with 7th degree polynomials, which is equivalent to the sum of the two terms above

$$s(x) - f(x) = \sum_{k=1}^n f(x_k) \hat{\psi}_k(x) - f(x) + \sum_{k=1}^n f(x_k) C_k(x). \quad (15)$$

Observe that for $R = 1$ the function is under-resolved. Cubic PHS produces an approximation error with oscillations not only at the edges, but also around $x = -1/2$ when the slope of the function is the largest. For this resolution, the correction term displays large edge oscillations that deteriorate the overall approximation. However, this trend reverts as the function becomes smoother. In this case, it is well-known that PHS becomes very accurate at the center of the stencil, with severe errors

near the edges of the domain [7]. The role of the correction term (14) is to account for this low accuracy near the edges. Observe it features similar edge oscillations ~~than-as~~ the PHS error, but with opposite sign. When added to (13), it restores the accuracy of the PHS+poly approximation (15). Figure 7 shows the convergence as a function of R for both, cubic PHS and cubic PHS+poly approximations. Remarkably, the action of the correction term in (15) transforms the convergence order from $O(R)$ to $O(R^8)$ for any n .

This behavior can be understood analytically using the approach derived in [2]. As R decreases, the function $f(x)$ can be approximated by a convergent Taylor series expansion centered at $x_0 \in [-1, 1]$,

$$f(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!} p_k(x - x_0).$$

Substitution into (13) leads to

$$\hat{s}(x) - f(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!} \left[\underline{p}_k^T \cdot \hat{\psi}(x) - p_k(x - x_0) \right], \quad (16)$$

where $\underline{p}_k^T = [p_k(x_1 - x_0), p_k(x_2 - x_0), \dots, p_k(x_n - x_0)]$. On the other hand, the term $\underline{f}^T W$ in the correction term (14) is equal to the coefficients of the Taylor series expansion up to the augmented degree l as proven in [2], i.e.

$$\underline{f}^T W = \left[\frac{f^{(0)}(x_0)}{0!}, \frac{f^{(1)}(x_0)}{1!}, \dots, \frac{f^{(l)}(x_0)}{l!} \right] + O(R^{l+1}).$$

Thereby, the addition of (14) and (16) in (15) cancels out all the terms of the expansion up to the augmented degree l , ensuring the convergence $O(R^{l+1})$. This numerical experiment illustrates why, by combining a PHS interpolant with polynomials, it is possible to significantly reduce boundary errors for highly resolved smooth functions (as found in different contexts in [7] and [8]).

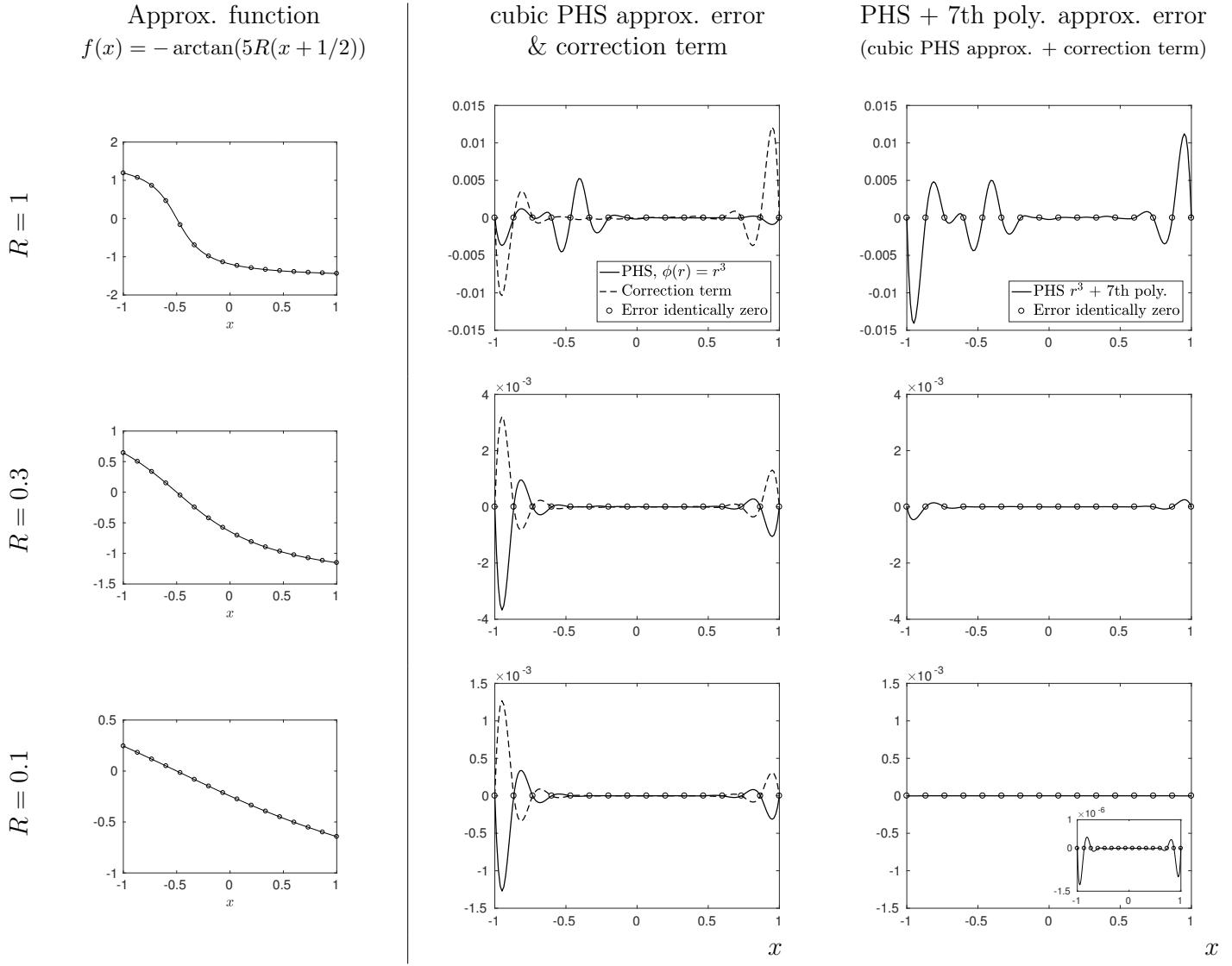


Figure 6: Interpolation error of the function $f(x) = -\arctan(5R(x+1/2))$ using cubic PHS (middle column) and cubic PHS combined with 7th degree polynomials (right column) with 16 equispaced nodes in the interval $[-1, 1]$ for $R = 1, 0.3$ and 0.1 (rows). The corresponding correction term (14) is plotted in dashed in the middle column. For each value of R , the interpolating function $f(x)$ is displayed in the left column of the figure.

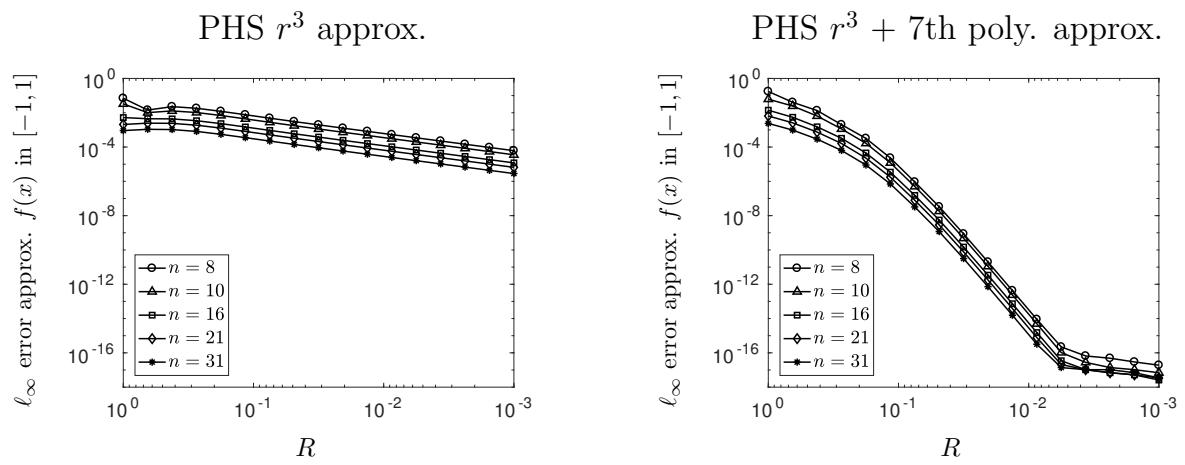


Figure 7: Convergence of the approximation errors displayed in Figure 6 as a function of R for different stencil sizes n .

