SIAM J. Sci. Comput.
Vol. 43, No. 2, pp. A1441–A1471

A LEAST SQUARES RADIAL BASIS FUNCTION FINITE DIFFERENCE METHOD WITH IMPROVED STABILITY PROPERTIES[*]

IGOR TOMINEC[†], ELISABETH LARSSON[†], AND ALFA HERYUDONO[‡]

Abstract. Localized collocation methods based on radial basis functions (RBFs) for elliptic problems appear to be nonrobust in the presence of Neumann boundary conditions. In this paper, we overcome this issue by formulating the RBF-generated finite difference method in a discrete least squares setting instead. This allows us to prove high-order convergence under node refinement and to numerically verify that the least squares formulation is more accurate and robust than the collocation formulation. The implementation effort for the modified algorithm is comparable to that for the collocation method.

Key words. radial basis function, least squares, partial differential equation, elliptic problem, Neumann condition, RBF-FD

AMS subject classifications. 65N06, 65N12, 65N35

DOI. 10.1137/20M1320079

1. Introduction.

Radial basis function–generated finite difference methods (RBF-FD) generalize classical finite difference methods (FD) to scattered node settings. However, while FD uses tensor products of one-dimensional (1D) derivative approximations, RBF-FD directly computes multivariate approximations, which is advantageous when differentiation is not aligned with a coordinate direction [14]. In this paper, we generalize RBF-FD to a least squares setting (RBF-FD-LS), which improves stability and accuracy.

RBF-FD is a meshfree method, which provides flexibility with respect to the geometry. In contrast to FD methods where an entire coordinate dimension is affected by adaptive refinement, RBF-FD allows for coordinate independent local adaptivity [19].

The RBF-FD method was first introduced by Tolstykh in 2000 [30], and other early papers include [28, 35]. The method is based on the idea that given scattered nodes $x_j \in \mathbb{R}^d$, $j = 1, \dots, n$, in the neighborhood of a point x, we can create a localized RBF approximation of the function $u(x)$ using these "stencil points,"

$$u_h(x) = \sum_{j=1}^{n} c_j \phi(\|x - x_j\|) \equiv \sum_{j=1}^{n} c_j \phi_j(x), \tag{1.1}$$

where h is a measure of the internodal distance, $\phi(r)$ is an RBF, and c_j are unknown coefficients. The interpolation conditions $u_h(x_i) = u(x_i)$ lead to the linear system

[*] Submitted to the journal's Methods and Algorithms for Scientific Computing section March 6, 2020; accepted for publication (in revised form) February 8, 2021; published electronically April 26, 2021.
https://doi.org/10.1137/20M1320079
Funding: The work of the first and second authors was supported by Swedish Research Council grant 2016-04849. The third author was supported by National Science Foundation grant NSF-DMS 2012011 and by the University of Massachusetts Dartmouth.

[†] Department of Information Technology, Uppsala University, SE-751 05 Uppsala, Sweden (igor.tominec@it.uu.se, Elisabeth.Larsson@it.uu.se).
[‡] Department of Mathematics, University of Massachusetts Dartmouth, Dartmouth, MA 02747 USA (aheryudono@umassd.edu).

$$\underbrace{\begin{pmatrix} \phi_1(x_1) & \cdots & \phi_n(x_1) \\ \vdots & & \vdots \\ \phi_1(x_n) & \cdots & \phi_n(x_n) \end{pmatrix}}_{A} \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} u(x_1) \\ \vdots \\ u(x_n) \end{pmatrix}. \tag{1.2}$$

If we let $\underline{c} = (c_1, \dots, c_n)^T$ and $\underline{u} = (u(x_1), \dots, u(x_n))^T$, we have that $\underline{c} = A^{-1}\underline{u}$. A benefit of using RBFs is that for commonly used radial functions $\phi(r)$ the matrix A is guaranteed to be nonsingular for distinct node points [25, 18]. We can then proceed to apply an operator to the approximation:

$$\begin{aligned} \mathcal{L}u_h(x) &= \sum_{j=1}^{n} c_j \mathcal{L}\phi_j(x) \\ &= \underbrace{(\mathcal{L}\phi_1(x), \dots, \mathcal{L}\phi_n(x))}_{a^{\mathcal{L}}}\underline{c} = a^{\mathcal{L}} A^{-1}\underline{u} \equiv (\mathcal{L}\psi_1(x), \dots, \mathcal{L}\psi_n(x))\underline{u} \equiv w^{\mathcal{L}}\underline{u}, \end{aligned} \tag{1.3}$$

where $\{\psi_j(x)\}_{j=1}^{n}$ forms a cardinal basis for the local interpolant, i.e., $\psi_j(x_i) = \delta_{ij}$, and $w^{\mathcal{L}}$ are the stencil weights used for approximating the operator at the point x.

In the early work on RBF-FD, infinitely smooth RBFs as the Gaussian RBF with $\phi(r) = \exp(-r^2)$ or the multiquadric RBF with $\phi(r) = \sqrt{1 + r^2}$ were used. Lately, there has been an increasing interest in using piecewise smooth polyharmonic splines (PHS) with $\phi(r) = |r|^{2k-1}$, $k \geq 1$. These are conditionally positive definite functions. It was shown in [18] that by adding a polynomial basis of a degree corresponding to the order of conditional positive definiteness and constraining the RBF coefficients $\underline{c}$ to be orthogonal to this basis, we can guarantee strict positive definiteness of the quadratic form $\underline{c}^T A \underline{c}$, which is important when proving optimality results. The RBF approximation then takes the form

$$u_h(x) = \sum_{j=1}^{n} c_j \phi_j(x) + \sum_{j=1}^{m} \mu_j p_j(x), \quad \sum_{j=1}^{n} c_j p_k(x_j) = 0, \tag{1.4}$$

where the second equation is the constraint. The dimension m of the polynomial space is given by the degree p of the polynomial as $m = \binom{p+d}{d}$, where d is the number of spatial dimensions. In the Ph.D. thesis [2], and the subsequent papers [11, 12, 5, 4], it was shown that it is beneficial to append a polynomial of a higher degree p than strictly required. First, the convergence order of the method depends on p [3]. Second, the behavior near boundaries is improved compared with classical polynomial-based FD [4]. It was suggested in [12] that for a 2D problem, using a stencil size $n = 2m$ leads to a robust method. We use this strategy in this paper.

