

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/335088467>

# Comparison of Moving Least Squares and RBF+poly for Interpolation and Derivative Approximation

Article in *Journal of Scientific Computing* · August 2019

DOI: 10.1007/s10915-019-01028-8

---

CITATION

1

READS

225

1 author:



Víctor Bayona  
University Carlos III de Madrid

19 PUBLICATIONS 481 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



mass fixer [View project](#)

# Comparison of Moving Least Squares and RBF+poly for interpolation and derivative approximation

Víctor Bayona \*

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

August 1, 2019

## Abstract

The combination of polyharmonic splines (PHS) with high degree polynomials (PHS+poly) has recently opened new opportunities for radial basis function generated finite difference (RBF-FD) approximations. The PHS+poly formulation, which relies on a polynomial least squares fitting to enforce the local polynomial reproduction property, resembles somehow the so-called moving least squares (MLS) method. Although these two meshfree approaches are increasingly used nowadays, no direct comparison has been done yet. The present study aims to fill this gap, focusing on scattered data interpolation and derivative approximation. We first review the MLS approach and show that under some mild assumptions PHS+poly can be formulated analogously. Based on heuristic perspectives and numerical demonstrations, we then compare their performances in 1-D and 2-D. One key result is that, as previously found for PHS+poly, MLS can also overcome the edge oscillations (Runge's phenomenon) by simply increasing the stencil size for a fixed polynomial degree. This is, however, controlled by a weighted least squares fitting which fails for high polynomial degrees. Overall, PHS+poly is found to perform superior in terms of accuracy and robustness.

## 1 Introduction

The problem of finding a continuous function  $u(\mathbf{x})$  that approximates a discrete set of data  $(\mathbf{x}_i, f_i)$ ,  $i = 1, \dots, N$ , with  $\mathbf{x}_i \in \mathbb{R}^d$  and  $f_i \in \mathbb{R}$ , is a basic and common issue in many scientific disciplines, where knowing information about the processes also at locations different from the measurements is needed. This *approximation problem* is usually solved by representing the *approximant*  $u(\mathbf{x})$  as a linear combination of certain basis functions  $\phi_k$ ,

$$u(\mathbf{x}) = \sum_{j=1}^N c_k \phi_k(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^d, \tag{1}$$

where  $\phi_k(\mathbf{x})$  form a linearly independent set and the coefficients  $c_k$  are unknown constants, which are to be determined in such a way that  $u(\mathbf{x})$  represent a “good” fit to the given data. There are

---

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

of course many ways to decide what “good” means, and the criterion is usually related with the minimization of an error function.

One common approach is to enforce (1) to be exact on the data values, i.e.  $u(\mathbf{x}_i) = f_i$ ,  $i = 1, \dots, N$ . This is known as *interpolation*, and leads to the system of linear equations

$$A\mathbf{c} = \mathbf{f},$$

where the entries of the interpolation matrix  $A$  are given by  $A_{ij} = \phi_j(\mathbf{x}_i)$ ,  $i, j = 1, \dots, N$ ,  $\mathbf{c} = [c_1, \dots, c_N]^T$  and  $\mathbf{f} = [f_1, \dots, f_N]^T$ . This is well-posed (a solution exists and is unique) if and only if  $A$  is non-singular.

The interpolation by univariate polynomials is a widespread approach. One can unique interpolate arbitrary data at  $N$  distinct data sites using a polynomial of degree  $N - 1$ . However, interpolation by polynomials of several variables is much more intricate. In the 1950s, the Mairhuber-Curtis theorem [10, 31] stated that it is not possible to perform unique interpolation with multivariate polynomials of degree  $N$  to data given at arbitrary locations for  $d \geq 2$ . This essentially rules out the use of multivariate polynomial basis for interpolation over scattered node sets (known as *scattered data interpolation*), which is required in most of the applications. Tensor product interpolation extends the univariate theory in this case, where the nodes are placed on structured grids. Such approach restricts the geometric flexibility severely.

Multivariate meshfree approximation methods, considered nowadays as the early precursors of the *moving least squares* (MLS) and the *radial basis function* (RBF) methods, appeared in the late 60s - early 70s from mapping problems in geodesy and geophysics. The key feature of these approaches is that they are guaranteed to be non-singular, no matter how the nodes (assume distinct) are scattered in any number of dimensions [16].

The general moving least squares method, as it is formulated nowadays, first appeared in 1981 by Lancaster and Salkauskas [27], who pointed out the connection with the weighted Shepard method [38] and the works of McLain [32] and Lancaster [26] on surface approximation. Bos and Salkauskas [8] noticed also the similarities with the Backus-Gilbert theory [2], and proved that MLS approximations can be interpreted as a minimization problem. The approximation power of MLS for functional approximations was investigated further by Levin in [28]. Since it first appeared, the MLS method has been widely used in different application areas. Some of the most relevant contributions have been in computer graphics, for surface representation [17, 22, 37]; in particle methods, to enforce boundary conditions [7, 11, 12, 40]; and in engineering, for derivative approximation [1, 29, 30].

The RBF method is a generalization of the multiquadric method (MQ), first introduced by Hardy around 1970 [23] to solve a problem in cartography. Several milestones in the history of RBF methods preceded this work, such as the pioneering works of Duchon [13, 14, 15] and Meinguet [33, 34] on thin plate splines (TPS) and surface splines, which fall under what we refer nowadays to as *polyharmonic splines*; the 1982 survey by Franke [21], who compared various scattered data interpolation methods (including the weighted Shepard method) and concluded that MQs and TPSs were the best (conjecturing also that the collocation matrix for MQs is invertible); and the proof by Micchelli of this conjecture [35] in 1986. These all accelerated the early development of RBF methods, which have been increasingly used in different areas of application since then. See the monographs [16] and [20] for references.

The use of RBF methods for the numerical solution of partial differential equations (PDEs) was

first proposed by Kansa [24, 25] in 1990. Since then, it has been successfully applied to a wide range of problems (the monograph [20] includes a full list of references). Several issues (such as a high computational cost or ill-conditioning) led a local formulation, the so-called RBF generated finite differences (RBF-FD), to appear in the early 2000's [39, 41, 42]. Essentially, it is a generalization of the classical finite difference (FD) method to scattered node layouts, yielding highly sparse differentiation matrices (better-conditioned than in the global approach) and high-order algebraic convergence. More recently, it has been found the advantages of PHS+poly generated RBF-FD weights over the standard approach based on infinitely smooth RBFs [3, 4, 5, 6, 18, 19]. Among the key features, (i) high orders of accuracy can be achieved without the need of selecting a shape parameter or the issues related to numerical ill-conditioning, and (ii) the harmful edge effects associated to the use of high order polynomials (better known as Runge's phenomenon) can be overcome by simply increasing the stencil size for a fixed polynomial degree.

PHS+poly approximations, which relies on an underlying polynomial least-square fitting, resembles the MLS method. This connection has not been investigated enough yet. In [19], where the use of PHS+poly for the generation of RBF-FD approximations was first proposed, the performance of PHS+poly approximations was compared against polynomial least squares. The main outcome of this analysis is that PHS+poly outperforms polynomial least-squares approximations in terms of accuracy. However, polynomial least squares is just the simplest choice within the moving least squares approach and it can be highly improved through the inclusion of a weight function. With that as motivation, the present work focuses on understanding their similarities and differences for scattered data interpolation and derivative approximation.

We first review the MLS approach in Section 2. Then, RBF approximations augmented with polynomials (RBF+poly) are revisited in Section 3, showing that under some mild assumptions they can also be formulated as either a minimization problem or as a weighted least-squares fitting. Following the numerical tests from [4, 5, 6, 19], we then compare their performances for scattered data interpolation and derivative approximation in Section 4. Heuristic perspectives and numerical demonstrations are provided in 1-D and 2-D. The polynomial reproduction property ensures an algebraic convergence order determined by the degree of polynomial terms in the two cases. The key result is finding that MLS approximations can also overcome the harmful edge oscillations due to the use of high order polynomials (Runge's phenomenon) by simply increasing the stencil size for a fixed polynomial degree. The underlying mechanism is then applied to the numerical solution of the Poisson equation over the unit disk (as in [6]). Unlike PHS+poly approximations, which become dominated by the spline nature of the interpolant, this mechanism is found to hold only up to moderately high polynomial degrees. The main conclusions are summarized in Section 5.

## 2 Moving Least Squares approximation

In this section we review the moving least squares method, focusing on the formulations as best least squares approximation and quadratic minimization problem (for a detailed description see [16]). The role of the weight function in the approximation is then discussed, as well as the convergence order. These results will be used later to compare MLS against RBF+poly approximations.

### 2.1 Weighted Least Squares approximation

Consider the following approximation problem. Assume we are given data values  $f(\mathbf{x}_i)$ ,  $i = 1, \dots, N$ , on some distinct data sites  $\mathbf{x}_1, \dots, \mathbf{x}_N$ , where  $\mathbf{x}_i \in \mathbb{R}^d$ ,  $d \geq 1$ , and  $f$  is some smooth

function. We wish to find the best approximation  $u(\mathbf{x})$  to  $f$  from the space of multivariate polynomials  $\Pi_l^d$  of total degree  $l$  in  $d$  dimensions, which is formed by  $s = \binom{l+d}{l} < N$  elements.

In standard weighted least squares approximations, this is obtained by finding the coefficients  $c_j$  in

$$u(\mathbf{x}) = \sum_{j=1}^s c_j p_j(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^d,$$

that minimizes the functional  $\|f - u\|_2$ . In this case, the inner product is defined as

$$\langle f, g \rangle_w = \sum_{i=1}^N f(\mathbf{x}_i)g(\mathbf{x}_i)w_i, \quad (2)$$

where the weights  $w_i = w(\mathbf{x}_i)$ ,  $i = 1, \dots, N$ , are positive discrete coefficients. The induced norm leads to the functional

$$\sum_{i=1}^N [f(\mathbf{x}_i) - s(\mathbf{x}_i)]^2 w_i.$$

Minimizing it with respect to the coefficients  $c_j$  yields the normal equations

$$\sum_{j=1}^s c_j \langle p_j, p_k \rangle_w = \langle f, p_k \rangle_w, \quad k = 1, \dots, s, \quad (3)$$

where  $\langle p_j, p_k \rangle_w$  is the Gram matrix and  $\langle f, p_k \rangle_w$  is the projection of the function  $f$  onto the space of multivariate polynomials  $\Pi_l^d$ . Alternatively, this system of equations can be obtained by making the residual of the approximation  $f - u$  orthogonal to the approximation space  $\Pi_l^d$  with respect to the inner product (2).

In matrix notation, equation (3) becomes

$$P^T W P \mathbf{c} = P^T W \mathbf{f},$$

where the  $N \times s$  matrix  $P$  has entries  $P_{ij} = p_j(\mathbf{x}_i)$ ,  $W$  is a diagonal matrix of weights  $W = \text{diag}(w_1, \dots, w_N)$ , and the vectors  $\mathbf{c}$  and  $\mathbf{f}$  are  $\mathbf{c} = [c_1, \dots, c_s]^T$  and  $\mathbf{f} = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_N)]^T$ , respectively. Since the weights are positive and the polynomial basis is unisolvant on the data sites, the Gram matrix is positive definite. Therefore, the coefficients  $c_j$  are given by

$$\mathbf{c} = (P^T W P)^{-1} P^T W \mathbf{f},$$

and the weighted least squares approximation at  $\mathbf{x}$  is equal to

$$u(\mathbf{x}) = \mathbf{f}^T W P (P^T W P)^{-1} \mathbf{p}(\mathbf{x}), \quad (4)$$

where  $\mathbf{p} = [p_1(\mathbf{x}), \dots, p_s(\mathbf{x})]^T$  contains a basis for the multivariate polynomial space  $\Pi_l^d$ . In general, (4) does not interpolate the data but represent some sort of approximation. Note that polynomial least squares is recovered if  $W$  is equal to the identity matrix.