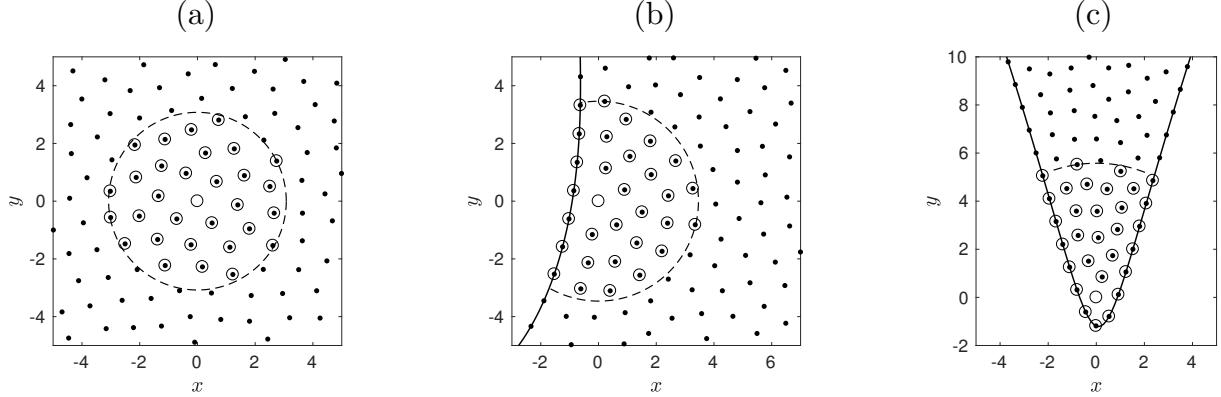


Figure 8: 2-D stencils (small circles) used to approximate the Laplacian at $(x, y) = (0, 0)$ (empty circle) for $n = 31$.

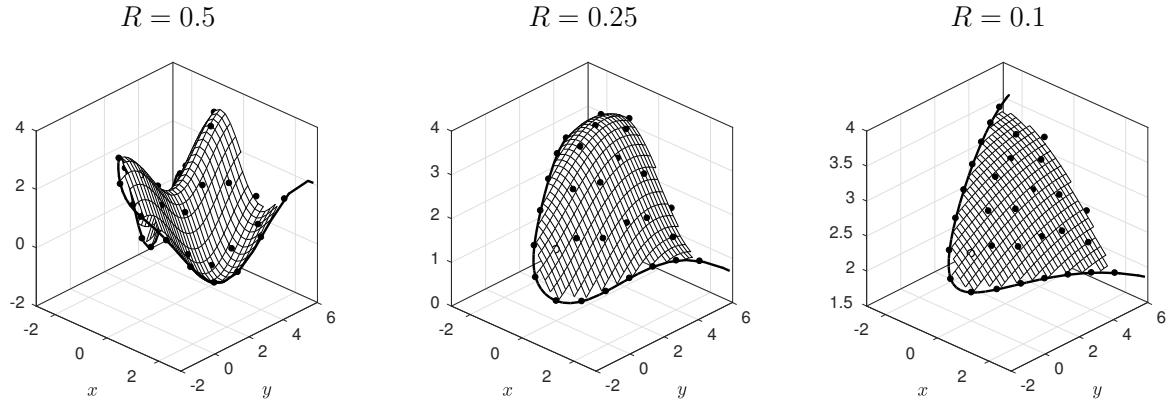


Figure 9: Test function (17), when displayed over the stencil (c) from Figure (8), with $R = 0.5$, $R = 0.25$ and $R = 0.1$, respectively. The nodes of the stencil are represented by black solid circles and the stencil center by an open circle.

4 Approximations near boundaries in 2-D: some illustrations of stencils

The aim of this section is to illustrate that the behavior observed for PHS+poly interpolants in 1-D holds as well in higher dimensions, regardless of the proximity of the boundary. To do so, we consider the 2-D stencils displayed in Figure 8. These correspond to different parts of the domain depicted in Figure 6 of [3] for an internodal distance $h \approx 0.025$.

4.1 Accuracy near edges

As a follow-up to Section 3.2.3, we want to show how the mechanism explained in 1-D applies as well to 2-D. In this way, we have approximated the test function

$$u(x, y) = 1 + \sin(4R x) + \cos(3R x) + \sin(2R y), \quad (17)$$

(also considered in [7]) over the stencil (c) in Figure 8 for $R = 0.5, 0.25$ and 0.1 . Figure 9 shows how the test function appears gradually more as a tilted plane over the stencil (c) from Figure (8).

The left column in Figure 10 shows the approximation error when using cubic PHS to approximate (17) for $n = 31$ (analogous to (13) but in 2-D). Observe the error decreases as a function of R (rows) at a very low rate. The corresponding correction term (defined by the 2-D version of (14)) is plotted in the middle column for polynomials up to 4th degree. As in the 1-D case, it gradually becomes similar to the cubic PHS error, but with opposite sign. When the test function becomes resolved over the 2-D stencil ($R = 0.1$), the addition of the two of them leads to an accurate PHS+poly approximation (right column).

The convergence of the approximation as a function of R is displayed in Figure 11 for both, cubic PHS approximations (left) and PHS+poly approximations (right). Observe that making the test function smoother has a very low impact on the accuracy for cubic PHS approximations. However, when combined with polynomials, they take over ensuring a converge rate $O(R^5)$ determined by the polynomial degree.

4.2 PHS+poly cardinal functions and Lebesgue functions

Two dimensional PHS+poly cardinal functions are displayed in Figure 7 of [2] for the stencil (a). A key observation in that study is the fact that the correction term required to ensure the polynomial reproduction decreases as a function of the stencil size n , leading to PHS+poly cardinal functions similar to PHS cardinal functions, but with an accuracy determined by the augmented polynomial degree.

The effects of this feature on the stability of the approach can also be analyzed through the Lebesgue function. As in the 1-D case, the Lebesgue function (12) is defined as the sum of the absolute values of the cardinal functions over the stencil. Figures 12 and 13 display the 2-D Lebesgue functions against n (rows) for each of the three geometries (columns) when using pure cubic PHS approximations and cubic PHS combined with 4th degree polynomials, respectively.

Observe in Figure 12 that the 2-D Lebesgue functions for cubic PHS approximations can deal very well with the domain edges (as already pointed out in Figure 5 in 1D). However, any approximation based on this approach would feature large error oscillations at the edges [7]. The key feature is that these can be overcome by simply augmenting the PHS interpolant with polynomials. It is well-known that polynomials can be very accurate at the edges, but it may also cause negative effects on the stability of the interpolant (Runge's phenomenon).

This is indeed the case observed for $n = 16$ in Figure 13. The approximation is dominated by the augmented polynomials as PHS requires a large correction term to ensure the polynomial reproduction. Specially bad is the stencil (b). For (a) and (c), the minimum value is obtained at the center of the stencil, increasing towards the edges. Nonetheless, the fact that the required correction term decreases as the stencil size increases (columns) makes the Lebesgue function to quickly revert to the ideal case of pure cubic PHS depicted in Figure 12. This illustrates heuristically the trade-off between stability and accuracy of PHS+poly approximations, which is found in [3] to be controlled by the stencil size n .

4.3 RBF-FD stencils

As established by equation (11), the RBF-FD weights approximating a linear differential operator over a stencil can be computed as derivatives of the cardinal functions. In the case of RBF augmented with polynomials, this is equivalent to the RBF-FD weights obtained from pure RBF ap-