The interpolation relation corresponding to (1.2) for the polynomially augmented case becomes

$$\underbrace{\begin{pmatrix} A & P \\ P^T & 0 \end{pmatrix}}_{\tilde{A}} \begin{pmatrix} \underline{c} \\ \underline{\mu} \end{pmatrix} = \begin{pmatrix} \underline{u} \\ 0 \end{pmatrix}, \tag{1.5}$$

where $P_{ij} = p_j(x_i)$ and $\underline{\mu} = (\mu_1, \dots, \mu_m)^T$. Similarly to (1.3), using (1.5) for the coefficient vectors, we get the differentiation relation

$$\begin{aligned}\mathcal{L}u_h(x) &= \underbrace{\begin{pmatrix}a^{\mathcal{L}} & p^{\mathcal{L}}\end{pmatrix}}_{b^{\mathcal{L}}}\begin{pmatrix}\underline{c}\\ \underline{\mu}\end{pmatrix}\\ &= b^{\mathcal{L}}\tilde{A}^{-1}\begin{pmatrix}\underline{u}\\ 0\end{pmatrix} = (b^{\mathcal{L}}\tilde{A}^{-1})_{1:n}\underline{u} \equiv (\mathcal{L}\psi_1(x)\cdots\mathcal{L}\psi_n(x))\underline{u} \equiv w^{\mathcal{L}}\underline{u},\end{aligned} \tag{1.6}$$

where $p^{\mathcal{L}} = (\mathcal{L}p_1(x), \dots, \mathcal{L}p_m(x))$. The PHS + polynomial RBF-FD method works well, but there is some sensitivity to the node layout, e.g., P can become rank deficient for Cartesian node layouts. Several authors have developed algorithms for high quality scattered node generation [13, 27, 29, 32]. Another issue that we have encountered, and that was also noted in [17], is that errors become large at boundaries with Neumann boundary conditions.

In this paper, we propose to improve the performance of the PHS + polynomial RBF-FD method by introducing least squares approximation (oversampling) at the PDE level. The least squares approach is also applicable to RBF-FD with other types of basis functions. A related study is [15], where least squares approximation is introduced in an RBF partition of unity method (RBF-PUM). It was shown that least squares RBF-PUM is numerically stable under patch refinement, which is not the case for collocation RBF-PUM. In [24] it is shown under quite general conditions that given enough oversampling, a broad class of discretizations is uniformly stable.

A recent paper [10] analyzes a least squares RBF-FD method formulated over a closed manifold. The formulation of the method is different from ours in that node points and evaluation points are the same; the oversampling is determined by the stencil size, and the theoretical analysis is performed using other strategies. Another recent paper is [20], where RBF-FD and RBF-PUM are combined to construct a method that is related but uses a different approximation strategy. Least squares approximation has been used together with RBF-FD by other authors to address some specific problems. In [17], an overdetermined linear system is formed by enforcing both the PDE and the Dirichlet boundary conditions on the boundary, to improve the stability of the method. In [22] the context is the closest point method applied to a problem with a moving boundary in combination with RBF-FD. Enforcement of both the PDE and the constant-along-a-normal property of the closest point solution leads to an overdetermined system and a robust method.

The main contributions of this paper are

- the RBF-FD-LS algorithm that performs better than collocation-based RBF-FD in terms of efficiency and stability for the tested PDE problems,
- error estimates that have been validated numerically for RBF-FD-LS approximations when using the PHS + polynomial basis,
- a better understanding of the properties of RBF-FD approximations in terms of a piecewise continuous trial space.

The outline of this paper is as follows. In section 2, we define a Poisson problem with Dirichlet and Neumann boundary conditions. Then in section 3, we derive the RBF-FD-LS method. Section 4 focuses on the properties of the RBF-FD trial space, and then convergence and error estimates are derived in section 5. Numerical experiments that validate the theoretical results are shown in section 6. The paper ends with final remarks on the method and results in section 7.

2. The model problem.

We build our understanding on a model problem, the Poisson equation with Dirichlet and Neumann boundary data:

$$
\begin{array}{rcrcll}
\mathcal{L}_2 u(y) & \equiv & \Delta u(y) & = & f_2(y), & y \in \Omega, \\
\mathcal{L}_0 u(y) & \equiv & u(y) & = & f_0(y), & y \in \partial\Omega_0, \\
\mathcal{L}_1 u(y) & \equiv & \nabla u(y) \cdot n & = & f_1(y), & y \in \partial\Omega_1.
\end{array} \tag{2.1}
$$

We also use the notation Ω_i for the domain associated with $\mathcal{L}_i$. When working with the PDE problem, it is practical to have a unified formulation. We reformulate the system above as

$$D(y)u(y) = F(y), \tag{2.2}$$

where the specific operator $D(y) = \mathcal{L}_i$ and right-hand-side function $F(y) = f_i(y)$ depend on the location of y.

The regularity of the problem depends on the geometry of the domain Ω in combination with the given right-hand-side functions. In the problems that we solve in this paper, the domain is either smooth or convex, and the data is chosen such that the solution has bounded and continuous second derivatives. This ensures that the PDE problem (2.1) is well-defined pointwise. In order to achieve high-order convergence, we require the solution to have additional smoothness. We define the L_2-norm over a domain Ω as $\|u\|^2_{L_2(\Omega)} = \int_\Omega u(y)^2\, dy$ and use the notation $\|u\|_{L_2(\Omega)} = \|u\|_\Omega$ for brevity. We require $u \in W^{p+1}_\infty(\Omega) \subset W^{p+1}_2(\Omega) = \{u \,|\, \|D^\alpha u\|_{L_2(\Omega)} < \infty,\ |\alpha| \leq p+1\}$, where $p \geq 2$ is the degree of the polynomial basis added to the PHS approximation (1.4) that we use in the numerical method.

Since we solve the discretized problem in the least squares sense, it is convenient for the theoretical results derived in section 5 to state also the continuous problem in least squares form. We require $\tilde{u} \in V \subset W^2_2(\Omega)$ for the least squares solution, where the subspace V is determined by the selected representation of the solution. Since we only use the continuous problem at a conceptual level, we are not specifying the subspace further. The squared L_2-norm of the residual of the PDE problem for a function $v \in V$ is given by

$$
\begin{aligned}
\|r(v)\|^2_{L_2(\Omega)} &= \int_{\partial\Omega_0} (\mathcal{L}_0 v(y) - f_0(y))^2 + \int_{\partial\Omega_1} (L_1 v(y) - f_1(y))^2 + \int_\Omega (\mathcal{L}_2 v(y) - f_2(y))^2 \\
&= \int_{\partial\Omega_0} \left(\mathcal{L}_0(v-u)\right)^2 dy + \int_{\partial\Omega_1} \left(\mathcal{L}_1(v-u)\right)^2 dy + \int_\Omega \left(\mathcal{L}_2(v-u)\right)^2 dy,
\end{aligned} \tag{2.3}
$$

where $f_i = L_i u$ was used in the second equality. If we introduce the bilinear form

$$a(u,v) = \int_{\partial\Omega_0} uv\, dy + \int_{\partial\Omega_1} \frac{\partial u}{\partial n}\frac{\partial v}{\partial n}\, dy + \int_\Omega \Delta u \Delta v\, dy, \tag{2.4}$$

and note that $\|r(v)\|^2_{L_2(\Omega)} = a(v-u, v-u)$, the least squares solution of (2.1) is

$$\tilde{u} = \arg\min_{v \in V} a(v-u, v-u). \tag{2.5}$$

Alternatively, using that the residual is a-orthogonal to V, we can write

$$a(\tilde{u} - u, v) = 0 \quad \forall v \in V. \tag{2.6}$$

When $u \in V$, the least squares problem solves the PDE problem exactly, but in general for a numerical approximation, u and $\tilde{u}$ reside in different subspaces, leading to a nonzero residual.