## 2.2 Standard Moving Least Squares approximation

Moving Least Squares can be interpreted as a local alternative to the global approach described above. Given data  $(\mathbf{x}_i, f(\mathbf{x}_i))$ ,  $i = 1, \dots, N$ , where  $\mathbf{x}_i \in \mathbb{R}^d$ ,  $d \geq 1$ , and  $f$  is some smooth function, we wish to find the best approximation  $u(\mathbf{x})$  to  $f$  at a point  $\mathbf{x}_c$  from the space of multivariate polynomials  $\Pi_l^d$ ,

$$u(\mathbf{x}_c) = \sum_{j=1}^s c_j p_j(\mathbf{x})|_{\mathbf{x}_c}, \quad (5)$$

with respect to the norm induced by the inner product

$$\langle f, g \rangle_{w(\mathbf{x}_c)} = \sum_{i=1}^n f(\mathbf{x}_i)g(\mathbf{x}_i)w_i(\mathbf{x}_c). \quad (6)$$

The key difference is that (6) sums over the  $n$ -th closest data sites  $\mathbf{x}_i$  to the stencil center  $\mathbf{x}_c$ , with  $s = \binom{l+d}{l} < n \ll N$ . The weights  $w_i(\mathbf{x}_c) = w(\mathbf{x}_i, \mathbf{x}_c)$ ,  $i = 1, \dots, n$ , are also positive continuous functions centered at the stencil nodes  $\mathbf{x}_i$  and evaluated at  $\mathbf{x}_c$ . As we will show in Section 2.4, the approximant will interpolate the data when the weight function is appropriately selected, having an important role in the interpolatory properties of MLS.

This approach leads to the functional

$$\sum_{i=1}^n [f(\mathbf{x}_i) - s(\mathbf{x}_i)]^2 w_i(\mathbf{x}_c),$$

which is minimum for the coefficients determined by the normal equations

$$(P^T W(\mathbf{x}_c) P) \mathbf{c} = P^T W(\mathbf{x}_c) \mathbf{f}, \quad (7)$$

where the same matrix notation as in Section 2.1 is used here, with  $W(\mathbf{x}_c) = \text{diag}(w_1(\mathbf{x}_c), \dots, w_n(\mathbf{x}_c))$  and  $\mathbf{f} = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_n)]^T$ . The positiveness of the weight function and the unisolvency of the polynomials guarantee that  $(P^T W(\mathbf{x}_c) P)$  is positive-definite. Thus, the coefficients  $c_j$  are uniquely determined and the MLS approximation (5) is equal to

$$u(\mathbf{x}_c) = \mathbf{f}^T W(\mathbf{x}_c) P (P^T W(\mathbf{x}_c) P)^{-1} \mathbf{p}(\mathbf{x})|_{\mathbf{x}_c}. \quad (8)$$

The main difference with respect to the global approach described in Section 2.1 is that one needs to solve the small linear system of equations (7) for each evaluation point  $\mathbf{x}_c$ .

## 2.3 Moving Least Squares formulation as a quadratic minimization problem

Moving least squares can be formulated as a quadratic minimization problem. The connection between this formulation (also known as Backus-Gilbert approach [2]) and moving least squares was pointed out by Bos and Salkauskas [9]. Consider the same approximation problem as before. In the Backus-Gilbert approach, one does attempt to find an approximant of the form

$$u(\mathbf{x}) = \sum_{i=1}^n f(\mathbf{x}_i) \psi_i(\mathbf{x}), \quad (9)$$

by determining the generating functions  $\psi_i(\mathbf{x})$  that minimize the quadratic functional

$$\frac{1}{2} \sum_{i=1}^n \frac{1}{w_i(\mathbf{x}_c)} \psi_i^2(\mathbf{x}) \quad (10)$$

subject to the polynomial reproduction constraints

$$\sum_{i=1}^n p_k(\mathbf{x}_i) \psi_i(\mathbf{x}) = p_k(\mathbf{x}), \quad k = 1, \dots, s, \quad (11)$$

where  $p_k(\mathbf{x})$  are elements of the space of  $d$ -variate polynomials  $\Pi_l^d$  of total degree at most  $l$ .

To determine the generating functions  $\psi(\mathbf{x})$ , we use the Lagrange multipliers and construct the Lagrangian

$$L(\psi, \lambda) = \frac{1}{2} \psi^T W^{-1}(\mathbf{x}_c) \psi + \lambda^T [P^T \psi - \mathbf{p}],$$

where  $\psi = [\psi_1(\mathbf{x}), \dots, \psi_n(\mathbf{x})]^T$ ,  $W^{-1}(\mathbf{x}_c) = \text{diag}\left(\frac{1}{w_1(\mathbf{x}_c)}, \dots, \frac{1}{w_n(\mathbf{x}_c)}\right)$ ,  $\lambda = [\lambda_1(\mathbf{x}), \dots, \lambda_s(\mathbf{x})]^T$  are the Lagrange multipliers,  $P$  has entries  $P_{ij} = p_j(\mathbf{x}_i - \mathbf{x}_c)$ , and  $\mathbf{p} = [p_1(\mathbf{x} - \mathbf{x}_c), \dots, p_s(\mathbf{x} - \mathbf{x}_c)]^T$  contains a basis for the multivariate polynomial space  $\Pi_l^d$ .

Since  $W^{-1}(\mathbf{x}_c)$  is positive definite by definition, the Lagrangian is convex and has a unique minimum. Enforcing  $\nabla_{\psi, \lambda} L = \mathbf{0}$  yields the system of equations

$$\begin{bmatrix} W^{-1}(\mathbf{x}_c) & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} \psi \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{p} \end{bmatrix}, \quad (12)$$

whose solution is

$$\begin{aligned} \lambda(\mathbf{x}) &= (P^T W(\mathbf{x}_c) P)^{-1} \mathbf{p}(\mathbf{x}), \\ \psi(\mathbf{x}) &= W(\mathbf{x}_c) P \lambda(\mathbf{x}). \end{aligned}$$

Therefore, the MLS approximation (9) at  $\mathbf{x}_c$  is given by

$$u(\mathbf{x}_c) = \sum_{i=1}^n f(\mathbf{x}_i) \psi_i(\mathbf{x})|_{\mathbf{x}_c} = \mathbf{f}^T W(\mathbf{x}_c) P (P^T W(\mathbf{x}_c) P)^{-1} \mathbf{p}(\mathbf{x})|_{\mathbf{x}_c},$$

which is equivalent to (8). Moreover, the generating functions are given by

$$\psi(\mathbf{x}) = W(\mathbf{x}) P (P^T W(\mathbf{x}) P)^{-1} \mathbf{p}(\mathbf{x}). \quad (13)$$

This result will be used in the following section to analyze the role of the weight function.

## 2.4 Interpolatory moving least squares: the role of the weight function

The generating functions  $\psi(\mathbf{x})$  determine the behavior of the MLS approximants. It is well-known that the point-wise error is minimized if the generating functions satisfy the cardinal conditions  $\psi_i(\mathbf{x}_j) = \delta_{ij}$ ,  $i, j = 1, \dots, n$  (interpolation conditions). The closeness of  $\psi(\mathbf{x})$  to the point-wise delta functions is controlled by the weight functions. As long as they are positive definite (such as radial

basis functions), the MLS approach is well-posed and will represent some sort of approximation (not necessarily interpolating the data) enforced by the polynomial reproduction constraint (11).

The work by Lancaster and Salkauskas [27], inspired by previous ideas from Shepard [38], MacLain [32] and Lancaster [26], was actually the first to formally introduce the interpolating moving least squares method based on the notion of inner-product spaces with appropriate singularities in the weights. As they showed, the weight functions must be singular at the data locations for the MLS approximant to interpolate the data.

To proof this, consider for simplicity the moving least squares method in the case  $s = 1$  with  $l = 0$ , which is a moving average originally attributed to Shepard [38] (known as Shepard's interpolant). In this case, the polynomial space is formed by the element  $p_1(\mathbf{x}) = 1$  and the generating functions (13) are

$$\psi_i(\mathbf{x}) = \frac{w_i(\mathbf{x})}{\sum_{j=1}^n w_j(\mathbf{x})}, \quad i = 1, \dots, n. \quad (14)$$

The idea is to consider as weights the functions

$$w_i(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_i\|^{-\alpha}, \quad \alpha \in \mathbb{N}, \quad (15)$$

which are infinite at the data locations  $\mathbf{x}_i$ , and decrease towards zero with distance from the origin. In this case, it is easy to proof analytically that (14) satisfies the following properties:

- (i)  $\psi_i(\mathbf{x}_j) = \delta_{ij}$ ,  $i, j = 1, \dots, n$ ,
- (ii)  $0 \leq \psi_i(\mathbf{x}) \leq 1$ ,  $\forall \mathbf{x}$ ,
- (iii)  $\psi_i(\mathbf{x}) = 0$  if and only if  $\mathbf{x} = \mathbf{x}_j$ ,  $j \neq i$ ,
- (iv)  $\sum_{i=1}^n \psi_i(\mathbf{x}) = 1$ ,  $\forall \mathbf{x}$ ,
- (v)  $\psi_i(\mathbf{x}) \rightarrow 1/n$  as  $\|\mathbf{x}\| \rightarrow \infty$ .

In particular, (i) ensures the interpolation of the data by the MLS approximant.

For the general case  $s > 1$ , it is not possible to find a simple closed-form expression as (14). Instead, Levin [28] realized that for singular weight functions as (15), the only solution of the quadratic minimization problem at the data locations  $\mathbf{x} = \mathbf{x}_j$ ,  $j = 1, \dots, n$ , is

$$\psi_i(\mathbf{x}_j) = \delta_{ij}, \quad i, j = 1, \dots, n, \quad (16)$$

since it solves the polynomial reproduction constraint

$$\sum_{i=1}^n p_k(\mathbf{x}_i) \psi_i(\mathbf{x}_j) = p_k(\mathbf{x}_j), \quad k = 1, \dots, s,$$

minimizing the functional (10), which is actually

$$\frac{1}{2} \sum_{i=1}^n \frac{1}{w_i(\mathbf{x}_j)} \psi_i^2(\mathbf{x}_j) = 0.$$

Therefore, as long as the weight functions are singular at the data locations, MLS approximations will interpolate the data. Moreover, the resulting MLS approximation (and so, the generating

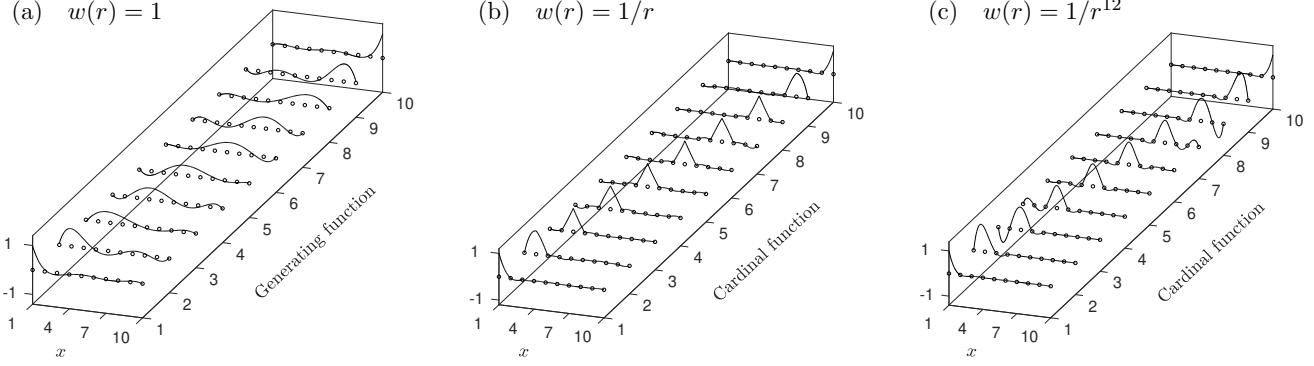


Figure 1: Generating functions over the equispaced stencil  $x = 1, \dots, 10$  using polynomials of degree  $l = 5$ , for the weight function (15) with  $\alpha = 0, 1$  and  $12$ . Observe that for  $\alpha = 0$ , MLS is equivalent to the polynomial least squares case and the generating functions do not interpolate the data. For  $w(r) = 1/r$  and  $w(r) = 1/r^{12}$ , the generating functions are actually interpolatory, increasing the smoothness with the degree  $\alpha$ .

functions) will be infinitely smooth as proven by Levin in [28]. In the following we use (15), but other choices are possible. For instance, he suggested the use of  $w_i(\mathbf{x}) = 1 / (\exp(||\mathbf{x} - \mathbf{x}_i||^2) - 1)$ , which is also infinitely smooth with a singularity at the origin. In this case, he also proof the non-singularity of the collocation system (12) if  $P$  is unisolvant and  $1 \in \Pi_l^d$ .

To illustrate the role of the weight function, we have displayed in Figure 1 the generating functions computed over the equispaced stencil  $x = 1, \dots, 10$  using polynomials of degree  $l = 5$  and the weight function (15) with  $\alpha = 0, 1$  and  $12$ . Observe that for (a)  $w(r) = 1$ , the generating functions do not satisfy the cardinal conditions. This case corresponds with pure polynomial least squares. For singular weight functions, such as (b)  $w(r) = 1/r$  and (c)  $w(r) = 1/r^{12}$ , the generating functions fulfill this property and the smoothness of the cardinal functions increases with the degree  $\alpha$ .