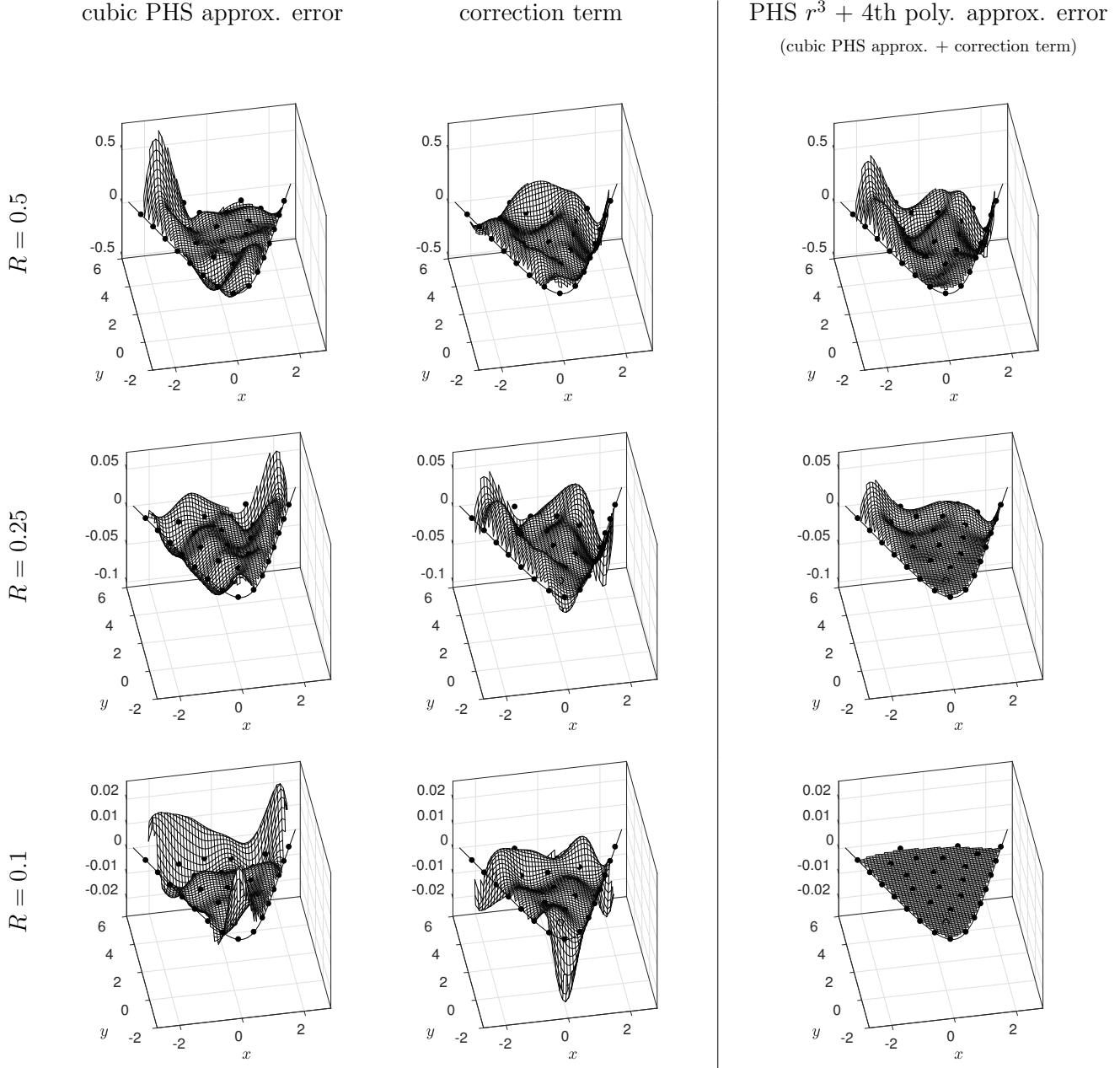


Figure 10: Interpolation error of the test function (17) when using cubic PHS (left column) and cubic PHS combined with 7th degree polynomials (right column) over the stencil (c) displayed in Figure 8 with $n = 31$ scattered nodes. The corresponding correction term is plotted in the middle column. Note the vertical scale for PHS+poly approx. error with $R = 0.1$.

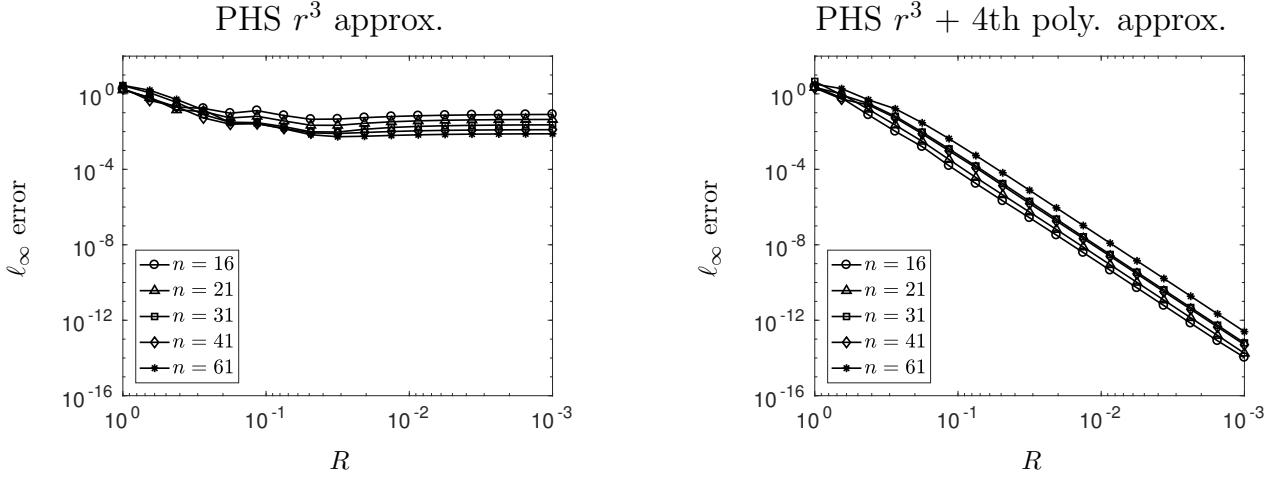


Figure 11: Convergence of the approximation errors displayed in Figure 10 as a function of R for different stencil sizes n .

proximations plus a correction term ensuring the exact reproduction of the augmented polynomial terms.

Following this result, Figures 14-16 show the RBF-FD weights approximating the Laplacian over the stencils from Figure (8), using cubic PHS augmented with 4th degree polynomials for stencil sizes $n = 16, 21$ and 31 . These have been split according to equation (11) into the RBF contribution and the correction term. The weights in the middle column of the subplots, when added to the ones in the right column, leads to the RBF-FD weights in the left column, ensuring the specified order of accuracy.

Figure 14 shows a symmetric stencil, where the evaluation point at $(0, 0)$ is surrounded by nodes in any direction. In this case, as the stencil size increases, the PHS quickly take over from the polynomials. The correction term becomes small and the RBF-FD weights display a pattern very similar to PHS.

Figure 15 shows an example of a near-one-sided stencil. This situation is similar to the one encountered when approximating differential operators near boundaries. For $n = 16$ the RBF-FD weights are dominated by the correction term, with large weights spread over the whole stencil. However, as the stencil size increases, it quickly reverts to the case where the pattern of weights is dominated by PHS, and the correction term has a negligible effect.

Figure 16 shows a more extreme case, where the stencil center is surrounded by nodes mostly in a single direction. Even in this case, as the stencil size increases, the RBF-FD weights recover the pattern determined by the PHS term. However, the effect of the correction term is slightly stronger in this case.

To test the accuracy of these stencils, we have computed the convergence rate that PHS+poly approximations provide when approximating the Laplacian of function (17). This is plotted in Figure 17. Observe the polynomial nature of the correction term ensures the converge rate $O(R^{l-1})$ found in [7] for any stencil. **The effect of the stencil size on the accuracy is almost negligible.**

As stated at the beginning of Section 4, these stencils corresponds to different parts of the domain