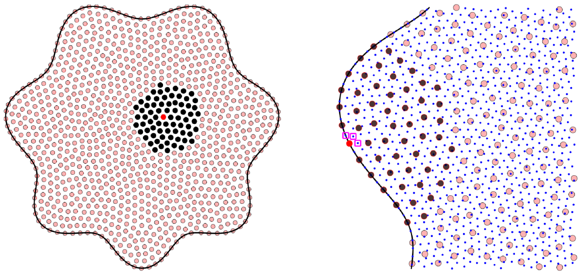

FIG. 1. *The black curve represents the domain boundary $\partial\Omega$. The pale-red dots distributed over Ω are the points in the node set X. The black dots are in each case the node points belonging to the stencil centered at the red point. The evaluation points in the node set Y are shown in the right subfigure as blue dots. Square markers indicate which evaluation points select this particular stencil for evaluating u_h.*

3. Formulation of RBF-FD-LS in practice.

We start with generating a node set $X = \{x_k\}_{k=1}^N$ that covers the domain Ω, on which we solve the PDE problem (2.2). It is beneficial for the approximation quality if the node distance is nearly uniform or varies smoothly over the domain. We associate each x_k with a stencil and denote the n points (including x_k) in the local neighborhood of x_k that contribute to the stencil by X_k. An example of a global node set and a stencil is given in the left part of Figure 1.

To evaluate the RBF-FD approximation at a point $y \in \bar{\Omega}$, we need a stencil selection method. In our algorithm we choose the stencil associated with the point x_k that is closest to y. That is,

$$k(y) = \arg\min_i \|y - x_i\|. \tag{3.1}$$

A practical issue is that there are always points that are equally close to two or more stencils. Therefore we also need to break the tie, such that each y is uniquely associated with one stencil. We then use (1.6) to write down the global RBF-FD approximation to the solution of the PDE problem evaluated at the point y as

$$u_h(y) = \left(b_k^{\mathcal{L}_0} \tilde{A}_k^{-1}\right)_{1:n} u_h(X_k) = \left(\psi_1^k(y), \ldots, \psi_n^k(y)\right) u_h(X_k) = w_k^{\mathcal{L}_0} u_h(X_k), \tag{3.2}$$

where the subscript or superscript $k = k(y)$ indicates quantities computed in the stencil associated with x_k and where $u_h(X_k)$ is a column vector with u_h evaluated at the local node set. The expression for the action of a differential operator $D(y)$ on the global RBF approximation u_h, evaluated at the point y, follows from (3.2):

$$D(y)u_h(y) = \left(D(y)\psi_1^k(y), \ldots, D(y)\psi_n^k(y)\right) u_h(X_k) = w_k^{D(y)} u_h(X_k). \tag{3.3}$$

We note that the local matrices $\tilde{A}_k$ can be reused for all points y that select the same stencil, and for all operators.

To solve the PDE problem (2.2), we sample the approximation (3.3) and the PDE data $F(y)$ at a global node set $Y = \{y_i\}_{i=1}^M$ that has to contain nodes in Ω, at $\partial\Omega_0$, and at $\partial\Omega_1$. An example of an evaluation node set Y is shown in the right part of Figure 1. We construct a sparse global linear system

$$D_h(Y, X)u_h(X) = F(Y), \tag{3.4}$$

where row i contains the equation for $D(y_i)u_h(y_i) = F(y_i)$ and the corresponding weights from (3.3) are entered into the columns corresponding to the global indices of the nodes in X_k. In the same way, we form the relation

$$u_h(Y) = E_h(Y, X)u_h(X), \tag{3.5}$$

using weights from (3.2). If the number of evaluation points $M > N$, both $D_h(Y, X)$ and $E_h(Y, X)$ are rectangular $M \times N$ matrices. In [31] we provide MATLAB code to generate rectangular RBF-FD matrices such as $E_h(Y, X)$ or $D_h(Y, X)$.

In the discretized PDE problem, $u_h(X)$ is the vector of unknowns, and we formally write the least squares solution of the linear system as

$$u_h(X) = D_h^+(Y, X)\, F(Y), \tag{3.6}$$

where the $N \times M$ matrix $D_h^+(Y, X) = (D_h^T D_h)^{-1} D_h^T$ is the pseudoinverse of $D_h(Y, X)$. To evaluate the solution at Y we add the step (3.5) to get

$$u_h(Y) = E_h(Y, X)\, D_h^+(Y, X)\, F(Y). \tag{3.7}$$

The overdetermined linear system (3.4) can also be formulated as a discrete residual minimization problem. We define the residual

$$r(Y) = D_h(Y, X)u_h(X) - F(Y). \tag{3.8}$$

Then the solution (3.6) minimizes $\|r(Y)\|_2^2$, and it also holds that

$$D_h^T(Y, X)r(Y) = 0, \tag{3.9}$$

due to the orthogonality property of the least squares residual.

The collocation RBF-FD method, where $Y = X$, is a special case of the derivation above, where the stencil selected for $y_k = x_k$ is always k, $D_h^+(X, X) = D_h^{-1}(X, X)$, and $E_h(X, X) = I_h$. This leads to

$$r(X) = D_h(X, X)u_h(X) - F(X) = D_h(X, X)(D_h^{-1}(X, X)F(X)) - F(X) = 0. \tag{3.10}$$