## 2.5 Approximation error

MLS reproduces exactly any element from the polynomial space  $\Pi_l^d$  of degree at most  $l$  in  $d$  variables. This is clear from the polynomial reproduction constraint (11). Moreover, it determines the convergence order of the approximation under refinement. To show this, consider the numerical approximation of a smooth function  $f$  at a point  $\mathbf{x}_0$  using MLS. It can be expanded in a Taylor series over the stencil as

$$\mathbf{f} = \sum_{k=1}^{\infty} L_k[f(\mathbf{x}_0)] \mathbf{p}_k, \quad (17)$$

where the operator  $L_k$  is equal to

$$L^{(\beta)} = \frac{1}{\beta_1! \cdots \beta_d!} \frac{\partial^\beta}{\partial x_1^{\beta_1} \cdots \partial x_d^{\beta_d}}, \quad (18)$$

and the vector

$$\mathbf{p}_k = [p_k(\mathbf{x}_1 - \mathbf{x}_0), p_k(\mathbf{x}_2 - \mathbf{x}_0), \dots, p_k(\mathbf{x}_n - \mathbf{x}_0)]^T$$

contains the element  $p^{(\beta)}(\mathbf{x}) = x_1^{\beta_1} \cdots x_d^{\beta_d}$  of total degree  $\beta = \beta_1 + \cdots + \beta_d$  evaluated on the stencil nodes. In our notation the subscript  $k$  just numbers the elements. When (17) is substituted in (9),

the polynomial reproduction constraint (11) leads to

$$u(\mathbf{x}) = \sum_{k=1}^s L_k[f(\mathbf{x}_0)] p_k(\mathbf{x} - \mathbf{x}_0) + \sum_{k=s+1}^{\infty} L_k[f(\mathbf{x}_0)] \mathbf{p}_k^T \cdot \boldsymbol{\psi}(\mathbf{x}),$$

where all the Taylor terms up through degree  $l$  are treated exactly by the generating functions (13). As a result, the approximation error is given by

$$f(\mathbf{x}) - u(\mathbf{x}) = \sum_{k=s+1}^{\infty} L_k[f(\mathbf{x}_0)] [p_k(\mathbf{x} - \mathbf{x}_0) - \mathbf{p}_k^T \cdot \boldsymbol{\psi}(\mathbf{x})],$$

which can be bounded as

$$\|f(\mathbf{x}) - u(\mathbf{x})\|_{\infty} \leq Ch^{l+1} \max_{\mathbf{x} \in \Omega} |L^{(l+1)}[f(\mathbf{x})]|.$$

Hence, the convergence order of the approximation is determined by the polynomial degree. The role of the weight function is to make the approach interpolatory, affecting only the residual as a multiplying constant  $C$ .

### 3 Local RBF approximations augmented with polynomials

In this section, we briefly revisit RBF approximations augmented with polynomials (RBF+poly). Under the assumption that the collocation matrix  $A$  is positive definite, we show that RBF+poly can also be formulated as either a best least squares fitting (after defining an appropriate norm), or as a quadratic minimization problem. Finally, the cardinal functions and approximation error are considered.

#### 3.1 Standard formulation

Consider the following interpolation problem. Given the data  $f_i = f(\mathbf{x}_i)$  at location  $\mathbf{x}_i \in \mathbb{R}^d$ ,  $i = 1, 2, \dots, N$  and  $d \geq 1$ , we wish to find the local multivariate interpolant at a point  $\mathbf{x}$  of the form

$$s(\mathbf{x}) = \sum_{j=1}^n \lambda_j \phi(\|\mathbf{x} - \mathbf{x}_j\|) + \sum_{k=1}^s \beta_k p_k(\mathbf{x}), \quad (19)$$

such that  $s(\mathbf{x}_i) = f_i$ ,  $i = 1, \dots, n$  is verified. Here, the sum is over the  $n$ -th closest data sites  $\mathbf{x}_i$  to the stencil center  $\mathbf{x}$  ( $n \ll N$ ),  $\phi(\|\mathbf{x} - \mathbf{x}_j\|)$  are radial basis functions (RBF) centered at the data locations  $\mathbf{x}_j$  and  $p_k(\mathbf{x})$  is an element of the multivariate polynomial space  $\Pi_l^d$  of total degree  $l$  in  $d$  dimensions formed by  $s = \binom{l+d}{l} < n$  elements.

In order to have an unisolvant linear system of equations, it is required to additionally enforce the matching conditions

$$\sum_{j=1}^n \lambda_j p_k(\mathbf{x}_j) = 0, \quad k = 1, \dots, s.$$

This approach leads to the linear system of equations determining the  $\boldsymbol{\lambda}$  and  $\boldsymbol{\beta}$  coefficients,

$$\begin{bmatrix} A & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\beta} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix}, \quad (20)$$

where  $A$  is the square matrix with elements

$$A_{i,j} = \phi(\|\mathbf{x}_i - \mathbf{x}_j\|), \quad i, j = 1, \dots, n, \quad (21)$$

$P_{ij} = p_j(\mathbf{x}_i)$ ,  $i = 1, \dots, n$ ,  $j = 1, \dots, s$ , is unisolvant on the stencil nodes, and  $\mathbf{f} = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_n)]^T$ . Depending on the RBF considered, the inclusion of polynomials up to a certain degree  $l$  may be necessary to guarantee a well-posed interpolation problem (for a rigorous discussion on unisolvency conditions see [16]).

Therefore, the RBF+poly interpolant (19) can be equivalently written as

$$s(\mathbf{x}) = [\phi^T(\mathbf{x}) \quad \mathbf{p}^T(\mathbf{x})] \begin{bmatrix} A & P \\ P^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix}, \quad (22)$$

where the vectors

$$\phi(\mathbf{x}) = [\phi(\|\mathbf{x} - \mathbf{x}_1\|), \phi(\|\mathbf{x} - \mathbf{x}_2\|), \dots, \phi(\|\mathbf{x} - \mathbf{x}_n\|)]^T$$

and

$$\mathbf{p}((\mathbf{x})) = [p_1(\mathbf{x}), p_2(\mathbf{x}), \dots, p_s(\mathbf{x})]^T,$$

contain a basis for the RBF space and polynomial space  $\Pi_l^d$ , respectively. It follows from (22) that the RBF+poly cardinal functions  $\psi(\underline{\mathbf{x}})$  are determined by the system of equations

$$\begin{bmatrix} \psi(\mathbf{x}) \\ \gamma(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} A & P \\ P^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} \phi(\mathbf{x}) \\ \mathbf{p}(\mathbf{x}) \end{bmatrix}, \quad (23)$$

where  $\gamma(\mathbf{x})$  correspond with the Lagrange multipliers of a certain minimization problem, as we will show later.

If the node distribution is such that  $A$  and  $P$  are both full rank matrices, it was proven in [4] that the exact expressions for the  $\lambda$  and  $\beta$  coefficients can be obtained via block decomposition as

$$\begin{aligned} \beta &= (P^T A^{-1} P)^{-1} P^T A^{-1} \mathbf{f}, \\ \lambda &= A^{-1} \mathbf{f} - A^{-1} P \beta, \end{aligned}$$

leading to the closed-form expression for the RBF+poly interpolant,

$$s(\mathbf{x}) = \mathbf{f}^T [I - W P^T] \hat{\psi}(\mathbf{x}) + \mathbf{f}^T W \mathbf{p}(\mathbf{x}), \quad (24)$$

and to the RBF+poly cardinal functions

$$\psi(\mathbf{x}) = \hat{\psi}(\mathbf{x}) - W \tau(\mathbf{x}), \quad (25)$$

where  $\hat{\psi}(\mathbf{x}) = A^{-1} \phi(\mathbf{x})$ ,  $\tau(\mathbf{x}) = [P^T \hat{\psi}(\mathbf{x}) - \mathbf{p}(\mathbf{x})]$  and  $W = A^{-1} P (P^T A^{-1} P)^{-1}$  is the  $n \times s$  matrix containing as columns the differentiation weights  $\mathbf{w}^{(m)}$  approximating the operator (18) exactly on  $\Pi_l^d$ .

### 3.2 Formulation as best least squares approximation

Following the same approach as in Section 2.2 for MLS approximations, we show in this section that RBF+poly approximations can be formulated as the best approximation to  $f$  in a least squares sense with respect to an appropriately defined inner product. To do so, assume that the node distribution is such that  $P$  is unisolvant and matrix  $A$  is positive definite. In this case, we can define the inner product as the bilinear form

$$\langle f, g \rangle_{A^{-1}} = \mathbf{f}^T A^{-1} \mathbf{g} = \sum_{i,j=1}^n f(\mathbf{x}_i) g(\mathbf{x}_j) a_{ij}^{-1}, \quad (26)$$

where  $a_{ij}^{-1}$  are the elements of the inverse of matrix  $A$  defined in (21). With the norm induced by (26), the RBF approximation (24) minimizes the functional

$$J = \|f - s\|_2 = \sum_{i,j=1}^n [f(\mathbf{x}_i) - s(\mathbf{x}_i)] [f(\mathbf{x}_j) - s(\mathbf{x}_j)] a_{ij}^{-1}.$$

To proof this, enforce  $\nabla_{\boldsymbol{\lambda}, \boldsymbol{\beta}} J = \mathbf{0}$ , from which follows the system of equations

$$A\boldsymbol{\lambda} + P^T \boldsymbol{\beta} = \mathbf{f}. \quad (27)$$

Since  $P^T \boldsymbol{\lambda} = \mathbf{0}$  by definition, multiplication by  $P^T A^{-1}$  leads to the normal equations

$$(P^T A^{-1} P) \boldsymbol{\beta} = P^T A^{-1} \mathbf{f}, \quad (28)$$

determining the  $\boldsymbol{\beta}$  coefficients. This can be written as a function of the inner product (26) as

$$\sum_{j=1}^l \beta_j \langle p_j, p_k \rangle_{A^{-1}} = \langle f, p_k \rangle_{A^{-1}}, \quad k = 1 \dots, s.$$

Equations (27) and (28) uniquely determine the  $\boldsymbol{\lambda}$  and  $\boldsymbol{\beta}$  coefficients, from which follows the RBF interpolant (24).

### 3.3 Formulation as quadratic minimization problem

RBF approximations with polynomial augmentation can also be interpreted as the solution of an equally-constrained minimization problem. Writing the interpolant (19) as a function of the cardinal functions,

$$s(\mathbf{x}) = \sum_{i=1}^n f(\mathbf{x}_i) \psi_i(\mathbf{x}),$$

it can be shown that the functions  $\psi_i(\mathbf{x})$  minimizing the functional

$$\frac{1}{2} \psi^T(\mathbf{x}) A \psi(\mathbf{x}) - \psi^T(\mathbf{x}) \phi(\mathbf{x})$$

subject to the polynomial reproduction constraints

$$P^T \psi(\mathbf{x}) = \mathbf{p}(\mathbf{x}),$$

are given by system of equations (23).

This can be proven as follows. By constructing the Lagrangian

$$L(\psi, \gamma) = \frac{1}{2} \psi^T A \psi - \psi^T \phi + \gamma^T [P^T \psi - p],$$

where  $\gamma$  are the Lagrange multipliers, one realizes that is convex and has a unique minimum (see [19] for details). Enforcing the condition  $\nabla_{\psi, \gamma} L = \mathbf{0}$ , leads to the system of equations (23). Moreover, if the node distribution is such that  $A$  and  $P$  are both full rank, the solution is

$$\begin{aligned}\gamma(x) &= (P^T A^{-1} P)^{-1} [P^T A^{-1} \phi(x) - p(x)], \\ \psi(x) &= A^{-1} \phi(x) - A^{-1} P \gamma(x),\end{aligned}$$

from which follows the closed-form expression for the RBF+poly cardinal functions (25).

### 3.4 RBF+poly cardinal functions

Unlike MLS, RBF+poly approximations are always interpolatory. Consider the case in which  $A$  and  $P$  are full rank matrices. The corresponding interpolant is equal to

$$s(x) = f^T \hat{\psi}(x) - f^T W \tau(x). \quad (29)$$

As pointed out in [4, 5], the pure RBF part of the approximation  $f^T \hat{\psi}(x)$  interpolates the data, while the term  $-f^T W \tau(x)$ , which is zero at all data points, acts as a correction term. It accounts for the polynomial reproduction of the combined RBF+poly interpolant, maintaining the convergence order (in the sense of being exact for polynomials up to specified degree). Remarkably, this approach overcomes the harmful edge oscillations related with the use of high polynomial degrees as found in [6] and analyzed further in [4, 5].