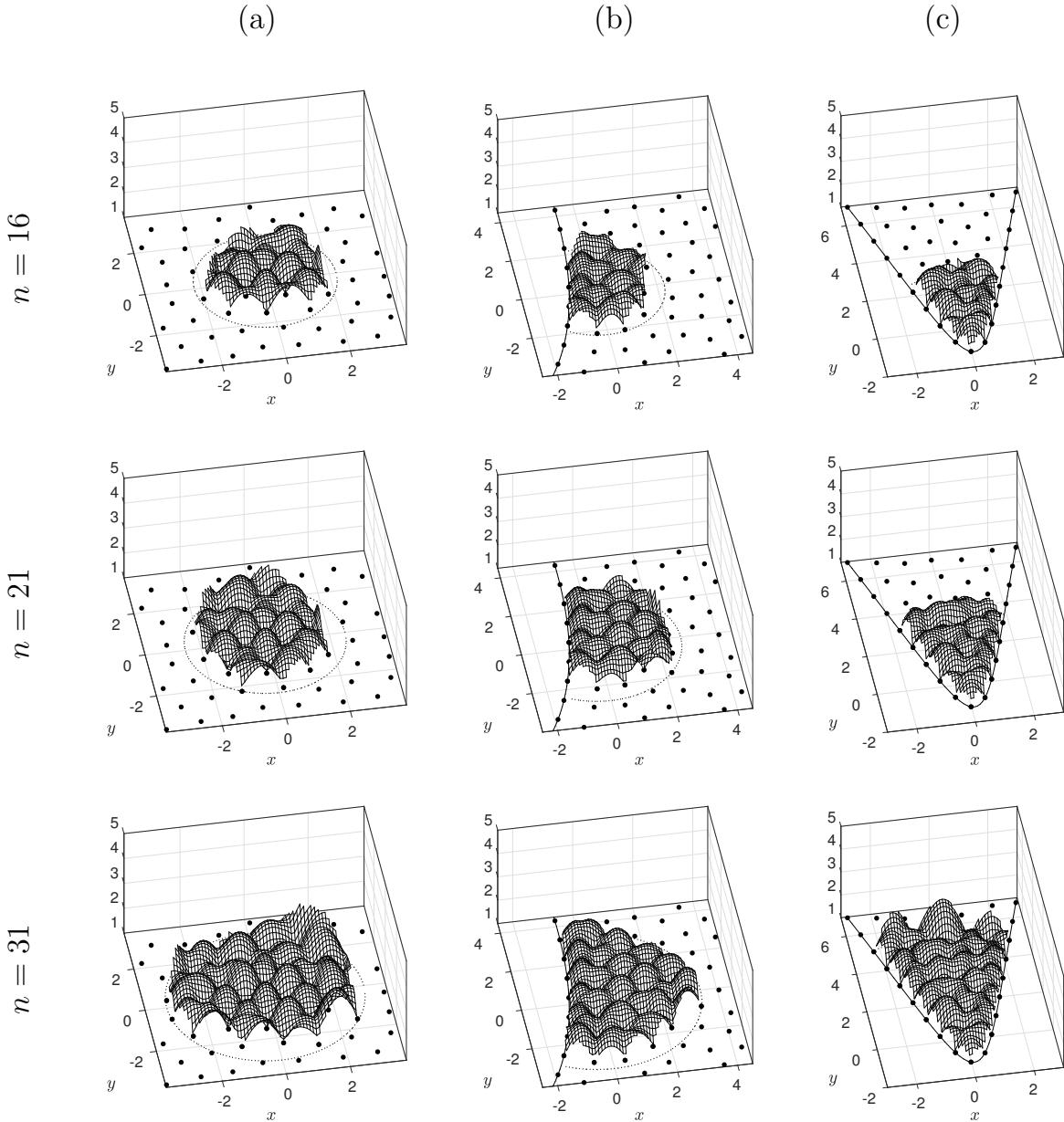


Figure 12: Lebesgue functions for the stencils shown in Figures 9 (columns) when using cubic PHS (no polynomials) with $n = 16, 21$ and 31 (rows). Note the vertical scale is the same in all subplots.

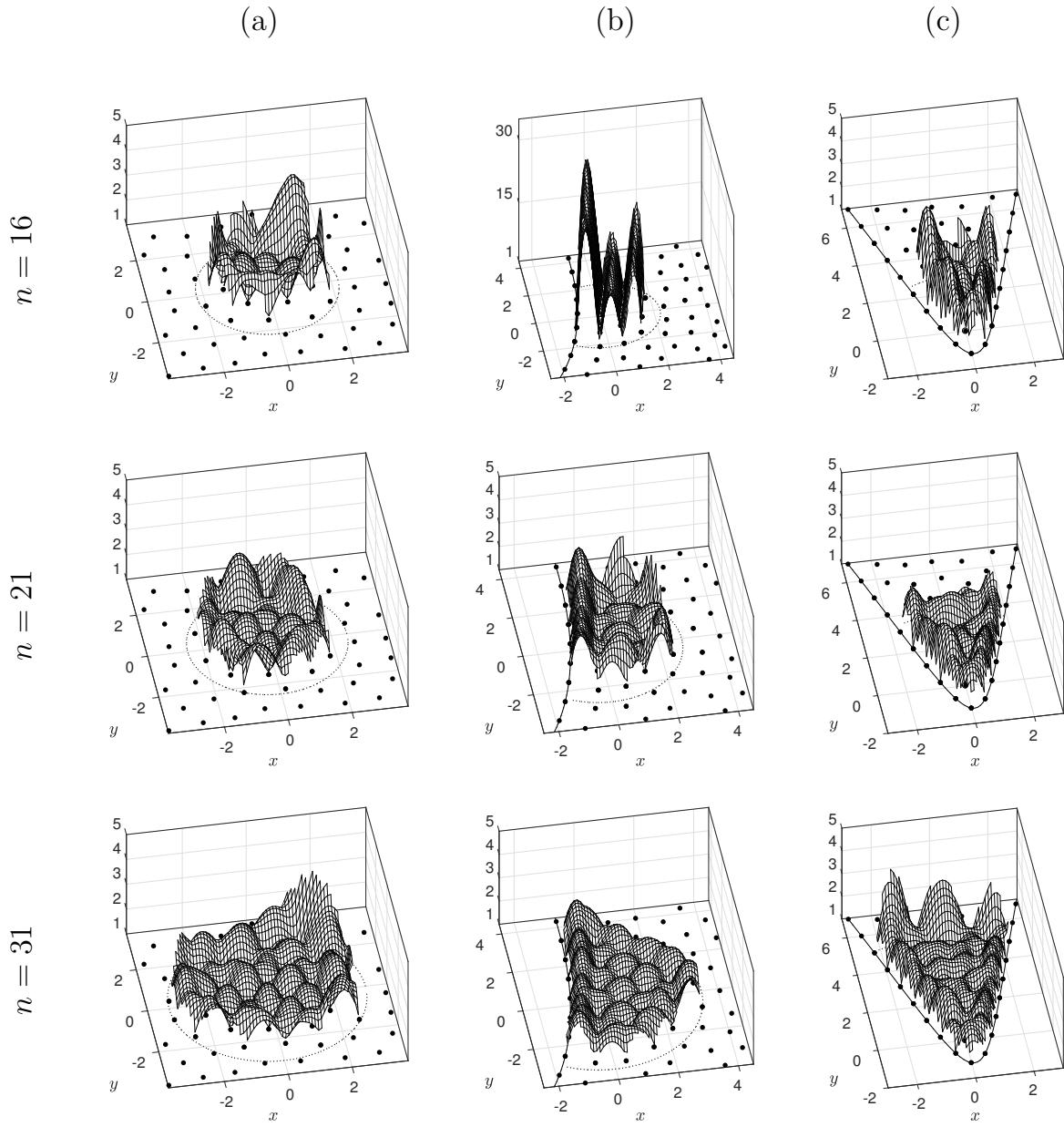
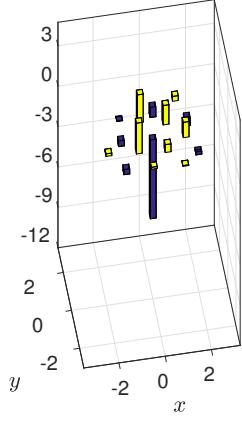


Figure 13: Lebesgue functions for the stencils shown in Figures 9 (columns) when using cubic PHS combined with 4th degree polynomials for $n = 16, 21$ and 31 (rows). Note the vertical scale is the same in all subplots apart from a factor 6 difference in (b), top case ($n = 16$).

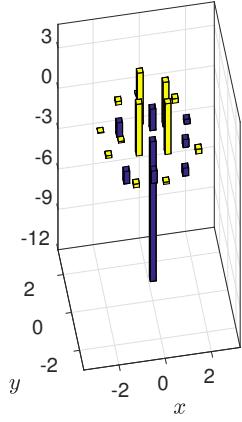
considered in Figure 6 of [3] for $h \approx 0.025$. Discretizing the Laplacian over the interior of this domain, leads to a differentiation matrix with an eigenvalue pattern as represented in Figure 18. Critically important for when solving elliptic PDEs is the fact that the positiveness of the matrix is restored by simply increasing the stencil size [without affecting accuracy](#) (in agreement with the behavior graphically represented by the pattern of weights from Figures 14 to 16, [and the convergence rates from Figure 17](#)). This illustrates the mechanism observed in [3] able to ensure both stability and accuracy when numerically solving elliptic PDEs over highly irregular domains.

PHS r^3 + 4th poly.
based RBF-FD weights
 $\Delta\hat{\psi}(0, 0)$

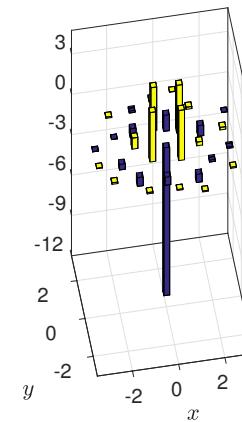
$n = 16$



$n = 21$



$n = 31$



PHS r^3 based
RBF-FD weights
 $\Delta\hat{\psi}(0, 0)$

Correction term
 $\Delta C(0, 0)$

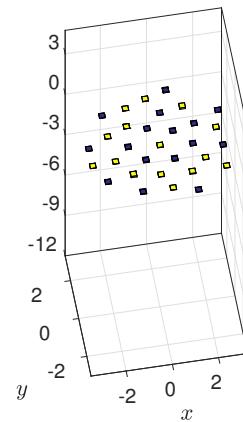
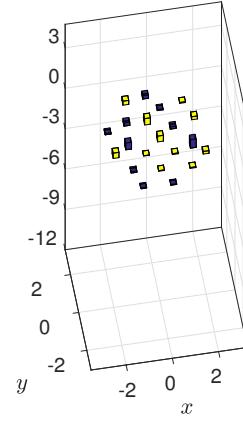
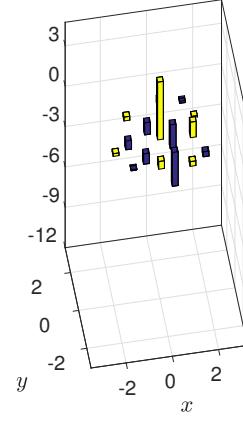
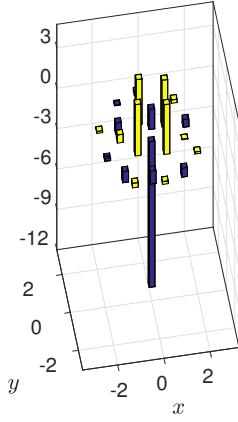
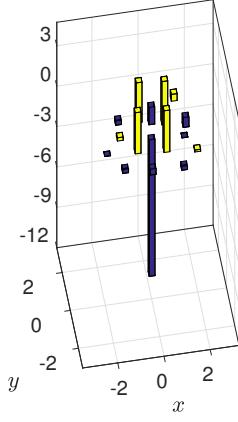