In the discrete minimization of the residual (3.8), each equation has the same weight. This may cause problems with convergence to the PDE solution under node refinement. We start by introducing a weighted discrete ℓ_2-norm that corresponds to the continuous L_2-norm with the integral replaced by a discrete quadrature formula. The error in this approximation is further discussed in subsection 4.4. We leave place holders β_i for additional balancing of the different parts of the residual and discuss these further in subsection 5.4. Let a domain Ω be discretized by M points y_i, $i = 1, \ldots, M$. Then

$$(u, v)_{\ell_2(\Omega)} = \frac{|\Omega|}{M} \sum_{i=1}^{M} u(y_i)v(y_i), \quad \|u\|_{\ell_2(\Omega)}^2 = (u, u)_{\ell_2(\Omega)}, \tag{3.11}$$

where $|\Omega| = \int_\Omega 1\, dy$. We denote the number of evaluation points that discretize the operator $\mathcal{L}_i$ in (2.1) by M_i and note that if the evaluation points are quasi uniform with node distance h_y, then for $d > 1$

$$h_y = c_0 \left(\frac{|\partial\Omega_0|}{M_0} \right)^{\frac{1}{d-1}} = c_1 \left(\frac{|\partial\Omega_1|}{M_1} \right)^{\frac{1}{d-1}} = c_2 \left(\frac{|\Omega|}{M_2} \right)^{\frac{1}{d}}, \tag{3.12}$$

where $c_0 \approx c_1 \approx c_2 \approx 1$. Scaling the evaluation matrix as $\bar{E}_h = (\frac{|\Omega|}{M})^{\frac{1}{2}} E_h$ leads to

$$\|u_h\|^2_{\ell_2(\Omega)} = \frac{|\Omega|}{M}\|u_h(Y)\|_2^2 = \|\bar{E}_h u_h(X)\|_2^2 = u_h(X)^T \bar{E}_h^T \bar{E}_h u_h(X). \tag{3.13}$$

For D_h, we scale according to the location of y such that $\bar{D}_h = \text{diag}(\beta(Y))D_h$, where

$$\beta(y) = \begin{cases} \left(\frac{|\partial\Omega_0|}{M_0}\right)^{\frac{1}{2}} \beta_0 & \approx & h_y^{\frac{d-1}{2}}\beta_0, & y \in \partial\Omega_0, \\ \left(\frac{|\partial\Omega_1|}{M_1}\right)^{\frac{1}{2}} \beta_1 & \approx & h_y^{\frac{d-1}{2}}\beta_1, & y \in \partial\Omega_1, \\ \left(\frac{|\Omega|}{M_2}\right)^{\frac{1}{2}} \beta_2 & \approx & h_y^{\frac{d}{2}}\beta_2, & y \in \Omega, \end{cases} \tag{3.14}$$

and, similarly, we let $\bar{F}(y) = \beta(y)F(y)$. For the scaled residual $\bar{r}(Y)$, noting that $D_h(Y,X)u_h(X) = D(Y)u_h(Y)$ and $F(Y) = D(Y)u(Y)$, we get

$$\begin{aligned} \|\bar{r}(Y)\|_2^2 &= \|\bar{D}_h(Y,X)u_h(X) - \bar{F}(Y)\|_2^2 = \|\beta(Y)D(Y)(u_h(Y) - u(Y))\|_2^2 \\ &= \beta_0^2\|\mathcal{L}_0(u_h - u)\|^2_{\ell_2(\partial\Omega_0)} + \beta_1^2\|\mathcal{L}_1(u_h - u)\|^2_{\ell_2(\partial\Omega_1)} + \beta_2^2\|\mathcal{L}_2(u_h - u)\|^2_{\ell_2(\Omega)}. \end{aligned} \tag{3.15}$$

Comparing with the residual of the continuous problem (2.3) and the continuous bilinear form (2.4), we introduce the discrete bilinear form

$$a_h(u,v) = \beta_0^2\,(u,v)_{\ell_2(\partial\Omega_0)} + \beta_1^2\left(\frac{\partial u}{\partial n}, \frac{\partial v}{\partial n}\right)_{\ell_2(\partial\Omega_1)} + \beta_2^2\,(\Delta u, \Delta v)_{\ell_2(\Omega)}. \tag{3.16}$$

So far, we have assumed that the Dirichlet boundary conditions are enforced in the least squares sense. It has been noted, e.g., in [23], that when Dirichlet conditions are imposed strongly, the overall accuracy is improved. Assuming that there are node points $X_{\partial\Omega_0} \subset X$ that discretize the Dirichlet boundary, we let $\tilde{X} = X \setminus X_{\partial\Omega_0}$ and $u_h(X_{\partial\Omega_0}) = 0$. Then we rewrite the discretized least squares PDE problem (3.4) as

$$\bar{D}_h(Y,\tilde{X})u_h(\tilde{X}) = \bar{F}(Y) - \bar{D}_h(Y, X_{\partial\Omega_0})u_h^0(X_{\partial\Omega_0}) \equiv \tilde{F}(Y), \tag{3.17}$$

where $u_h^0(X_{\partial\Omega_0}) = f_0(X_{\partial\Omega_0})$ is a subset of the Dirichlet boundary data. Note that u_h is in general nonzero at the Dirichlet boundary between the data points. We denote the trial space containing all functions of the form (3.2) by V_h and we denote the subspace with zero Dirichlet data by V_h^0. The solution to the original problem is given by $u_h + u_h^0 \in V_h$, where $u_h \in V_h^0$. Similarly to (2.5), we write the least squares problem on the form

$$u_h = \underset{v_h \in V_h^0}{\arg\min}\, a_h\left(v_h + u_h^0 - u, v_h + u_h^0 - u\right), \tag{3.18}$$

where V_h is the RBF-FD trial space. We have the orthogonality property

$$a_h\left(u_h + u_h^0 - u, v_h\right) = 0 \quad \forall v_h \in V_h^0. \tag{3.19}$$

To see how this relates to the matrix-based description of the discrete least squares problem, we introduce a (nonorthogonal) basis for V_h. For each evaluation point y there is a unique representation of u_h in terms of the local cardinal functions (3.2). We define global cardinal functions as

$$\Psi_j(y) = \begin{cases} \psi_i^k(y), & x_j \in X_k, \\ 0, & x_j \notin X_k, \end{cases} \tag{3.20}$$

where $k = k(y)$ is the stencil selected for the evaluation point y, and $i(j)$ is the local index i in X_k of $x_j \in X$. We represent a nonhomogeneous function $u_h \in V_h$ as

$$u_h(y) = \sum_{j=1}^{N} u_h(x_j)\Psi_j(x). \tag{3.21}$$

We note that $D_h(y_i, x_j) = D(y_i)\Psi_j(y_i)$ and $E_h(y_i, x_j) = \Psi_j(y_i)$. If we insert (3.21) in (3.19) and let $v_h = \Psi_i$, we get

$$\sum_{j=1}^{N} a_h(\Psi_j, \Psi_i)u_h(x_j) = a_h(u, \Psi_i), \tag{3.22}$$

where $a_h(\Psi_j, \Psi_i)$ is an element of the matrix $\bar{D}_h(Y, X)^T \bar{D}_h(Y, X)$, and $a_h(u, \Psi_i)$ is an element of the right-hand-side vector $\bar{D}_h(Y, X)^T \bar{F}(Y)$ in the weighted normal equations. The specific properties of the trial space and the cardinal basis functions are further discussed in the following section.