The left column in Figure 2 displays 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  $16$  (rows). They are split according to equation (25) into pure PHS cardinal function  $\hat{\psi}(x)$  (middle column) and correction term  $-W \tau(x)$  (right column). For  $n = 8$ , the approximation is totally polynomial (as the degree satisfy  $l = n - 1$ ) and the cardinal function features large edge oscillations (equivalent to Lagrange's interpolant). Increasing the stencil size for a fixed polynomial degree  $l$  makes the magnitude of the correction term to decrease. The RBF+poly approximation becomes then dominated by the RBF part. If the RBFs display a non-oscillatory behavior near the boundary (as it is the case here for cubic PHS), the Runge phenomenon is removed and high order of accuracy is maintained (in the sense of being exact for polynomials up to the specified degree  $l$ ).

### 3.5 Approximation error

RBF+poly approximations reproduce any polynomial up to degree  $l$  in  $d$  variables exactly. This can be obtained following the same approach as in [4], where it is used the fact that matrix  $W = A^{-1} P (P^T A^{-1} P)^{-1}$  in the RBF interpolant (24) exactly differentiates the augmented polynomial terms from  $\Pi_l^d$ .

Alternatively, here we use the same approach as in Section 2.5 for MLS approximations. Based on the polynomial reproduction properties of the cardinal functions, the approximation error becomes

$$f(x) - s(x) = \sum_{k=s+1}^{\infty} L_k[f(x_0)] [p_k(x - x_0) - p_k^T \cdot \psi(x)].$$

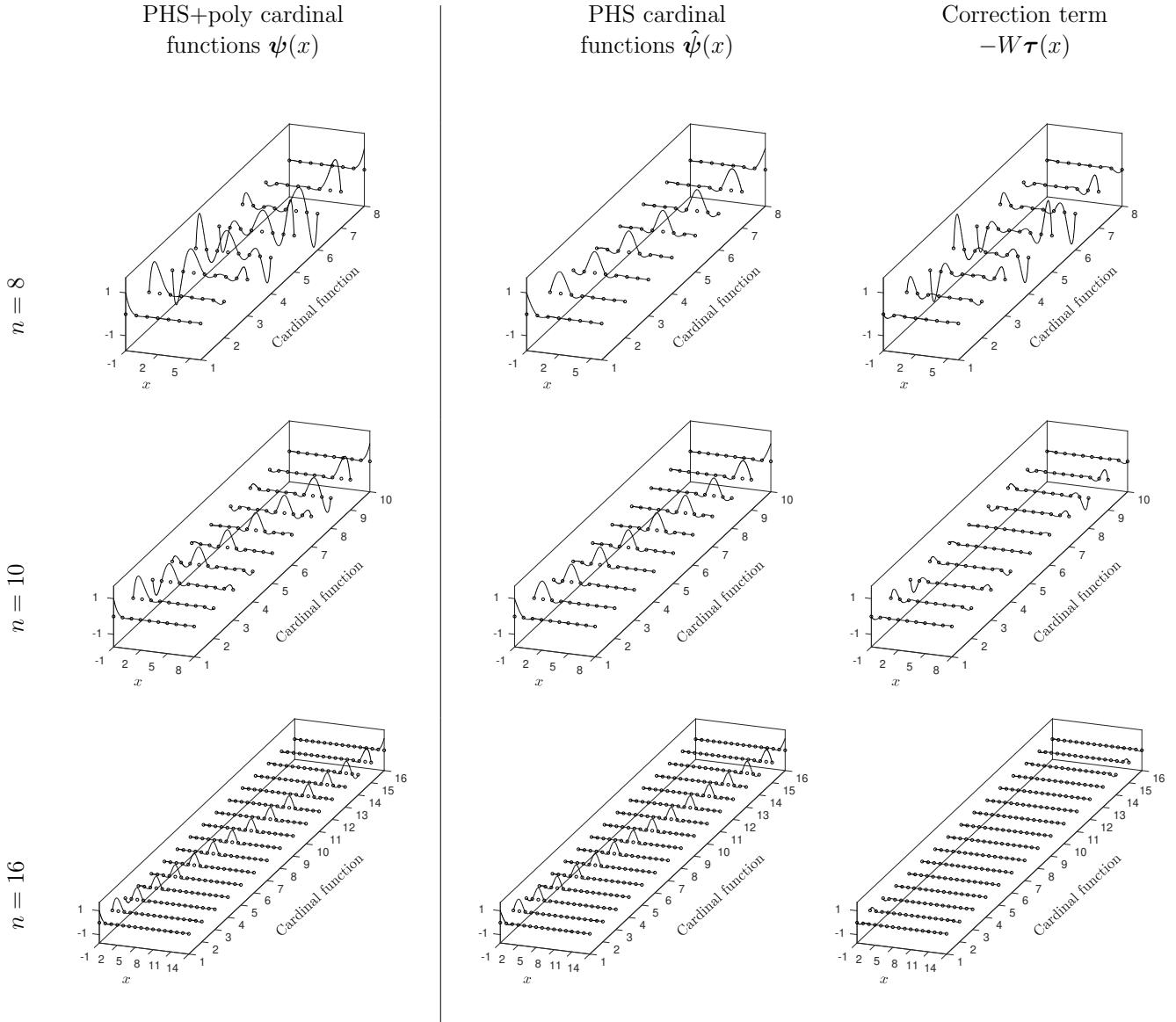


Figure 2: PHS+poly cardinal functions (left column) when using PHS  $r^3$  augmented with polynomial terms up to 7th degree over the 1-D stencil  $x = -1, 0, 1, 2, 3, \dots, n-2$ , for  $n = 8, 10, 16$  (rows). According to equation (25), they are split into the contribution of pure PHS cardinal functions  $\hat{\psi}(x)$  (middle column) and the correction term  $-W\tau(x)$  (right column) as displayed in [4, 5].

Substitution of the cardinal functions (25) yields

$$f(\underline{x}) - s(\underline{x}) = \sum_{k=s+1}^{\infty} L_k[f(\underline{x}_0)] \left[ \left( p_k(\underline{x} - \underline{x}_0) - \mathbf{p}_k^T \hat{\psi}(\underline{x}) \right) + \mathbf{p}_k^T W \boldsymbol{\tau}(x) \right],$$

which can be bounded as

$$\|f(\mathbf{x}) - u(\mathbf{x})\|_{\infty} \leq Ch^{l+1} \max_{\mathbf{x} \in \Omega} |L^{(l+1)}[f(\mathbf{x})]|.$$

All Taylor terms of  $f$  up through degree  $l$  are treated exactly by the supplementary polynomials. Therefore, the convergence order is  $O(h^{l+1})$ . The RBFs have here a similar role as the weight function in MLS approximations, acting on the remainder terms of the expansion.

#### 4 Numerical comparison: MLS vs PHS+poly

The previous analysis has revealed several similarities between MLS and RBF+poly approximations. For instance, MLS approximations, which can be written as

$$u(\mathbf{x}) = \mathbf{f}_p^T G_w^{-1} \mathbf{p}(\mathbf{x}), \quad (30)$$

where

$$\mathbf{f}_p^T = \mathbf{f}^T W(\mathbf{x}) P = [\langle f, p_1 \rangle_{w(\mathbf{x})}, \dots, \langle f, p_s \rangle_{w(\mathbf{x})}],$$

is the projection of the data  $f$  onto the polynomial space  $\Pi_l^d$ , and

$$G_w = (P^T W(\mathbf{x}) P) = \langle p_j, p_k \rangle_{w(\mathbf{x})}, \quad j, k = 1, \dots, N,$$

is the Gram matrix, are purely polynomial. The convergence order is determined by the polynomial degree  $l$ , and the role of the weight function is to make the approximant to interpolate the data.

RBF+poly approximations (29) combine RBFs and polynomials. The RBF part in the approximation ensures the interpolation of the data, while the remainder acts as a correction term enforcing the approximation to be exact up to the augmented polynomial degree  $l$ . Under refinement, the polynomial part takes over and treats exactly all Taylor terms of the target function  $f$  [4]. In this sense, we can also say that RBF+poly approximations are polynomial up to the augmented degree  $l$ . With the same notation as in (30), RBF+poly approximations can be written as

$$s(\mathbf{x}) = \mathbf{f}_p^T G_{\phi}^{-1} \mathbf{p}(\mathbf{x}) + \underbrace{f_{\phi}(\mathbf{x}) - \mathbf{f}_p^T G_{\phi}^{-1} \mathbf{p}_{\phi}(\mathbf{x})}_{\perp \Pi_l^d}, \quad (31)$$

where

$$\begin{aligned} f_{\phi}(\mathbf{x}) &= \mathbf{f}^T A^{-1} \phi(\mathbf{x}) = \langle f, \phi(\mathbf{x}) \rangle_{A^{-1}}, \\ \mathbf{f}_p^T &= \mathbf{f}^T A^{-1} P = [\langle f, p_1 \rangle_{A^{-1}}, \dots, \langle f, p_s \rangle_{A^{-1}}], \\ \mathbf{p}_{\phi}(\mathbf{x}) &= P^T A^{-1} \phi(\mathbf{x}) = [\langle p_1, \phi(\mathbf{x}) \rangle_{A^{-1}}, \dots, \langle p_s, \phi(\mathbf{x}) \rangle_{A^{-1}}]^T, \end{aligned}$$

are projections into different subspaces and  $G = (P^T A^{-1} P) = \langle p_j, p_k \rangle_{A^{-1}}$ ,  $j, k = 1, \dots, N$  denotes the Gram matrix. Observe that the RBFs span the subspace orthogonal to the polynomial space  $\Pi_l^d$ . This is the key to understanding the differences between the two methods observed numerically in the following tests.

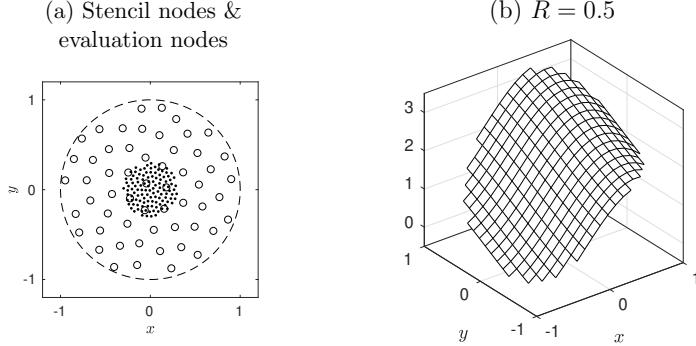


Figure 3: (a) Stencil used for  $n = 56$  nodes (small circles) and set of evaluation points to asses accuracy (dots). (b) Test function (32) displayed over  $x^2 + y^2 \leq 1$  for  $R = 0.5$ .

#### 4.1 Function approximation and accuracy

The aim of this section is to explore how the accuracy of MLS approximations compares to PHS+poly. We have considered a numerical experiment similar to the one used in [19], where the test function

$$f(x, y) = 1 + \sin(4Rx) + \cos(3Rx) + \sin(2Ry) \quad (32)$$

is approximated over a scattered stencil formed by  $n$  minimum energy-like nodes. The left panel of Figure 3 displays the stencil used for  $n = 56$  (small circles), and the points where the interpolant is evaluated near the stencil center (dots). The right panel shows the test function (32) over the stencil for  $R = 0.5$ .

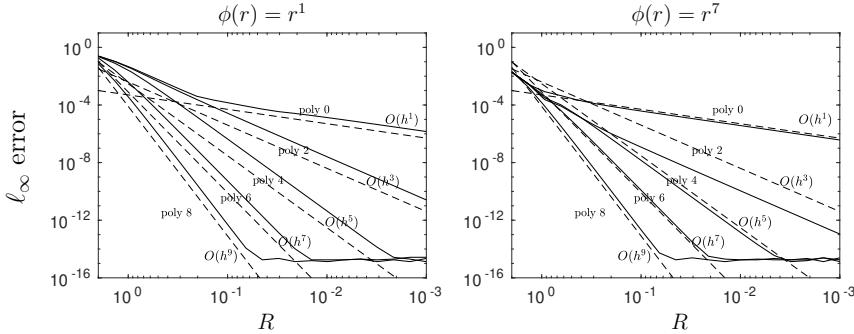
As first experiment, we have computed the approximation error over this stencil when using PHS  $r^m + \text{poly}$ ,  $m = 1, 7$ , and MLS,  $\alpha = 0, 1, 12$ , for different polynomial degrees as the parameter  $R$  goes to zero and the test function behaves more as a tilted plane. The worst error encountered is displayed in Figure 4 as a function of  $R$ .