Figure 14: RBF-FD weights (left column) approximating the Laplacian over the 2-D stencil (a) displayed in Figure 8 using cubic PHS augmented with 4th degree polynomials for stencil sizes $n = 16, 21, 31$ (rows). According to equation (9), they are split into the contribution of pure PHS weights $\Delta\hat{\psi}(0, 0)$ (middle column) and the correction term $\Delta C(0, 0)$ (right column), as in the 1-D case displayed in Figure (4). Note that the weights in the right column ensure the specified order of accuracy when added to the ones in the middle column.

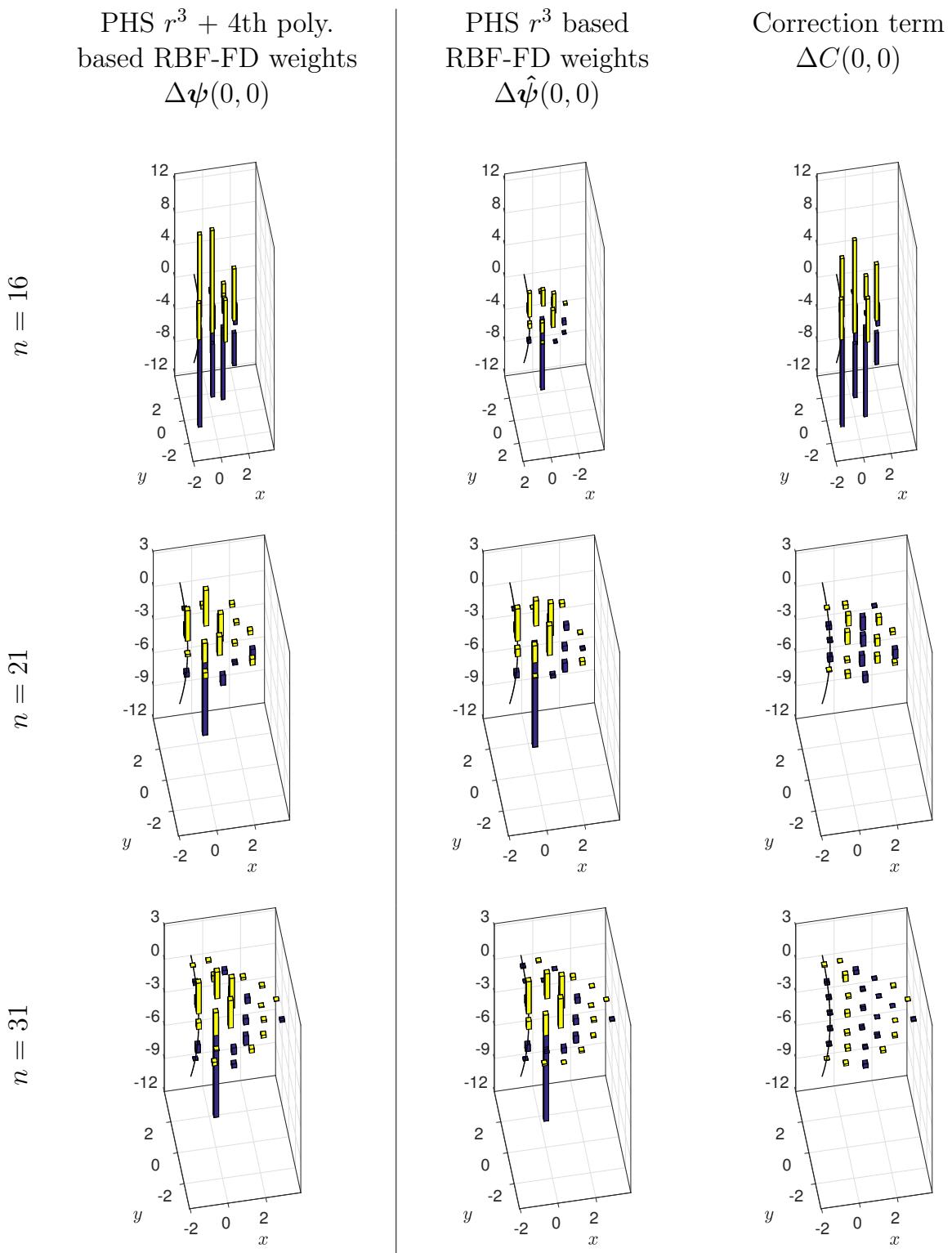


Figure 15: Same as Figure 14 but for the 2-D stencil (b) displayed in Figure 8.

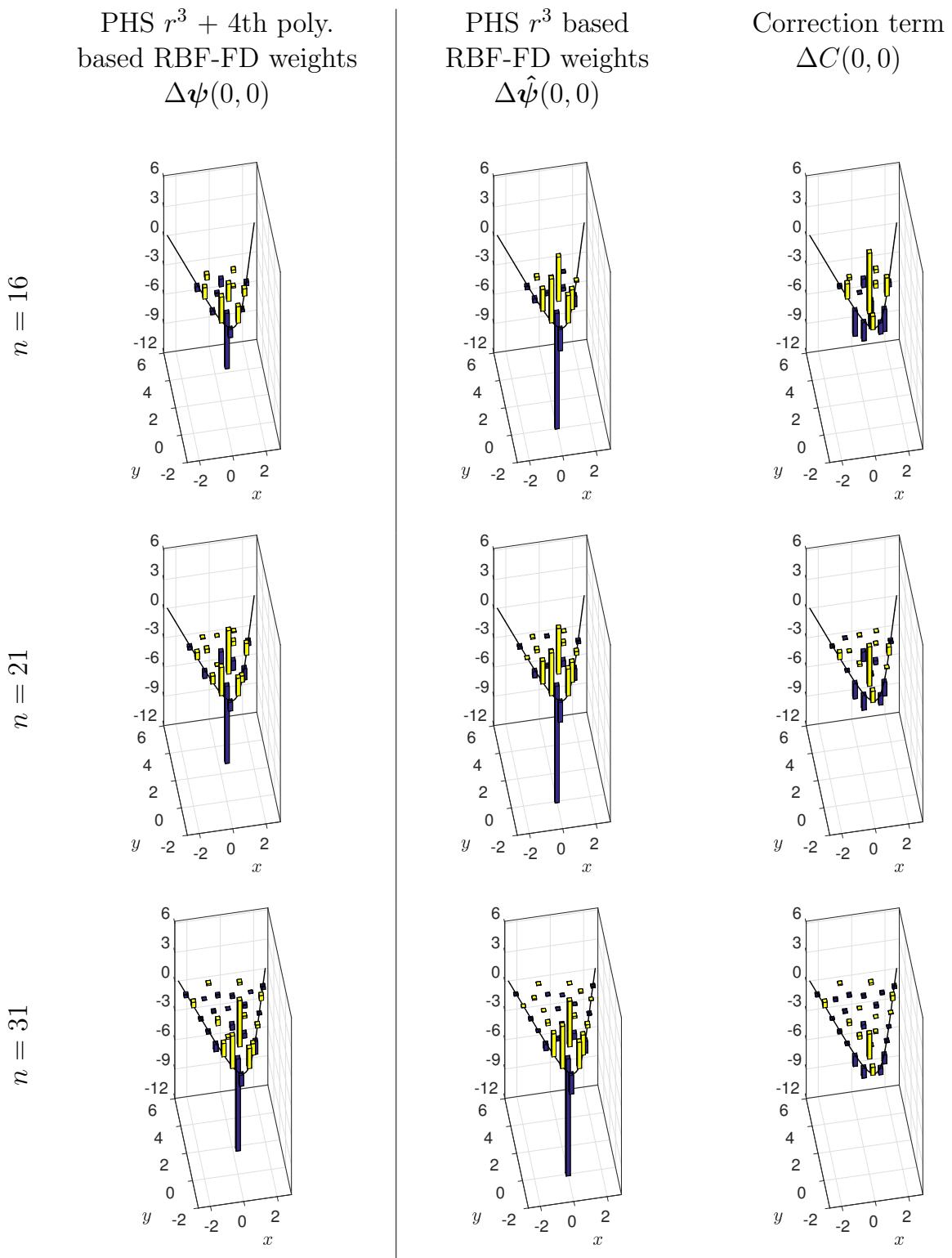


Figure 16: Same as Figure 14 but for the 2-D stencil (c) displayed in Figure 8.