4. The discontinuous trial space.

The trial space V_h is a piecewise space. The stencil selection algorithm that we use for the evaluation points results in the domain being divided into Voronoi regions $\mathcal{V}_k$ around each stencil center point $x_k \in X$. For an illustration of the Voronoi regions in two dimensions, see Figure 3. Locally we have $u_h \in W^2_\infty(\mathcal{V}_k) \subset W^2_2(\mathcal{V}_k)$ due to the smoothness of the at least cubic PHS basis.

Theorem 4.1. *Assume that for each stencil underlying the trial space approximation, the node set X_k is unisolvent with respect to polynomials of degree p. Then $u_h|_{\mathcal{V}_k} = 0$ if and only if $u_h(X_k) = 0$, and $u_h|_\Omega = 0$ if and only if $u_h(X) = 0$.*

Proof. The results follow from the uniqueness of the local interpolation problems. □

A scattered node set is quantified by its fill distance h, measuring the radius of the largest ball empty of nodes in Ω, and its separation distance q, defined by

$$h = \sup_{x\in\Omega} \min_{x_j \in X} \|x - x_j\|_2 \geq q = \frac{1}{2} \min_{\substack{j\neq k \\ x_j, x_k \in X}} \|x_j - x_k\|_2. \tag{4.1}$$

The quality of a node set is related to $c_q = q/h < 1$. The trial space approximation improves with increasing node quality. In the following subsections, we derive the results that we need for the error estimates in section 5, in terms of the fill distance h of X, the fill distance h_y of Y, and the node quality c_q.

4.1. Interpolation errors.

We define the interpolant $I_h(u) \in V_h$ of a function u as $I_h(u) = \sum_{j=1}^{N} u(x_j)\Psi_j$. For a function $u \in W^{p+1}_\infty(\Omega)$ that allows Taylor series expansion around $x_k \in \mathcal{V}_k$, we can assess the local interpolation error $e_I = I_h(u) - u$ and its derivatives using a result from [3]. When h is small enough, we have that

$$|\mathcal{L}_i\left(I_h(u(y)) - u(y)\right)| \leq \alpha_{k,i} h^{p+1-i} |u|_{W^{p+1}_\infty(\mathcal{V}_k)}, \quad y \in \mathcal{V}_k, \tag{4.2}$$

where $\mathcal{L}_i$ is a differential operator of order i, $|u|_{W^q_\infty(\Omega)} = \sum_{|\alpha|=q} \|D^\alpha u\|_{L_\infty(\Omega)}$, and $\alpha_{k,i}$ are constants that depend on the degree p of the polynomial basis, and on the node quality c_q of the stencil node set X_k. If the node layout is nonuniform, indicated by a small value of c_q, the interpolation problem has a large Lebesgue constant [26], and consequently a larger interpolation error. The error is also larger for skewed stencils that are evaluated close to their support boundary.

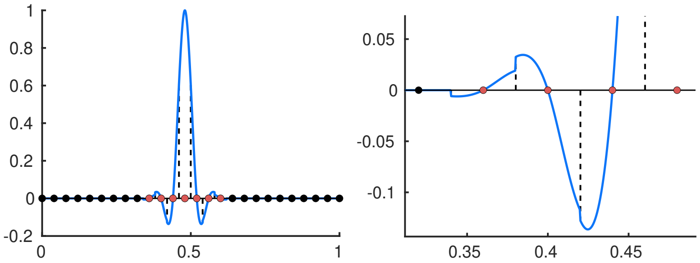

Fig. 2. *Left: A cardinal function generated with RBF-FD on a uniform node set in one dimension with stencil size $n = 7$. Right: A close-up to illustrate the discontinuities at the intersections of the Voronoi regions.*

When we use the interpolation error in the global error estimate, we take a norm over the domain. If we let $\alpha_i = |\Omega| \max_k \alpha_{k,i}$, we have

$$\|\mathcal{L}_i(I_h(u) - u)\|_{\ell_2(\Omega)} \leq \alpha_i h^{p+1-i} |u|_{W_\infty^{p+1}(\Omega)}. \tag{4.3}$$

At the edge of a Voronoi region $I_h(u)$ takes slightly different values from each side. That is, if u is not represented exactly in V_h, the interpolant $I_h(u)$ has a discontinuity proportional to h^{p+1}, that goes to zero as the space is refined, along the edges of the Voronoi regions. This means that the cardinal basis functions also have discontinuities between Voronoi regions; see Figure 2.

4.2. Derivatives and norms in the trial space. We want to reuse the results that we derive here also for the smoothed trial space defined in subsection 4.3. Therefore, we define the local stencil-based functions $v_h^k(y)$ that together form v_h (cf. (3.2)) over the domains $\mathcal{V}_k^\delta \supset \mathcal{V}_k$, which represent an extension of the Voronoi regions $\mathcal{V}_k$ with at most distance $\delta h/2$ in any direction, where $\delta < 1$. We have

$$v_h^k(y) = \sum_{i=1}^{n} \psi_i^k(y) v_h(x_i^k), \quad y \in \mathcal{V}_k^\delta, \tag{4.4}$$

where x_i^k is the ith element of X_k. When discussing bounds, we also use the following explicit form derived in [3]:

$$v_h^k(y) = v_h(X_k)^T (I - WP) A^{-1} \phi(y) + v_h(X_k)^T W p(y), \tag{4.5}$$

where $W = A^{-1}P(P^T A^{-1} P)^{-1}$, the PHS vector $\phi(y) = (\phi_1^k(y), \ldots, \phi_n^k(y))^T$, and the polynomial vector $p(y) = (p_1(y), \ldots, p_m(y))^T$.

Theorem 4.2. *Assume that we use cubic splines, that Theorem 4.1 holds, and that the node quality c_q of X has a lower bound $c_{\min}$, within each stencil node set, for a sequence of discretizations with different fill distances h. We define $\tilde{y} = y/h$ and $\tilde{v}_h^k = \sum_{i=1}^n \tilde{\psi}_i^k(\tilde{y}) v_h(x_i^k)$, where $\tilde{\psi}_i^k(\tilde{y}) = \psi_i^k(y)$, and we define a scaled Voronoi region $\tilde{\mathcal{V}}_k$ such that $\tilde{y} \in \tilde{\mathcal{V}}_k$ when $y \in \mathcal{V}_k$. Let $\mathcal{V}_{k,i} = \mathcal{V}_k \cap \Omega_i \subset \mathbb{R}^{d_i}$ and let $\tilde{\mathcal{V}}_{k,i}$ be the scaled counterpart. Then*

$$\begin{aligned}(D_y^s v_h^k, D_y^t v_h^k)_{L_2(\mathcal{V}_{k,i})} &= h^{d_i - |s| - |t|} \left(D_{\tilde{y}}^s \tilde{v}_h^k, D_{\tilde{y}}^t \tilde{v}_h^k \right)_{L_2(\tilde{\mathcal{V}}_{k,i})} \\ &= h^{d_i - |s| - |t|} v_h(X_k)^T \tilde{\Psi}_{k,i}^{s,t} v_h(X_k), \quad |s|, |t| \leq 3,\end{aligned} \tag{4.6}$$