In agreement with our analysis, the convergence order is determined by the degree  $l$  of polynomial terms. The PHS degree  $m$  does not affect the convergence order, but shifts the error vertically, as shown when going from  $\phi(r) = r^1$  to  $\phi(r) = r^7$ . The weight function has a similar effect in MLS. There is little gain when making the approximant interpolatory, increasing  $\alpha$  from 0 to 1. The difference becomes much more significant for  $w(r) = 1/r^{12}$ , acting in the residual as a multiplying constant (see Section 2.5).

In this second part of the experiment, we want to analyze the dependence of the stencil size  $n$  and the weight function degree  $\alpha$  (MLS) or PHS degree  $m$  (PHS+poly) on the accuracy and conditioning of the methods. To do so, we have fixed the parameter  $R = 0.5$ , and computed the worst error encountered, as well as the condition number of the corresponding systems (12) and (20), for the stencil sizes  $n = \binom{l+d}{l}, \dots, 150$  and coefficients  $\alpha = 0, 1, \dots, 20$  (MLS) or  $m = 1, 3, \dots, 11$  (PHS). The results obtained are displayed in Figure 5 for different polynomial degrees  $l$  (rows). Several observations can be made:

- (i) The accuracy of PHS+poly approximations improves significantly as both  $n$  and/or the PHS

(a) PHS augmented with polynomials



(b) Moving Least Squares

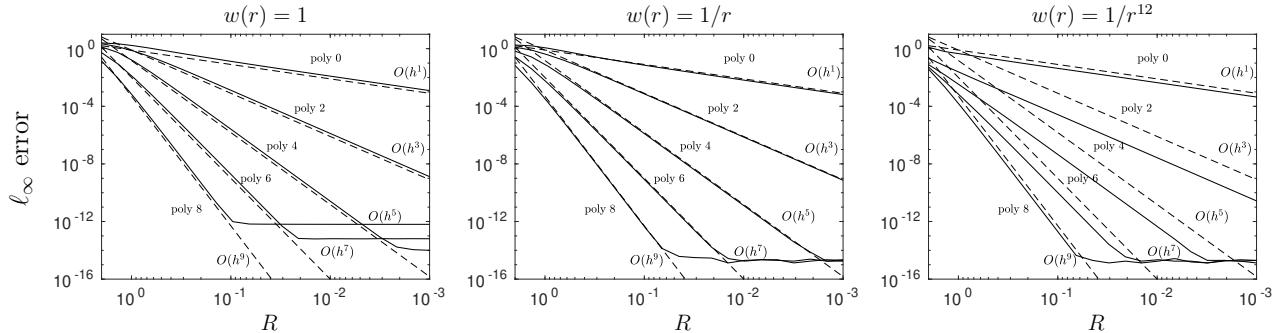


Figure 4: The worst error encountered under refinement when approximating function (32) over the stencil shown in Figure 3, using (a) PHS  $r^m$  augmented with polynomials and (b) Moving Least Squares, for different polynomial degrees  $l$ . To compute the convergence under refinement, the parameter  $R$  in the test function (32) is decreased towards 0.

degree  $m$  increase. By contrast, the effect of the stencil size  $n$  or degree  $\alpha$  on the accuracy in MLS approximations is less significant.

- (ii) For low polynomial degrees (in this case,  $l \leq 4$ ), PHS+poly is significantly more accurate than MLS. Such a difference, however, vanishes for higher polynomial degrees. This behavior can be explained from our previous observation. While MLS approximations (30) only span the polynomial space  $\Pi_l^d$ , PHS+poly (31) combines two functional spaces, RBFs and polynomials. The approximation is performed by the RBFs (interpolation), but is corrected to exactly reproduce polynomials of degree  $l$  (correction term). For low  $l$ , RBF+poly has mainly RBF nature. While MLS can barely approximate the function in this case, the RBFs do span the functional space and the accuracy improves as a function of  $n$  and  $m$ . For higher polynomial degrees, the polynomials take over and the differences are not so relevant.
- (iii) The dependence of the stencil size  $n$  and weight function degree  $\alpha$  or PHS degree  $m$  on the conditioning of the collocation systems (12) and (20) is similar in the two cases.

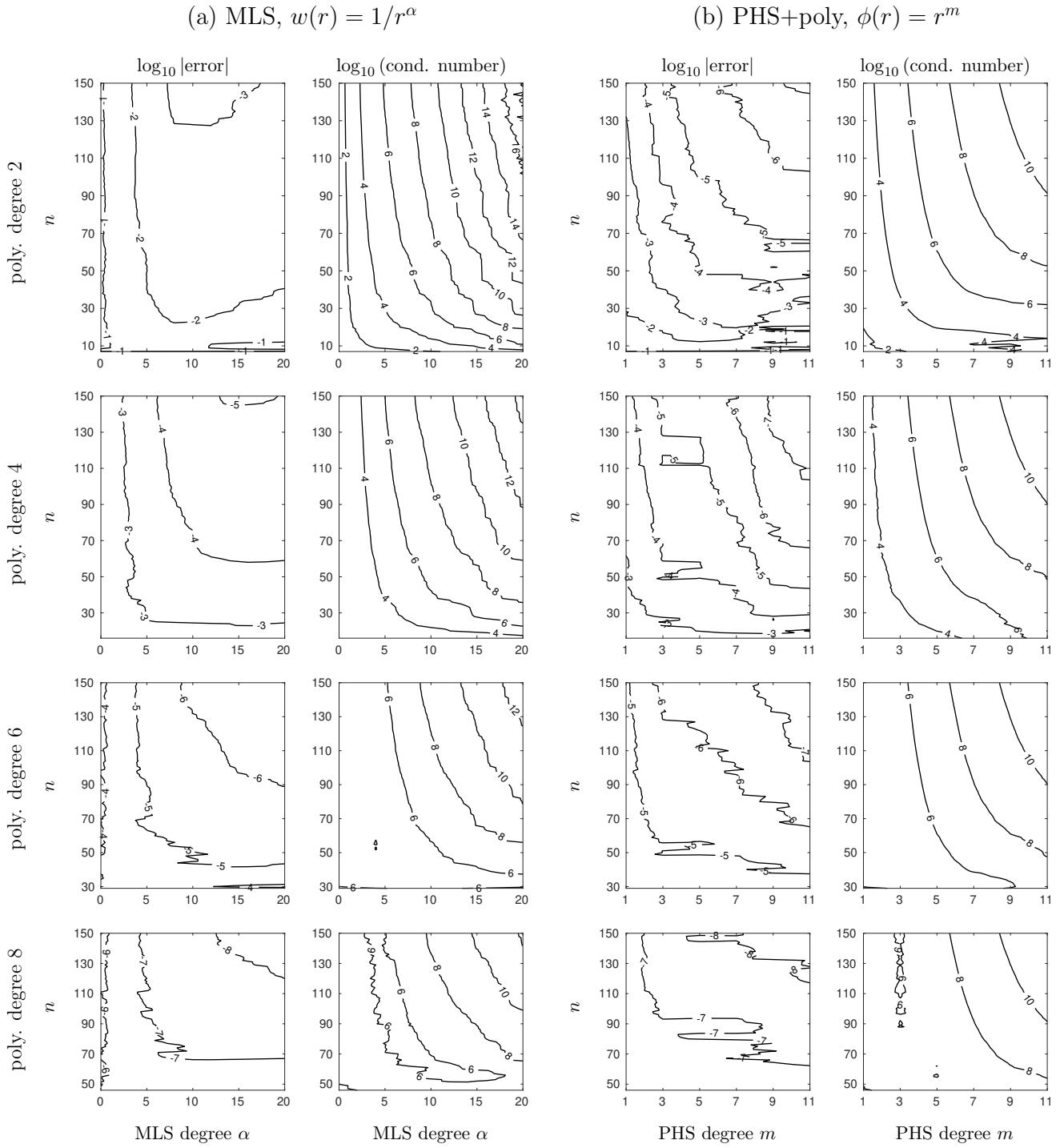
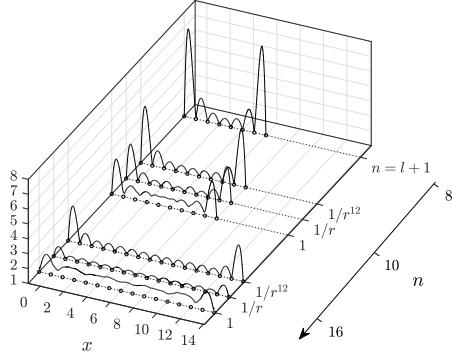


Figure 5: (a) Moving Least Squares, and (b) PHS augmented with polynomials. In each case, it is shown the accuracy ( $\log_{10}$  error) and the conditioning ( $\log_{10}$  of the condition number) for the systems (12) or (20), as a function of the stencil size  $n$  ( $y$ -axis) and MLS degree  $\alpha$  / PHS degree  $m$  ( $x$ -axis), for different polynomial degrees  $l$  (rows). Plotting the level curves utilizes interpolation as the stencil sizes  $n$  and MLS degrees  $\alpha$  can only take on integer values, and the PHS degrees  $m$  odd integers.

MLS,  $w(r) = 1/r^\alpha$



PHS+poly,  $\phi(r) = r^m$

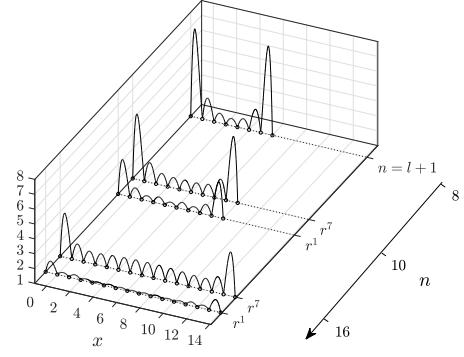


Figure 6: Lebesgue functions for MLS,  $w(r) = 1/r^\alpha$ ,  $\alpha = 0, 1, 12$ , (left) and PHS+poly,  $\phi(r) = r^m$ ,  $m = 1, 7$ , (right) when using polynomials of degree  $l = 7$  and increasingly large stencil sizes,  $n = l + 1, 10$  and  $16$  ( $y$ -axis) over the stencil  $x = -1, 0, 1, 2, \dots, n - 2$ . Dotted line represents optimal value equals one.

## 4.2 Lebesgue functions and differentiation weights in 1D

It was found in [6] that PHS+poly approximations can be augmented with high degree polynomials  $l$  (becoming  $O(h^{l+1})$  accurate) without the adverse Runge Phenomena effects by simply increasing the stencil size for a fixed polynomial degree. The mechanism underlying this feature, which was briefly described in Section 3.4, was found in [4] and analyzed further in [5]. The goal of this section is to analyze the performance of MLS approximations near domain boundaries, with a special focus on the comparison against PHS+poly.

We consider the same 1-D interpolation problem proposed in [6], which requires us to compute the cardinal functions over a 1-D stencil  $x = -1, 0, 1, 2, \dots, n - 2$ , for  $n = 8, 10$  and  $16$ , using MLS and cubic PHS for a fixed polynomial degree  $l$ . The MLS cardinal functions with 5th degree polynomials,  $n = 10$  and  $\alpha = 0, 1$  and  $12$  were displayed previously in Figure 1, and the cardinal functions for cubic PHS+7th degree polynomials were shown in Figure 2. Typically, the stability of an interpolation method can be visualized through the Lebesgue function,

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

which will feature low amplitude oscillations if the method is free of edge anomalies. Following the same approach as in [5], Figure 6 shows the Lebesgue functions for MLS,  $w(r) = 1/r^\alpha$ ,  $\alpha = 0, 1, 12$ , (left), and PHS+poly,  $\phi(r) = r^m$ ,  $m = 1, 7$ , (right), when using polynomials of degree  $l = 7$  and increasingly large stencil sizes,  $n = l + 1, 10$  and  $16$  ( $y$ -axis) over the stencil  $x = -1, 0, 1, 2, \dots, n - 2$ .