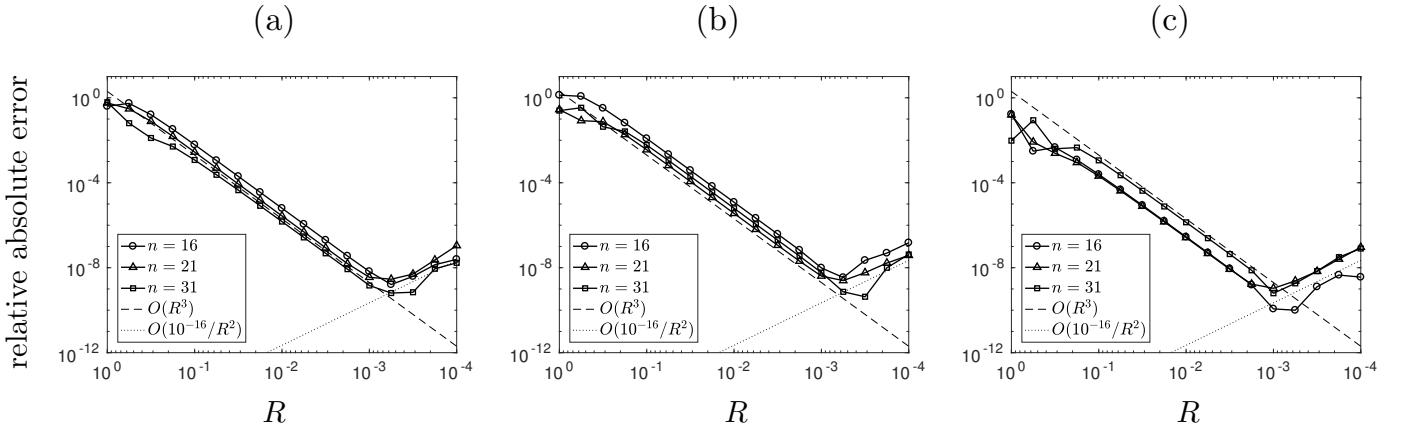


Figure 17: Relative absolute error approximating the Laplacian of function (17) on the previous stencils centered at $(0, 0)$ as a function of R . The thin dashed lines represent the convergence order $O(R^3)$. The dotted line marks the round-off limit $10^{-16}/R^2$.

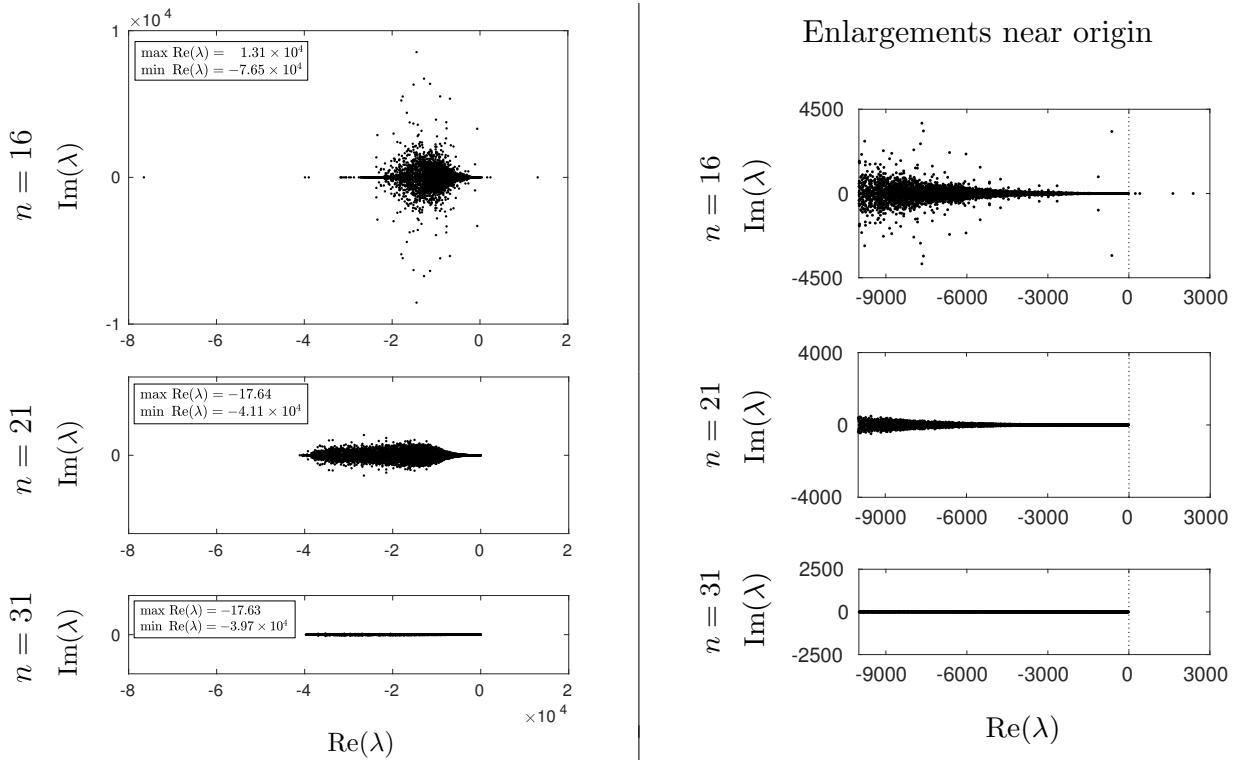


Figure 18: **Eigenvalue**—Left column: eigenvalue distribution of the differentiation matrix approximating the Laplacian over the irregular domain depicted in Figure 6 of [3] for $h \approx 0.025$, using cubic PHS combined with 4th degree polynomial terms and $n = 16, 21, 31$. The scales on the axis are the same in the three subplots. The $\min \operatorname{Re}(\lambda)$ and $\max \operatorname{Re}(\lambda)$ values are displayed in each case. Right column: same eigenvalue distributions enlarged near the origin.

5 Some illustrations of 3-D RBF-FD stencils

The meshless nature of RBF approximations makes this approach specially suitable for PDE problems in high dimensions. Since the approximations only depend on the distance between nodes, the procedure to discretize a differential operator is basically the same regardless of the number of space dimensions. The aim of this section is to illustrate through a numerical test that the features found for RBF+poly approximations near boundaries in 1-D and 2-D hold as well in 3-D (and presumably in any space dimension), as established by the results in [2].

We have therefore considered a three dimensional node distribution, and selected three different stencils that generalize the 2-D geometries considered in the previous Section. The domain chosen is the Stanford bunny, discretized in this case using JIGSAW¹ [4]. The results are shown in Figures 19-21. For each stencil, we have computed the RBF-FD weights approximating the 3-D Laplacian at the stencil center \underline{x}_0 using cubic PHS combined with 4th degree polynomials (left column), split according to equation (11) into the contribution of cubic PHS weights (middle column) and the correction term (right column). This leads to 3rd order accurate approximations.

Figure 19 shows the pattern of weights over a symmetric stencil, similar to stencil (a) in the previous section. Blue corresponds to negative differentiation weights and yellow to positive. The radius of the spheres reflects the magnitude of the weights, and the grey line connects each node with the stencil center. Observe in this case that the correction term has an impact on the structure of the RBF-FD weights only for $n = 39$. As the stencil size increases, the correction required to enforce the polynomial reproduction of the approximation becomes smaller and the pattern of RBF+poly weights recovers the cubic PHS structure.

For near one-sided stencils, the role of the correction term is stronger, but the same behavior holds. For instance, Figure 20 (which displays a stencil inside the nose of the bunny), generalizes to 3-D the stencil (b) from the previous section. In this type of geometry, the stencil center is close to the boundary and the information is picked nearly from one side. A more extreme case is displayed in Figure 21, corresponding with a stencil inside the left ear of the bunny. In this geometry, the correction term required to make the approximation exact for polynomials is the largest. Observe that in both cases, as the stencil size increases, the magnitude of the correction term decreases as a function of the stencil size, leading to a recovery of the cubic PHS structure.