where $\tilde{\Psi}^{s,t}_{k,i}(q,r) = (D^s_{\tilde{y}}\tilde{\psi}^k_q, D^t_{\tilde{y}}\tilde{\psi}^k_r)_{L_2(\tilde{\mathcal{V}}_{k,i})}$. *Furthermore for* $|s|, |t|, |q| \leq 3$,

$$\max_{\substack{\|v_h(X_k)\|_2 > 0 \\ \|D^q v^k_h\|_{L_2(\mathcal{V}_k)>0}}} \frac{(D^s v^k_h, D^t v^k_h)_{L_2(\mathcal{V}_{k,i})}}{\|D^q v^k_h\|^2_{L_2(\mathcal{V}_k)}} \leq h^{d_i - d + 2q - |s| - |t|} K^{s,t,i}_q, \quad k = 1, \ldots, N, \tag{4.7}$$

where $K^{s,t,i}_q$ *is an upper bound for the largest eigenvalue* $K_{\max}$ *of the generalized eigenproblem* $\tilde{\Psi}^{s,t}_{k,i} v = K \tilde{\Psi}^{q,q}_{k,0} v$ *in the subspace of eigenvectors* $v \in \mathbb{R}^n$ *that are in the range of* $\tilde{\Psi}^{q,q}_{k,0}$. *Apart from* s, t, *and* q, *the bound* $K^{s,t,i}_q$ *only depends on the lower bound on the node quality* $c_{\min}$, *the dimension* d, *the stencil size* n, *the polynomial degree* p, *and the dimension of the polynomial space* m.

Proof. For the scaling of the derivatives, we have $D^s_y \psi^k_i(y) = D^s_y \tilde{\psi}^k_i(y/h) = h^{-s} D^s_{\tilde{y}} \tilde{\psi}^k_i(\tilde{y})$. The integration relation is $\int_{\mathcal{V}_k} v(y)dy = \int_{\tilde{\mathcal{V}}_k} v(\tilde{y}) h^{-d}\, d\tilde{y}$. For integration along a boundary intersecting $\mathcal{V}_k$, we instead get h^{d-1}. To understand which factors influence the bounds, we introduce the matrices A_1 and P_1, which correspond to a scaling of the nodes such that $h = 1$, and note that $A = h^3 A_1$ and $P = P_1 H$, where H is a diagonal matrix with elements $h_{jj} = h^{|t_j|}$, where $p_j(x) = x^{t_j}$. Combining (4.5) and the scaled matrices leads to

$$h^{|s|} D^s v^k_h(y) = v_h(X_k)^T (I - W_1 P_1^T) A_1^{-1} D^s_{\tilde{y}} \phi(\tilde{y}) + v_h(X_k)^T W_1 D^s_{\tilde{y}} p(\tilde{y}). \tag{4.8}$$

We note the right-hand side of (4.8) is scale invariant. The smallest eigenvalue of the symmetric matrix A_1 can be bounded in terms of the separation distance q of the stencil node set [34, Corollary 12.7]. When the nodes are scaled such that $h = 1$, we have $q = c_q$. That is, by choosing $q = c_{\min}$ for the bound, it holds for all discretizations in the sequence. All other matrices and vectors can be bounded in terms of the stencil size n, the dimension of the polynomial basis m, the polynomial degree p, and the derivative s. The bounds are larger for skewed stencils and grow with p. An upper bound for the numerator of (4.7) can be obtained by directly bounding the matrices and vectors in (4.8). The denominator is a mass matrix/stiffness matrix for the local scaled cardinal functions. The lower bound is attained for the (nonzero differentiated) cardinal function that has the smallest amount of mass in $\mathcal{V}_k$. The mass is smaller for skewed stencils as well as small Voronoi regions. The same parameters, $c_{\min}$, n, m, p, determine the behavior. The local trial space functions are twice continuously differentiable, while the third derivatives exist at all points, but are piecewise continuous. ∎

To investigate the relations between the discrete and continuous norms (cf. subsection 4.4), and the smoothed and discontinuous trial spaces (cf. subsection 4.3), we introduce a semidiscrete bilinear form $a^*(v,v)$ using the norm

$$\|v\|^2_{L^*_2(\Omega_i)} = (v,v)_{L^*_2(\Omega_i)} = \sum_{k=1}^{N} \int_{\mathcal{V}_k \cap \Omega_i} v(y)^2\, dy. \tag{4.9}$$

For a function $v \in W^2_2$ $\|\mathcal{L}_i v\|_{L^*_2(\Omega)} = \|\mathcal{L}_i v\|_{L_2(\Omega)}$, $i \leq 2$, while for the trial space functions, the semidiscrete norm eliminates the derivatives of the jumps.

4.3. The smoothed trial space. We introduce a smoothing operator $S^0_2 : V^0_h \mapsto V^0$, where $V^0 = \{v \in W^2_\infty(\Omega) \,|\, v(y)|_{\partial\Omega_0} = 0\}$, and let $v = S^0_2(v_h)$. We never construct the smoothed trial space function v in practice, but we use it as a tool in the error analysis (cf. section 5). First we define a set of overlapping patches by extending each Voronoi region $\mathcal{V}_k$ a distance $\delta h/2$, $\delta < 1$, in the normal direction at each interior edge/face. To handle the Dirichlet boundary, we imagine a mirrored set of Voronoi regions outside the boundary and let these extend δh into the domain. The Voronoi regions along the inside of the Dirichlet boundary are assumed to conform to the domain boundary where it intersects the region. As in the previous subsection, we denote the extended Voronoi regions by $\mathcal{V}_k^\delta$.

Then we construct a set of nonnegative, compactly supported, partition of unity weight functions $\{\omega_k\}_{k=1}^{N+N_0}$, where the extra N_0 weight functions belong to the regions outside the Dirichlet boundary. We let $\omega_k \in W_\infty^2(\Omega)$ be supported on $\mathcal{V}_k^\delta$. That is, the weight functions overlap with a distance δh at all shared edges. Let $\mathcal{V}_k^{-\delta}$ denote the interior part of the Voronoi region, and let $\Gamma_k^\delta = \mathcal{V}_k \setminus \mathcal{V}_k^{-\delta}$. For the weight functions it holds that $\sum_{k=1}^{N+N_0} \omega_k = 1$, $y \in \Omega$, $\omega_k(y) = 1$, $y \in \mathcal{V}_k^{-\delta}$, $D^s\omega_k(y) = 0$, $y \in \mathcal{V}_k^{-\delta}$, and

$$|D^s\omega_k| \le \frac{G_{|s|}}{h^{|s|}}, \quad |s| \le 3, \quad y \in \mathcal{V}_k^\delta \setminus \mathcal{V}_k^{-\delta}, \tag{4.10}$$

where $G_0 = 1$. The second derivatives of the weight functions are continuous, while the third derivatives exist but are only piecewise continuous. For further details on the construction of weight functions and their properties, see [33, 15].