For  $n = 8$ , they are purely polynomial (the degree satisfies  $l = n - 1$ ), featuring equivalently large edge oscillations as the Lagrange cardinal functions. Increasing the stencil size for PHS+poly approximations reduces the magnitude of the correction term (see Section 3.4). As a result, the approximation recovers the PHS dominated edge behavior. It is well-known that low degree PHS  $r^m$  do not suffer from edge oscillations (as it minimizes a functional over all possible interpolants

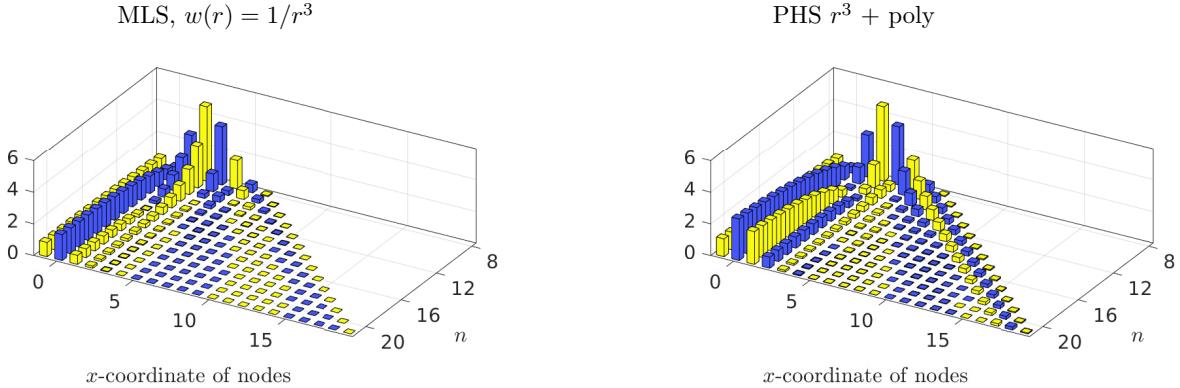


Figure 7: Magnitude of the differentiation weights approximating the second derivative centered at  $x_0 = 0$  for MLS  $w(r) = 1/r^3$  (left) and RBF+poly  $\phi(r) = r^3$  (right) over the stencil  $x = -1, 0, 1, 2, \dots, n - 2$ , for increasingly large stencil sizes  $n$  and fixed polynomial degree  $l = 7$ . Blue corresponds to negative differentiation weights and yellow to positive. In each case, the pattern  $[1 - 2 1]$  is recovered as the stencil size increases.

[36]). Therefore, PHS+poly bypasses the Runge Phenomena effects while maintaining high order of accuracy (in the sense of being exact for polynomials up to the specified degree  $l$ ). This is observed in the figure for  $\phi(r) = r^7$ . The case  $\phi(r) = r^7$  presents the same underlying mechanism as  $n$  increases, but with larger edge oscillations.

By contrast, MLS is purely a polynomial method. The approximation of the data is enforced through a weighted least squares, where the weight functions defining the inner product plays a crucial role in the interpolatory properties of the approximant (as discussed in Section 2). Observe in Figure 1 that for  $\alpha = 0$ , the generating functions are equivalent to classical polynomial least squares and do not interpolate the data (the corresponding Lebesgue functions in Figure 6 are different to 1 at all data sites). However, and this is the key feature, the edge oscillations decrease its amplitude for increasingly large stencil sizes  $n$ , going from pure polynomial edge oscillations to be almost free of edge anomalies. Increasing the value  $\alpha$  maintains this feature, while making the approximant to also interpolate the data. This is shown in Figure 6 when using  $w(r) = 1/r$  and  $w(r) = 1/r^{12}$ , which displays a similar behavior as  $r^1$  and  $r^7$  for PHS+poly interpolants.

Figure 7 shows the differentiation weights approximating the second derivative at  $x = 0$ , i.e.  $w_i = \psi''_i(0)$ ,  $i = 1, \dots, n$ , for MLS  $w(r) = 1/r^3$  (left) and cubic PHS+poly (right), for  $l = 7$  and increasingly large stencil sizes  $n$  ( $y$ -axis). As found in [4, 5, 6] for PHS+poly, the weights recover the pattern  $[1 - 2 1]$  centered at  $x = 0$  (as expected for second order standard FD) with an accuracy  $O(h^{l-1})$ . Remarkably, the underlying least squares fitting in MLS approximations has a similar effect on the pattern of weights. As the stencil size increases, the edge oscillations in the Lebesgue functions decrease and the differentiation weights recover the largest values around the stencil center, analogously to the PHS+poly case.

Figure 8 shows the effect of the weight function degree  $\alpha$  (MLS) or the PHS degree  $m$  (PHS+poly) on the differentiation weights approximating the second derivative at  $x = 0$  for  $n = 16$  and  $l = 7$ . The red line highlights the values of  $\alpha$  or  $m$  for which the method recovers the pattern  $[1 - 2 1]$  centered at the evaluation point  $x = 0$ . Observe that the PHS degree  $m$  has a stronger effect than the MLS degree  $\alpha$ . For  $n = 16$ , only  $m = 3$  and  $5$  display such pattern of weights. In MLS,

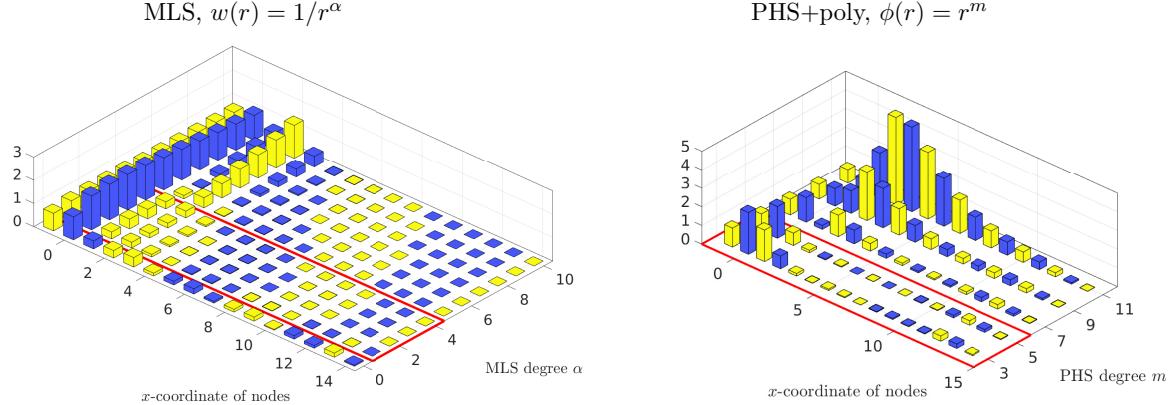


Figure 8: Magnitude of the differentiation weights approximating the second derivative centered at  $x_0 = 0$  over the stencil  $x = -1, 0, 1, 2, \dots, 14$  ( $n = 16$ ) and polynomial degree  $l = 7$ , as a function of the MLS degree  $\alpha = 0, 1, \dots, 10$  (left) and PHS degree  $m = 3, \dots, 11$  (right). Blue corresponds to negative differentiation weights and yellow to positive. The red line shows the values for  $\alpha$  or  $m$  for which the method recovers the pattern  $[1 \ -2 \ 1]$  centered at  $x = 0$ .

any value  $\alpha = 1, 2, 3$  and  $4$  is found to work. Also, the magnitude of the weights is smaller and increasing the value of  $\alpha$  for a fixed stencil size  $n$  has a weaker effect.

### 4.3 Lebesgue functions and differentiation weights in 2D

It was shown numerically in [5] that this mechanism holds in any space dimension for PHS+poly. In order to compare the performance against MLS, we have computed the Lebesgue functions and differentiation weights approximating the Laplacian over a near one-sided stencil in 2-D. Figure 9 shows the corresponding results when using MLS (top) and cubic PHS (bottom) with 4th degree polynomials and stencil sizes  $n = 16$  and  $n = 31$ .

Observe that the case  $n = 16$  is very close to the pure polynomial case  $n = \binom{4+2}{4} = 15$ . As a consequence, the Lebesgue functions feature strong edge oscillations for both MLS and PHS+poly. For  $n = 31$  (roughly twice number of polynomial terms), the Lebesgue functions are free of edge anomalies not only for PHS+poly (as reported in [5]), but also for MLS. The corresponding differentiation weights inherit this behavior. For  $n = 16$ , the pattern of weights (computed as the Laplacian of the cardinal functions) is far from being positive definite. By increasing the stencil size, it recovers a structure where the central weight has the largest magnitude at the evaluation point, surrounded by oscillatory weights whose magnitude decays with the distance. The accuracy of these stencils is also tested under refinement by approximating the Laplacian of function (32) as a function of  $R$  for  $n = 16, 31$  and  $61$ . 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.

Traditionally, the negative boundary effects are overcome by clustering the nodes towards the boundaries. However, following this approach over irregular domains might not be trivial. One key result here is finding that these two mesh-free approaches have intrinsic mechanisms to overcome the edge oscillations, which are independent on the node distributions. In the following section, we analyze them further through an elliptic PDE problem.

## Lebesgue Functions

## Differentiation weights for the Laplacian

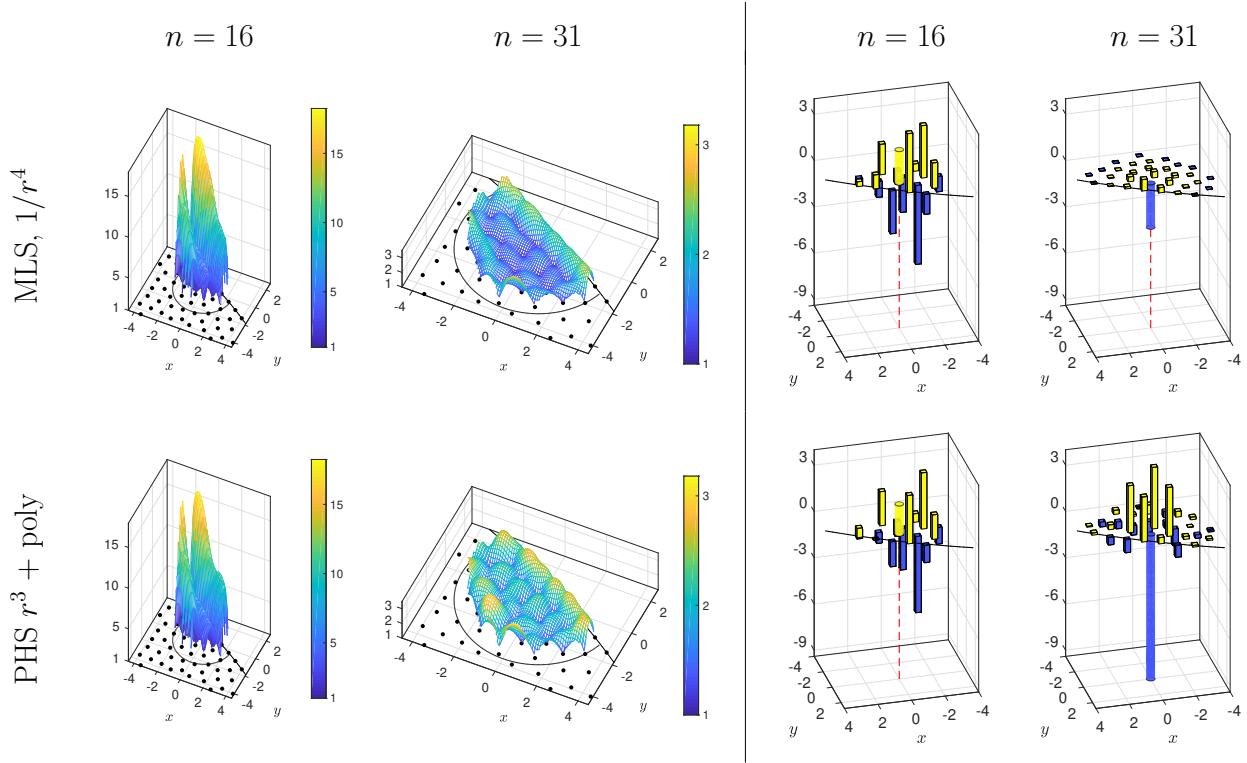


Figure 9: Left panel: Lebesgue functions for a near-one sided stencil close to a domain boundary in 2-D, when using MLS  $w(r) = 1/r^4$  (top) and cubic PHS (bottom) with 4th degree polynomials for  $n = 16$  and  $31$  (columns). Note the vertical scale is the same in all subplots, and the factor 5 difference in  $n = 16$ . Right panel: corresponding differentiation weights approximating the Laplacian at  $(x, y) = (0, 0)$ . The red dashed line marks the location of the weight at the stencil center. Blue corresponds to negative differentiation weights and yellow to positive.