Figure 22 shows the convergence of approximating the Laplacian of the test function

$$u(x, y, z) = 1 + \sin(4R x) + \cos(3R x) + \sin(2R y) + e^{-(x^2+y^2+z^2)R^2} \quad (18)$$

over the previous stencils as a function of R . Observe that in all of them, the convergence rate is $O(R^{l-1})$, being determined by the augmented polynomial degree $l = 4$. The stencil size has a negligible effect on the accuracy of the approximation. Therefore, as pointed out previously in the 2-D case when solving elliptic PDEs, increasing the stencil size can restore the positiveness of the differentiation matrix without having negative effects on the accuracy of the approximation. This certainly represents an advantageous feature when handling irregular domains in 3-D as found in [3].

¹JIGSAW is an unstructured mesh generator freely available at <https://sites.google.com/site/dengwirda/jigsaw>.

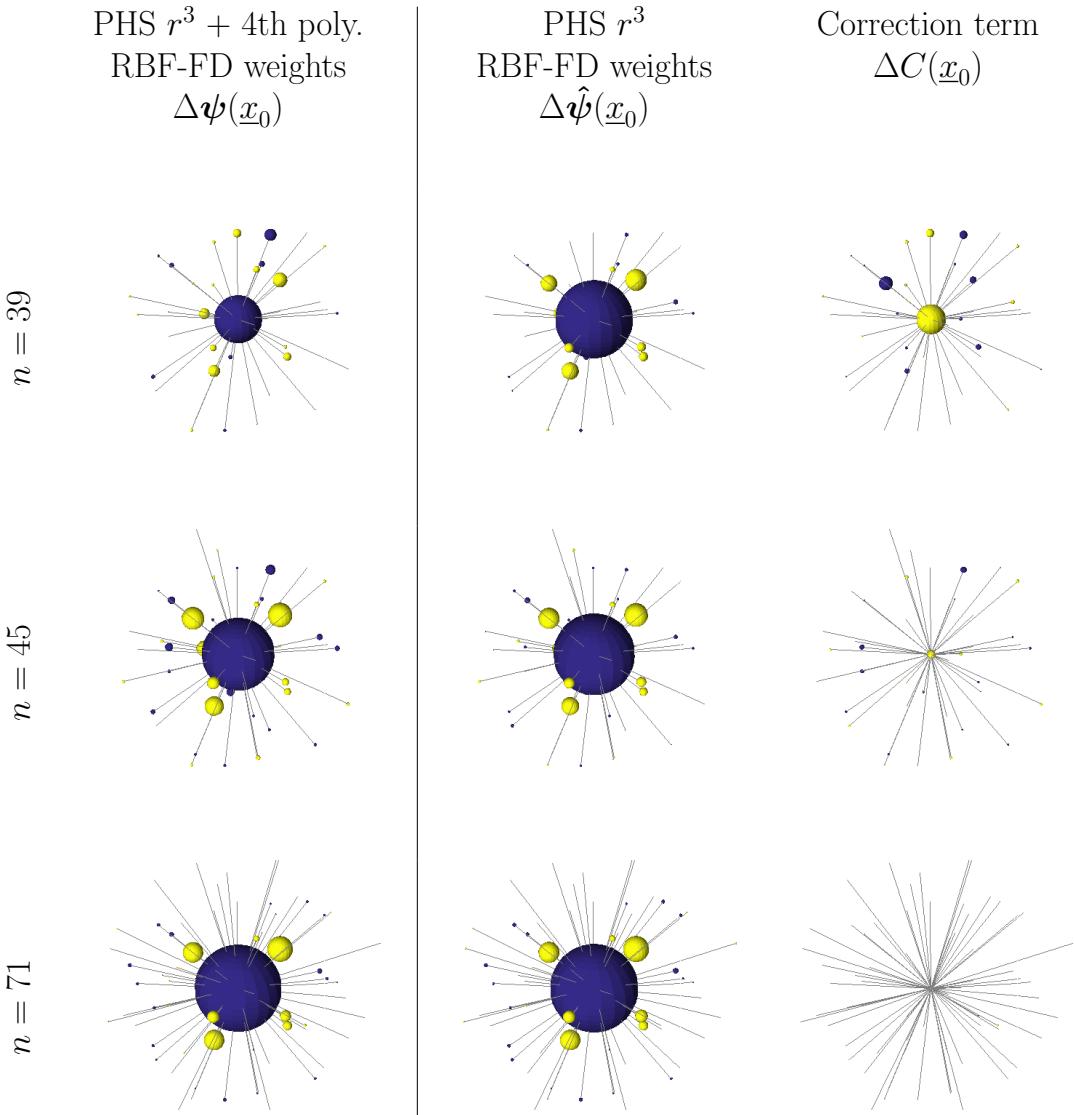


Figure 19: RBF-FD weights (left column) approximating the Laplacian over a 3-D stencil using cubic PHS augmented with 4th degree polynomials for $n = 39, 45, 71$ (rows). They have been split according to equation (11) into the contribution of pure PHS weights $\Delta\hat{\psi}$ (middle column) and the correction term ΔC (right column). Blue corresponds to negative differentiation weights and yellow to positive. The radius of the spheres reflects the magnitude of the weights. The grey line connects each node with the stencil center.

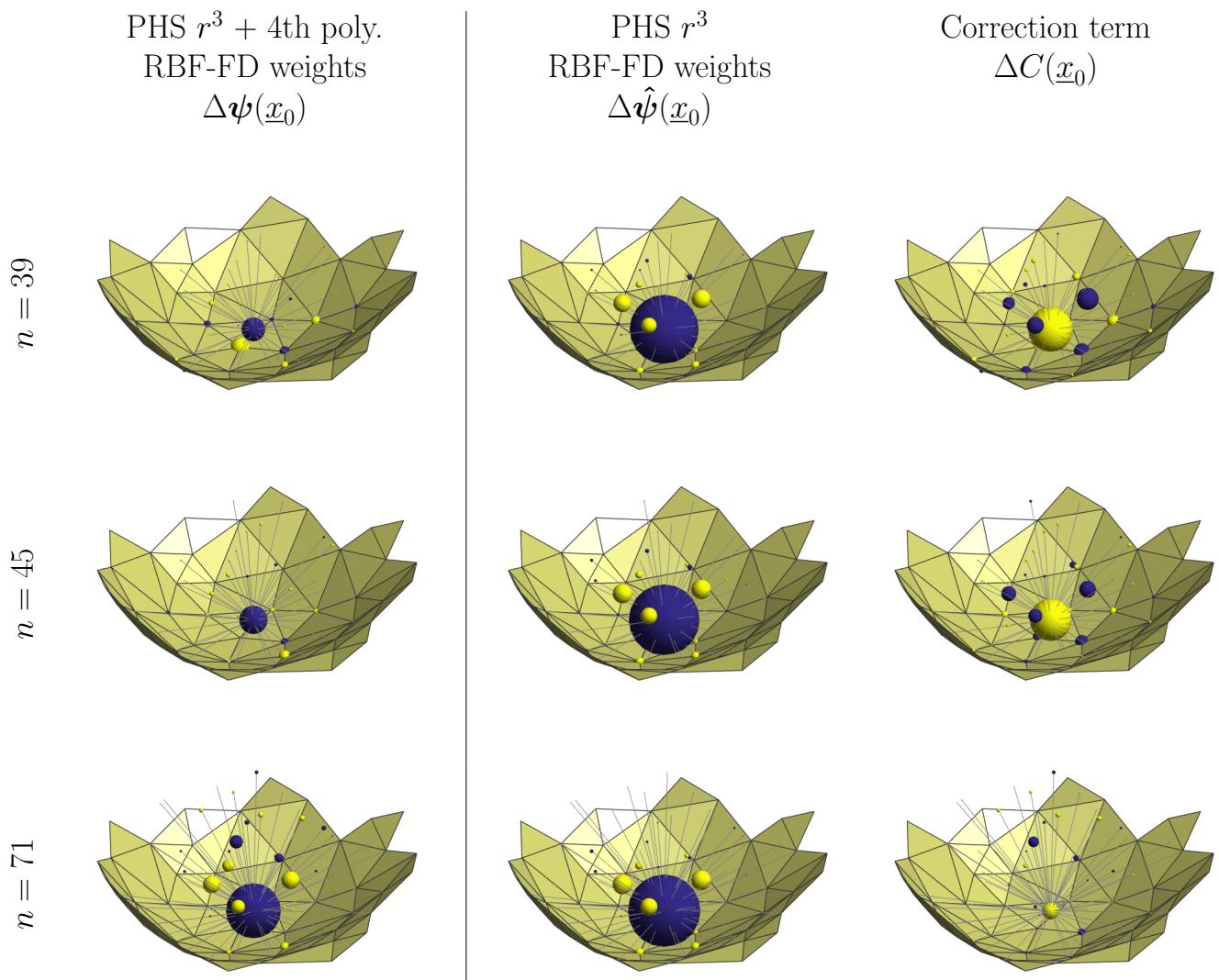


Figure 20: Same as Figure 19 but for a near one-sided 3-D stencil.