We connect the exterior weight functions with zero functions, i.e., $v_h^k \equiv 0$, $k > N$. Then we combine the local interpolants and weight functions to get

$$v = S_2^0(v_h) = \sum_{j=1}^N \omega_j(y) v_h^j(y) = \sum_{j \in J_k}^N \omega_j(y) v_h^j(y), \quad y \in \mathcal{V}_k,$$

where J_k contains the indices of the patches that overlap with $\mathcal{V}_k$.

Theorem 4.3. *Assume that Theorem 4.2 holds, that $\delta < 1$ is fixed for all discretizations, and that $v = S_2^0(v_h)$ for $v_h \in V_h^0$. Then the following relations hold:*

$$\|v_h\|_{L_2(\Omega)}^2 \le (1+\eta_0)\|v\|_{L_2(\Omega)}^2, \tag{4.11}$$

$$a(v,v) \le (1+\eta_a) a^*(v_h, v_h), \tag{4.12}$$

where the bounds η_0 and η_a depend on δ, the node quality $c_{\min}$, the dimension d, the stencil size n, the polynomial degree p, and the dimension of the polynomial space m.

Proof. If $\|v_h\|_{L_2(\mathcal{V}_k)} > 0$, then $\|v\|_{L_2(\mathcal{V}_k)} > 0$ and $\|v\|_{L_2(\mathcal{V}_k^{-\delta})} > 0$, and we use Theorem 4.2 to bound the ratio

$$\frac{\|v_h\|_{L_2(\Gamma_k^\delta)}^2}{\|v\|_{L_2(\mathcal{V}_k^{-\delta})}^2} = \frac{\|v_h\|_{L_2(\Gamma_k^\delta)}^2}{\|v_h\|_{L_2(\mathcal{V}_k^{-\delta})}^2} = \frac{v_h(X_k)^T \tilde{\Psi}_{k,\delta}^{I,I} v_h(X_k)}{v_h(X_k)^T \tilde{\Psi}_{k,-\delta}^{I,I} v_h(X_k)} \le \eta_{k,0}, \tag{4.13}$$

where the subscript $k, \pm\delta$ denotes integration over Γ_k^δ and $\mathcal{V}_k^{-\delta}$. Due to the size ratio of the domains, $\eta_{k,0}$ is approximately proportional to $\delta/(1-\delta)$. The other dependencies come from the basis functions (cf. Theorem 4.2). We sum the local results to get

$$\begin{aligned}\|v_h\|_{L_2(\Omega)}^2 &= \sum_{k=1}^N \|v_h\|_{L_2(\mathcal{V}_k^{-\delta})}^2 + \|v_h\|_{L_2(\Gamma_k^\delta)}^2 \le \sum_{k=1}^N (1+\eta_{k,0})\|v\|_{L_2(\mathcal{V}_k^{-\delta})}^2 \\ &\le \max_k (1+\eta_{k,0})\|v\|_{L_2(\Omega)}^2. \end{aligned} \tag{4.14}$$

Setting $\eta_0 = \max_k(1 + \eta_{k,0})$ gives the result (4.11). For the bilinear form, we start from the restriction to a Voronoi region. We have

$$
\begin{aligned}
a_k(v,v) = a(v,v)|_{\mathcal{V}_k} &= a_k(v,v)|_{\mathcal{V}_k^{-\delta}} + a_k(v,v)|_{\Gamma_k^\delta} \\
&= a_k(v_h^k, v_h^k)|_{\mathcal{V}_k^{-\delta}} + a_k\left(\sum_{j\in J_k} \omega_j v_h^j, \sum_{j\in J_k} \omega_j v_h^j\right)\Bigg|_{\Gamma_k^\delta} \\
&= a_k(v_h^k, v_h^k)|_{\mathcal{V}_k^{-\delta}} + \sum_{i\in J_k}\sum_{j\in J_k} a_k(\omega_i v_h^i, \omega_j v_h^j)|_{\Gamma_k^\delta\cap\mathcal{V}_i^\delta\cap\mathcal{V}_j^\delta} \\
&\leq a_k(v_h^k, v_h^k) + Q\sum_{j\in J_k} a_k(\omega_j v_h^j, \omega_j v_h^j)|_{\Gamma_k^\delta\cap\mathcal{V}_j^\delta},
\end{aligned}
\tag{4.15}
$$

where $Q \approx 2^d$ is the largest number of extended Voronoi regions that overlap at any given point. We note that $\omega_k|_{\mathcal{V}_k^\delta\cap\partial\Omega_0} = 0$, we introduce the notation $\Gamma_{k,j}^\delta = \Gamma_k^\delta \cap \mathcal{V}_j^\delta$ and $\Gamma_{k,j,1}^\delta = \Gamma_k^\delta \cap \mathcal{V}_j^\delta \cap \partial\Omega_1$, and we use (4.10) to estimate one term in the sum in (4.15) as

$$
\begin{aligned}
a_k(\omega_j v_h^j, \omega_j v_h^j)|_{\Gamma_{k,j}^\delta} &= \|\Delta(\omega_j v_h^j)\|^2_{L_2(\Gamma_{k,j}^\delta)} + \|\partial(\omega_j v_h^j)/\partial n\|^2_{L_2(\Gamma_{k,j,1}^\delta)} \\
&\leq 2\left(\|\Delta v_h^j\|^2_{L_2(\Gamma_{k,j}^\delta)} + 2G_1^2h^{-2}\|\nabla v_h^j\|^2_{L_2(\Gamma_{k,j}^\delta)} + G_2^2h^{-4}\|v_h^j\|^2_{L_2(\Gamma_{k,j}^\delta)}\right) \\
&\quad + 2\left(\|\partial v_h^j/\partial n\|^2_{L_2(\Gamma_{k,j,1}^\delta)} + G_1^2h^{-2}\|v_h^j\|^2_{L_2(\Gamma_{k,j,1}^\delta)}\right).
\end{aligned}
\tag{4.16}
$$