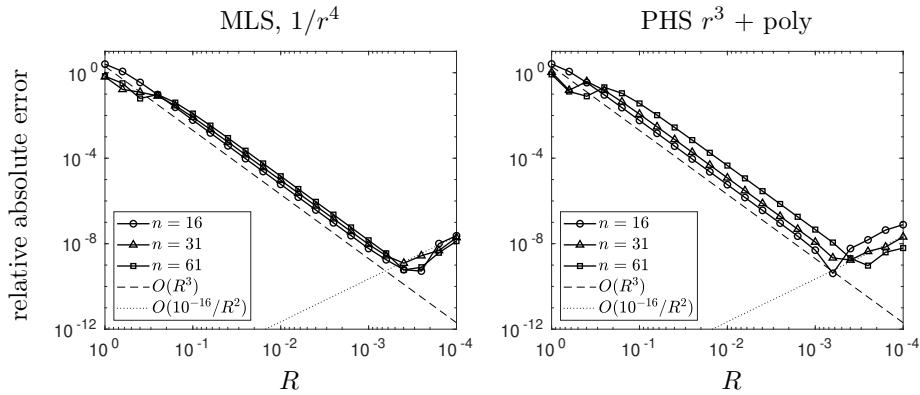


Figure 10: Relative absolute error approximating the Laplacian of function (32) over the stencil in Figure 9 as a function of  $R$ . The thin dashed lines represent the convergence order  $O(R^3)$  and the dotted line marks the round-off limit  $10^{-16}/R^2$ .

#### 4.4 A PDE problem to compare performances

The use of PHS+poly for the numerical solution of elliptic PDEs was tested in [6]. This was the first work to point out the advantageous behavior of PHS+poly approximations near domain boundaries. In particular, it was found the mechanism discussed in the previous section (controlling the recovery of the pattern of weights as a function of the stencil size  $n$ ), and how it improves the diagonal dominance of the differentiation matrices without the use of any boundary treatment such as ghost nodes.

The aim of this section is to compare its performance against the analogous feature found for MLS. To do so, we have considered the same PDE problem as in [6], i.e.

$$\begin{cases} \Delta u = f(x, y), & \text{in } \Omega \\ u = g(x, y), & \text{on } \partial\Omega \end{cases} \quad (33)$$

on the unit disc  $\Omega = \{(x, y), x^2 + y^2 \leq 1\}$ , where  $f(x, y)$  and  $g(x, y)$  are computed from the test function  $u(x, y) = \sin[10(x + y)]$ . The same scattered nodes over the unit disc are employed here. The discretized equations are solved using BiCGSTAB with incomplete LU factorization as preconditioner (with 0 level of fill in) and reverse Cuthill-McKee ordering.

Figure 11 shows the relative  $\ell_\infty$  norm error as a function of the stencil size and polynomial degree, when solving (33) using MLS  $w(r) = 1/r^4$  (left) and cubic PHS+poly (right) for  $h \approx 0.05, 0.025$  and  $0.01$  (rows). The dashed curves  $n = \binom{l+2}{l}$  represent the case where the stencil size  $n$  is equal to the number of polynomial terms  $s$ , while the dash-dotted line shows twice the number of polynomial terms  $2n$ . Plotting the level curves utilizes interpolation as the stencil sizes  $n$  and polynomial degrees can only take on integer values. The vertical dotted lines are displayed here as a reference to visualize the integer polynomial degrees on which the solution of (33) is computed.

Regarding accuracy, observe in Figure 11 a similar behavior as in Figures 4 and 5 from Section 4.1. MLS is less accurate than PHS+poly for low degree polynomials (up to  $l = 4$ ). For  $l = 4$ , the two methods perform similarly when  $n \lesssim 70$ . Increasing the stencil size further reduces the MLS accuracy. For  $l = 5$  and  $6$ , the accuracy is similar. The key difference is found for higher polynomial degrees ( $l > 6$ ), for which MLS appears to stop working. To understand this issue, Figure 12 displays the eigenvalue spectra of the Laplacian differentiation matrices for  $h \approx 0.025$ . Observe that:

- For  $l = 6$  and stencil size  $n = 30$ , which is very close to the pure polynomial case  $n = \binom{l+d}{l}$ , both MLS and cubic PHS+poly eigenvalues are widely scattered over the complex plane (some of them with positive real part). By increasing the stencil size to  $n = 60$ , which corresponds to almost twice as many nodes as polynomials, the eigenvalues move all to the left half of the complex plane, lying very close to the real axis (specially for PHS+poly) as expected for the Laplacian.
- For the case  $l = 8$  and  $n = 50$  (which is also very close to the pure polynomial case), the eigenvalue spectra is spread over the complex plane. The key difference here is that increasing the stencil size to  $n = 95$  only recovers the stable eigenvalue spectra for cubic PHS+poly. MLS displays a 'bad' eigenvalue distribution, with some of them still in the right half of the complex plane.

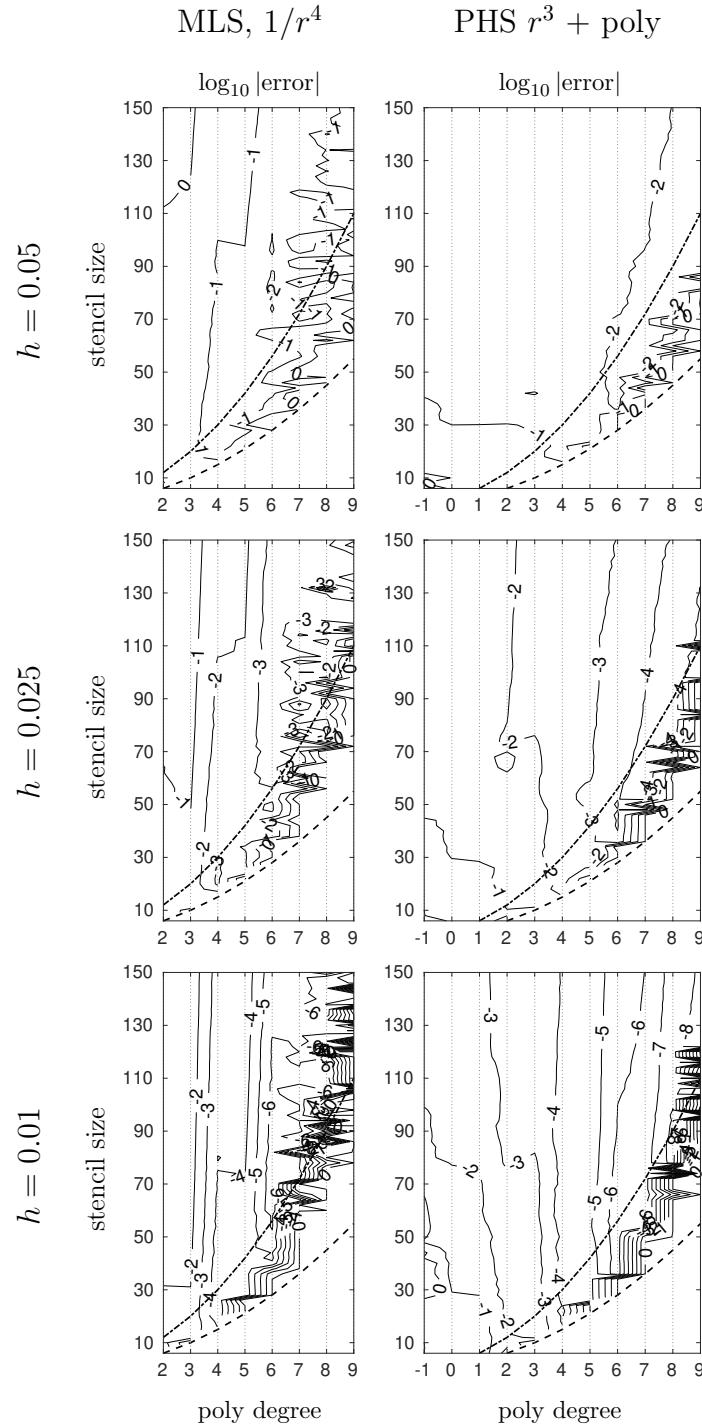


Figure 11: Accuracy ( $\log_{10}$  relative  $\ell_\infty$ -norm error) when using MLS (left) and cubic PHS+poly (right) to solve (33) as a function of the stencil size  $n$  and degree of polynomial terms  $l$ . The rows show different resolutions. The dashed line shows the stencil size where  $n = \frac{l+2}{l}$  which would be equivalent to using polynomials only. The dotted line shows twice the number of polynomial terms  $2n$ . Plotting the level curves utilizes interpolation as the stencil sizes  $n$  and polynomial degrees can only take on integer values. The vertical dotted lines are displayed to visualize the polynomial degrees along which the accuracy is assessed.

To understand this behavior, we may look at the patterns of weights discretizing the Laplacian (which determine the differentiation matrix). Figure 13 displays the pattern of weights over a (near-one sided) stencil next to the domain boundary for both MLS and cubic PHS+poly with  $n = 50$  and  $95$ . Unlike cubic PHS+poly, MLS cannot recover the ‘near-positive definite’ pattern centered at the evaluation point  $(x, y) = (0, 0)$  for  $n = 95$ . This issue, acting arbitrarily on the stencils near the domain boundary, makes the MLS differentiation matrix (assembled with all the weights over the whole domain) to miss some properties (such as diagonal dominance). As a result, the eigenvalue spectra spreads over the complex plane, and the iterative solver fails to converge (see Figure 11 for polynomial degrees  $l > 6$ ).

PHS+poly approximations seems oblivious to this adverse effect due to the inclusion of PHS-type RBFs in the process of generating RBF-FD weights. As found in [4] and investigated further in [5], PHS+poly cardinal functions (25) can be decomposed into pure PHS cardinal functions plus a correction term (ensuring the order of accuracy). As the stencil size increases, the correction term decreases in size. The RBF+poly approximations become then dominated by the PHS part (which is specially well behaved at boundaries due to its spline nature), 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). The RBF-FD weights inherit this behavior, guaranteeing the differentiation matrix to preserve some properties and making the eigenvalue spectra to recover the observed stable pattern, even for high degree polynomials.

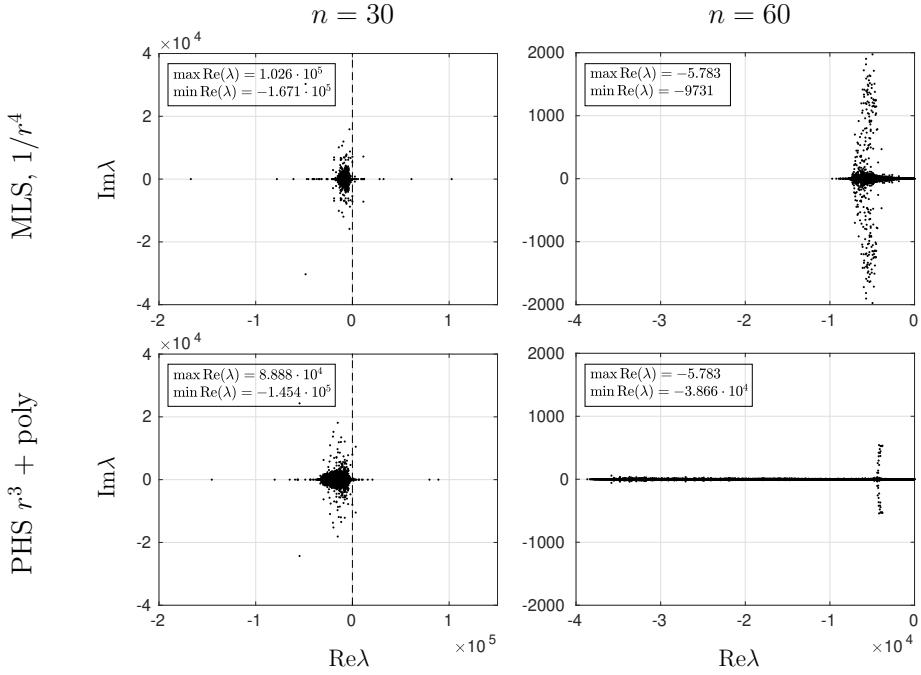
## 5 Conclusions

We have investigated the use of RBF+poly and MLS approximations for scattered data interpolation and derivative approximation. After revisiting the MLS and RBF+poly formulations, we found that under some mild assumptions both can be formulated analogously, either as a equality constrained optimization problem or as a best least squares fitting. In the two cases, the local polynomial reproduction constraint enforces an algebraic convergence order. More striking is the fact that, as previously observed for PHS+poly, MLS can also overcome the edge oscillations (Runge’s phenomenon) by simply increasing the stencil size for a fixed polynomial degree. The underlying mechanism is, however, controlled by a weighted least squares fitting which is found to fail for high polynomial degrees. Overall, PHS+poly approximations perform superior. The inclusion of PHS-type RBFs not only lead to more accurate approximations (specially for low degree polynomials), but can also bypass the edge oscillations more robustly for any polynomial degree. This study potentially opens new opportunities for PHS+poly in areas of application where MLS is currently the preferred choice.