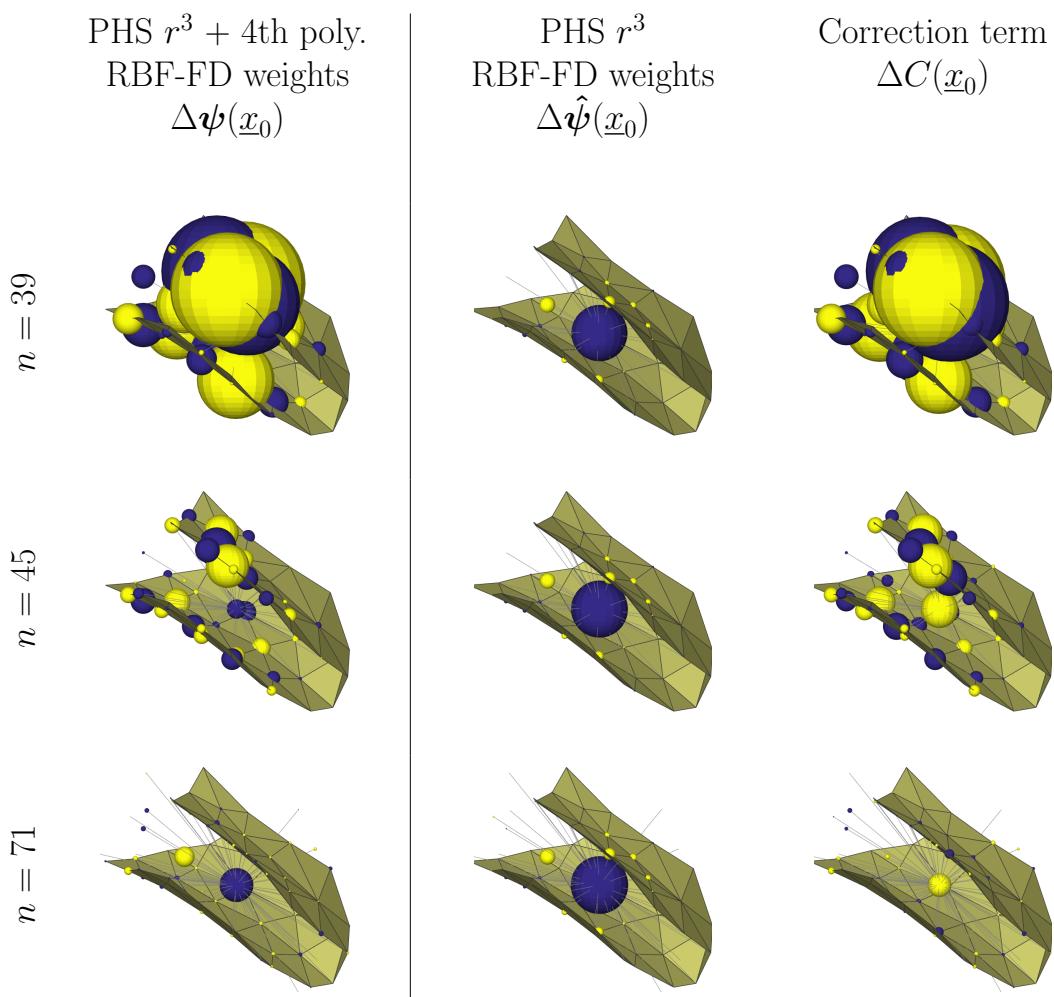


Figure 21: Same as Figure 19 but for a more extreme one-sided 3-D stencil.

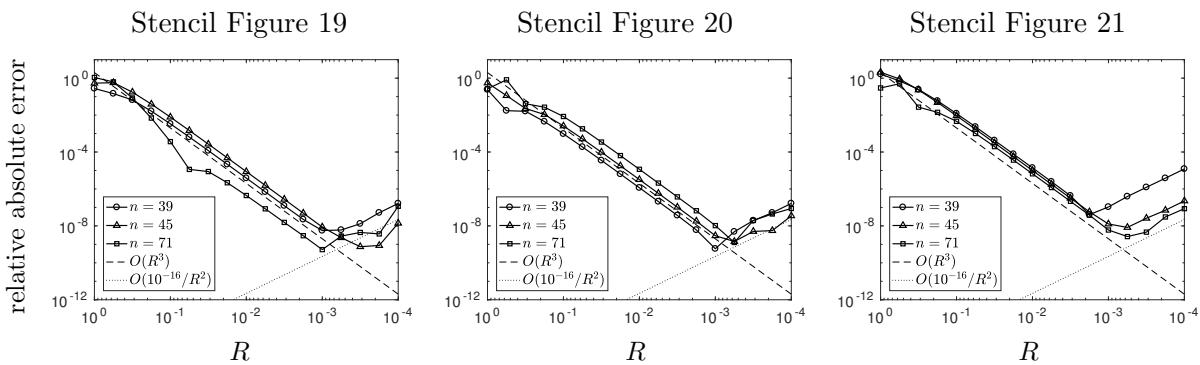


Figure 22: Relative absolute error approximating the Laplacian of function (18) on the previous stencils as a function of R . As in Figure 17, the thin dashed lines represent the convergence order $O(R^3)$ and the dotted line marks the round-off limit $10^{-16}/R^2$.

6 Conclusions

Some previous studies [2, 3, 7] have described how combining PHS-type RBFs with high degree polynomials becomes an especially attractive approach for creating RBF-FD formulas. One key feature is that high orders of accuracy then can be achieved without having to choose an optimal shape parameter and without having to deal with issues related to numerical ill-conditioning. This present study focuses on another both surprising and very useful feature of using PHS+poly to calculating RBF-FD weights, namely that relatively large stencils will then combine high orders of accuracy with an absence of Runge-phenomenon-type boundary errors. We have here provided several perspectives on this property. Following a mostly heuristic discussion which suggests this result, we decompose PHS+poly interpolants into pure PHS interpolants plus a correction term (ensuring the order of accuracy). As the stencil size increases, these correction terms are found to decrease in size. The RBF+poly approximations ~~become then then become~~ dominated by the RBF part (which is well behaved also at boundaries), and has thus removed the Runge phenomenon (while having maintained high order of accuracy, in the sense of being exact for polynomials up to specified degree). This feature of PHS+poly generated RBF-FD approximations has here been numerically shown to hold also in two and three space dimensions.

Acknowledgements

The work of Víctor Bayona was supported by Spanish MECD Grant FIS2016-77892-R.

References

- [1] G. A. Barnett, *A robust RBF-FD formulation based on polyharmonic splines and polynomials*, Ph.D. thesis, University of Colorado, Boulder, CO, 2015.
- [2] V. Bayona, *An insight into RBF-FD approximations augmented with polynomials*, (submitted to CAMWA).
- [3] V. Bayona, N. Flyer, B. Fornberg, and G. A. Barnett, *On the role of polynomials in RBF-FD approximations: II. Numerical solution of elliptic PDEs*, J. Comput. Phys. **332** (2017), 257–273.
- [4] D. Engwirda, *Voronoi-based point-placement for three-dimensional Delaunay-refinement*, Procedia Engineering **124** (2015), 330–342.
- [5] G. E. Fasshauer, *Meshfree Approximation Methods with MATLAB*, Interdisciplinary Mathematical Sciences - Vol. 6, World Scientific Publishers, Singapore, 2007.
- [6] N. Flyer, G.A. Barnett, and L.J. Wicker, *Enhancing finite differences with radial basis functions: experiments on the Navier-Stokes equations*, Journal of Computational Physics **316** (2016), 39–62.
- [7] N. Flyer, B. Fornberg, V. Bayona, and G. A. Barnett, *On the role of polynomials in RBF-FD approximations: I. Interpolation and accuracy*, J. Comput. Phys. **321** (2016), 21–38.
- [8] B. Fornberg, T.A. Driscoll, G. Wright, and R. Charles, *Observations on the behavior of radial basis function approximations near boundaries*, Computers & Mathematics with Applications **43** (2002), no. 3-5, 473–490.

- [9] B. Fornberg and N. Flyer, *A Primer on Radial Basis Functions with Applications to the Geosciences*, SIAM, Philadelphia, 2015.
- [10] ———, *Solving PDEs with radial basis functions*, Acta Numerica **24** (2015), 215–258.
- [11] B. Fornberg, N. Flyer, S. Hovde, and C. Piret, *Locality properties of radial basis function expansion coefficients for equispaced interpolation*, IMA J. Num. Anal. **28** (2007), no. 1, 121–142.
- [12] B. Fornberg and J. Zuev, *The Runge phenomenon and spatially variable shape parameters in RBF interpolation*, Comput. Math. Appl. **54** (2007), 379–398.
- [13] J. A. Reeger and B. Fornberg, *Numerical quadrature over smooth surfaces with boundaries*, J. Comput. Phys. **355** (2018), 176–190.