## Acknowledgements

The author would like to thank Bengt Fornberg, who took the time to carefully review this manuscript and make useful comments. This work was supported by Spanish MECD Grant FIS2016-77892-R.

(a) poly degree 6



(b) poly degree 8

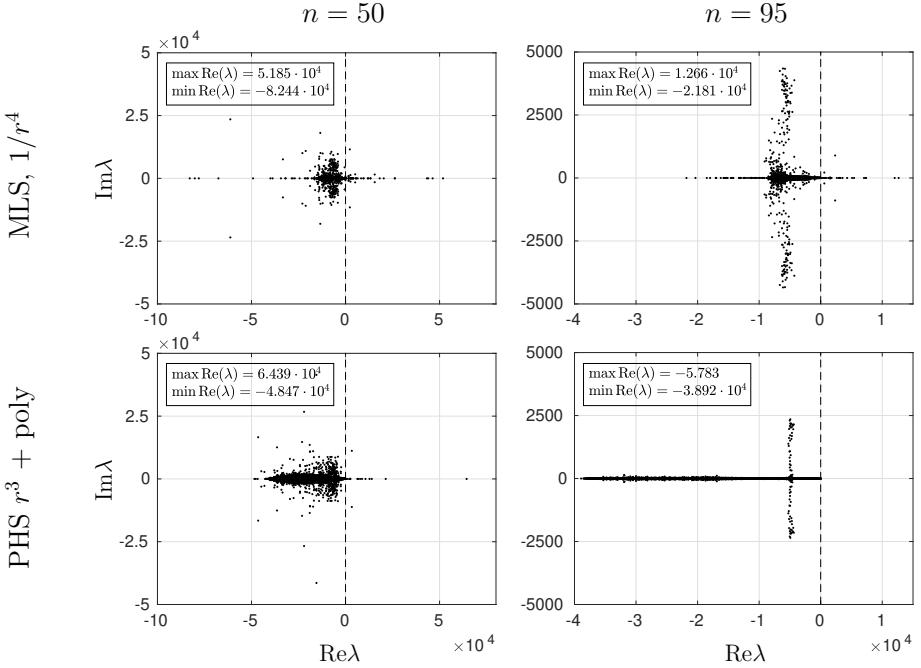


Figure 12: Eigenvalue distribution of the differentiation matrices approximating the Laplacian over the unit disc  $x^2 + y^2 \leq 1$  for  $h \approx 0.025$ , using MLS and cubic PHS with (a) 6th degree polynomials,  $n = 30$  and  $60$ ; and (b) 8th degree polynomials,  $n = 50$  and  $95$ . Note a factor 10 difference in the horizontal and vertical scales when increasing the stencil size.

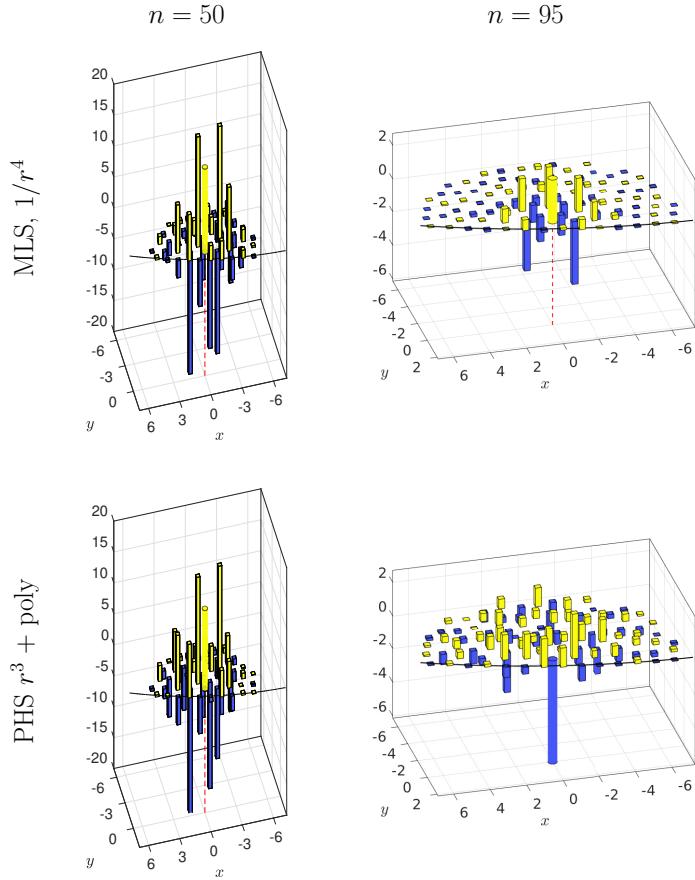


Figure 13: Differentiation weights approximating the Laplacian at  $(x, y) = (0, 0)$  over a near-one sided 2-D stencil close to a domain boundary when using MLS (top) and cubic PHS (bottom) with 8th degree polynomials for stencil sizes  $n = 50$  (left) and  $95$  (right). The red dashed line marks the location of the weight at the stencil center. Blue corresponds to negative differentiation weights and yellow to positive.

## References

- [1] S.N. Atluri, J.Y. Cho, and H-G. Kim, *Analysis of thin beams, using the meshless local Petrov–Galerkin method, with generalized moving least squares interpolations*, Computational Mechanics **24** (1999), no. 5, 334–347.
- [2] G. Backus and F. Gilbert, *The resolving power of gross earth data*, Geophysical Journal of the Royal Astronomical Society **16** (1968), no. 2, 169–205.
- [3] G. A. Barnett, *A robust RBF-FD formulation based on polyharmonic splines and polynomials*, Ph.D. thesis, University of Colorado, Boulder, CO, 2015.
- [4] V. Bayona, *An insight into RBF-FD approximations augmented with polynomials*, Comput. Math. Appl. **77** (2019), no. 9, 2337–2353.
- [5] V. Bayona, N. Flyer, and B. Fornberg, *On the role of polynomials in RBF-FD approximations: III. Behavior near domain boundaries*, J. Comput. Phys. **380** (2019), 378–399.
- [6] 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.
- [7] T. Belytschko, Y. Krongauz, M. Fleming, D. Organ, and W. K. S. Liu, *Smoothing and accelerated computations in the element free Galerkin method*, Journal of Computational and Applied Mathematics **74** (1996), no. 1-2, 111–126.
- [8] L.P. Bos and K. Salkauskas, *Moving least-squares are Backus-Gilbert optimal*, Journal of Approximation Theory **59** (1989), no. 3, 267–275.
- [9] \_\_\_\_\_, *Moving least-squares are Backus-Gilbert optimal*, Journal of Approximation Theory **59** (1989), no. 3, 267–275.
- [10] Philip C Curtis et al., *n-parameter families and best approximation.*, Pacific Journal of Mathematics **9** (1959), no. 4, 1013–1027.
- [11] G. A. Dilts, *Moving-least-squares-particle hydrodynamics – I. Consistency and stability*, International Journal for Numerical Methods in Engineering **44** (1999), no. 8, 1115–1155.
- [12] \_\_\_\_\_, *Moving least-squares particle hydrodynamics II: conservation and boundaries*, International Journal for Numerical Methods in Engineering **48** (2000), no. 10, 1503–1524.
- [13] J. Duchon, *Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces*, Revue française d’automatique, informatique, recherche opérationnelle. Analyse numérique **10** (1976), no. R3, 5–12.
- [14] \_\_\_\_\_, *Splines minimizing rotation-invariant semi-norms in Sobolev spaces*, Constructive theory of functions of several variables, Springer, 1977, pp. 85–100.
- [15] \_\_\_\_\_, *Sur l’erreur d’interpolation des fonctions de plusieurs variables par les  $D^m$ -splines*, RAIRO. Analyse numérique **12** (1978), no. 4, 325–334.
- [16] G. E. Fasshauer, *Meshfree Approximation Methods with MATLAB*, Interdisciplinary Mathematical Sciences - Vol. 6, World Scientific Publishers, Singapore, 2007.

- [17] S. Fleishman, D. Cohen-Or, and C. T. Silva, *Robust moving least-squares fitting with sharp features*, ACM transactions on graphics (TOG) **24** (2005), no. 3, 544–552.
- [18] N. Flyer, G.A. Barnett, and L.J. Wicker, *Enhancing finite differences with radial basis functions: experiments on the Navier–Stokes equations*, J. Comput. Phys. **316** (2016), 39–62.
- [19] 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.
- [20] B. Fornberg and N. Flyer, *A Primer on Radial Basis Functions with Applications to the Geosciences*, SIAM, Philadelphia, 2015.
- [21] R. Franke, *Scattered data interpolation: tests of some methods*, Mathematics of computation **38** (1982), no. 157, 181–200.
- [22] G. Guennebaud and M. Gross, *Algebraic point set surfaces*, ACM Transactions on Graphics (TOG), vol. 26, ACM, 2007, p. 23.
- [23] R. L. Hardy, *Multi-quadric equations of topography and other irregular surfaces*, Journal of geophysical research **76** (1971), no. 8, 1905–1915.
- [24] E. J. Kansa, *Multi-quadrics – A scattered data approximation scheme with applications to computational fluid-dynamics – I surface approximations and partial derivative estimates*, Comput. Math. Appl. **19** (1990), no. 8-9, 127–145.
- [25] ———, *Multi-quadrics – A scattered data approximation scheme with applications to computational fluid-dynamics – II solutions to parabolic, hyperbolic and elliptic partial differential equations*, Comput. Math. Appl. **19** (1990), no. 8-9, 147–161.
- [26] P. Lancaster, *Moving weighted least-squares methods*, Polynomial and spline approximation, 1979.
- [27] P. Lancaster and K. Salkauskas, *Surfaces generated by moving least squares methods*, Mathematics of Computation **37** (1981), no. 155, 141–158.
- [28] D. Levin, *The approximation power of moving least-squares*, Mathematics of Computation of the American Mathematical Society **67** (1998), no. 224, 1517–1531.
- [29] K.M. Liew, Y.Q. Huang, and J.N. Reddy, *Moving least squares differential quadrature method and its application to the analysis of shear deformable plates*, International Journal for Numerical Methods in Engineering **56** (2003), no. 15, 2331–2351.
- [30] ———, *Vibration analysis of symmetrically laminated plates based on FSDT using the moving least squares differential quadrature method*, Computer Methods in Applied Mechanics and Engineering **192** (2003), no. 19, 2203–2222.
- [31] J. C. Mairhuber, *On Haar’s theorem concerning Chebychev approximation problems having unique solutions*, Proceedings of the American Mathematical Society **7** (1956), no. 4, 609–615.
- [32] D. H. McLain, *Drawing contours from arbitrary data points*, The Computer Journal **17** (1974), no. 4, 318–324.
- [33] J. Meinguet, *An intrinsic approach to multivariate spline interpolation at arbitrary points*, Polynomial and Spline Approximations (1979), 163–190.

- [34] ———, *Multivariate interpolation at arbitrary points made simple*, Zeitschrift für angewandte Mathematik und Physik ZAMP **30** (1979), no. 2, 292–304.
- [35] C. A. Micchelli, *Interpolation of scattered data: distance matrices and conditionally positive definite functions*, Approximation theory and spline functions, Springer, 1984, pp. 143–145.
- [36] M. J. D. Powell, *Approximation theory and methods*, Cambridge university press, 1981.
- [37] S. Schaefer, T. McPhail, and J. Warren, *Image deformation using moving least squares*, ACM transactions on graphics (TOG), vol. 25, ACM, 2006, pp. 533–540.
- [38] D. Shepard, *A two-dimensional interpolation function for irregularly-spaced data*, Proceedings of the 1968 23rd ACM national conference, ACM, 1968, pp. 517–524.
- [39] C. Shu, H. Ding, and K.S. Yeo, *Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier–Stokes equations*, Computer Methods in Applied Mechanics and Engineering **192** (2003), no. 7-8, 941–954.
- [40] S. Tiwari and J. Kuhnert, *Finite pointset method based on the projection method for simulations of the incompressible navier-stokes equations*, Meshfree methods for partial differential equations, Springer, 2003, pp. 373–387.
- [41] A.I. Tolstykh and D.A. Shirobokov, *On using radial basis functions in a "finite difference mode" with applications to elasticity problems*, Computational Mechanics **33** (2003), no. 1, 68–79.
- [42] G. B. Wright and B. Fornberg, *Scattered node compact finite difference-type formulas generated from radial basis functions*, J. Comput. Phys **212** (2006), no. 1, 99–123.