Then, we consider the quotient of the overlap terms in $\mathcal{V}_k$ and the bilinear form evaluated over the stencils surrounding and including $\mathcal{V}_k$. We use Theorem 4.2 to write the norms in (4.16) on scale invariant form:

$$
\begin{aligned}
\frac{\sum_{j\in J_k} a_k(\omega_j v_h^j, \omega_j v_h^j)|_{\Gamma_{k,j}^\delta}}{\sum_{j\in J_k} a_j(v_h^j, v_h^j)} &\leq \frac{2\sum_{j\in J_k} v_h(X_j)^T\left(\tilde{\Psi}_{k,j}^{\Delta,\Delta} + 2G_1^2\tilde{\Psi}_{k,j}^{\nabla,\nabla} + G_2^2\tilde{\Psi}_{k,j}^{I,I}\right)v_h(X_j)}{\sum_{j\in J_k} v_h(X_j)^T\left(\tilde{\Psi}_{j}^{\Delta,\Delta} + h\tilde{\Psi}_{j,1}^{\mathcal{L}_1,\mathcal{L}_1} + h^3\tilde{\Psi}_{j,0}^{I,I}\right)v_h(X_j)} \\
&\quad + \frac{2h\sum_{j\in J_k} v_h(X_j)^T\left(\tilde{\Psi}_{j,k,1}^{\mathcal{L}_1,\mathcal{L}_1} + G_1^2\tilde{\Psi}_{j,k,1}^{I,I}\right)v_h(X_j)}{\sum_{j\in J_k} v_h(X_j)^T\left(\tilde{\Psi}_{j}^{\Delta,\Delta} + h\tilde{\Psi}_{j,1}^{\mathcal{L}_1,\mathcal{L}_1} + h^3\tilde{\Psi}_{j,0}^{I,I}\right)v_h(X_j)},
\end{aligned}
\tag{4.17}
$$

where we extended the notation for $\tilde{\Psi}$ to allow integration over $\Gamma_{j,k}$ as in (4.16) and to include composite scalar operators, when the order is the same in each term. The denominator is zero if the bilinear form is zero over all involved stencils. However, with the PHS and polynomial basis functions that we use, this implies that the data is sampled from a polynomial of degree $\leq p$ in the nullspace of the Laplacian operator. Due to the overlap of the stencils and polynomial unisolvency, the data is consistent across stencils. The polynomial is represented exactly on all stencils and we have $a_k(v,v) = a_k(v_h^k, v_h^k)$. If the denominator is positive, similarly as in Theorem 4.2, we can find an upper bound through the generalized eigenproblem on the extended domain involving a Voronoi region and its neighbors. We denote the specific bound for $\mathcal{V}_k$ by $\eta_{k,a}$ and combine (4.15) and (4.17), resulting in

$$
a(v,v) = \sum_{k=1}^N a_k(v,v) \leq \sum_{k=1}^N\left(a_k(v_h^k, v_h^k) + 2Q\eta_{k,a}\sum_{j\in J_k} a_j(v_j^h, v_j^h)\right) \leq (1+\eta_a)a^*(v_h, v_h),
$$

where $\eta_a = 2Q\max_k \eta_{k,a}|J_k|$ provides the result (4.12). □

4.4. Discrete norm errors. The discrete norm $\|\cdot\|_{\ell_2(\Omega)}$ on the set of nodes $Y=\{y_i\}_{i=1}^M$ is an approximation of the continuous norm $\|\cdot\|_{L_2(\Omega)}$, and for the global error estimate, we need to quantify the difference. We start from a generic integral:

$$\mathcal{I}=\int_\Omega f(y)dy=\frac{|\Omega|}{M}\sum_{i=1}^M f(y_i)+\gamma_I(f)=\mathcal{I}_h+\gamma_I(f). \tag{4.18}$$

We want to use this relation for nontrivial domains, which means that we need to consider scattered nodes. Even if the nodes are regular in parts of the domain, they need to be somewhat irregular near the boundary. A very general error estimate for scattered node quadrature is given by

$$|\gamma_I(f)|\le D_M(Y)V(f,\Omega), \tag{4.19}$$

where $D_M(Y)$ is the star discrepancy of the node set and $V(f,\Omega)$ is the Hardy–Krause variation of f [1]. This has been shown for general domains and piecewise smooth functions in [7, 8]. Both of the factors in (4.19) are hard to quantify in general. However, the standard deviation of the error for an arbitrary node layout (Monte Carlo integration) in practical cases decreases as $\mathcal{O}(1/\sqrt{M})$, and for a low discrepancy (quasi random) node layout, it decreases as $\mathcal{O}((\log M)^d/M)$. Furthermore, for a (piecewise) differentiable function, the total variation can be measured as $V(f,\Omega)=\int_\Omega|\nabla f|dy$. Assuming that the evaluation node set Y is quasi uniform and that d is small enough for $(\log M)^d$ to be viewed as almost constant, we can estimate the error in the squared norm of a function $v\in W_2^1$ as

$$\left|\|v\|^2_{L_2(\Omega_i)}-\|v(Y)\|^2_{\ell_2(\Omega_i)}\right|\le C_I h_y^{d_i}\int_{\Omega_i}|\nabla(v^2)|dy, \tag{4.20}$$

where d_i is the dimensionality of Ω_i. To apply this estimate to the trial space function it is easiest to apply it locally to each Voronoi region. The number of points in each region is then small, but we get a statistical averaging through the sum.

THEOREM 4.4. *If Theorem* 4.2 *holds and h_y is chosen according to* (4.26) *for a relative integration error tolerance $\tau<1$, then*

$$\|v_h(Y)\|_{\ell_2(\Omega)}\le(1+\tau)\|v_h\|_{L_2(\Omega)}, \tag{4.21}$$

$$a^*(v_h,v_h)\le\frac{1}{(1-\tau)}a_h(v_h,v_h). \tag{4.22}$$

Proof. We use Theorem 4.2 for the gradients

$$\int_{\mathcal{V}_{k,i}}|\nabla((\mathcal{L}_i v_h^k)^2)|dy=2(|\nabla\mathcal{L}_i v_h^k|,|\mathcal{L}_i v_h^k|)_{\mathcal{V}_{k,i}}=h^{d_i-2i-1}v_h(X_k)^T\tilde{\Psi}_{k,i}^{|\nabla\mathcal{L}_i|,|\mathcal{L}_i|}v_h(X_k) \tag{4.23}$$

and then again for the relative errors when $\|v_h\|_{L_2(\mathcal{V}_k)}>0$ and $a^*_{k,k}(v,v)>0$ to get

$$\frac{\left|\|v_h\|^2_{L_2(\mathcal{V}_k)}-\|v_h(Y)\|^2_{\ell_2(\mathcal{V}_k)}\right|}{\|v_h\|^2_{L_2(\mathcal{V}_k)}}\le 2C_I\frac{h_y^d}{h}K_I^{|\nabla|,|I|}, \tag{4.24}$$

$$\begin{aligned}
(4.25)\quad & \frac{\left|\sum_{i=0}^{2}\left(\|\mathcal{L}_i v_h\|^2_{L_2(\mathcal{V}_{k,i})} - \|\mathcal{L}_i v_h(Y)\|^2_{\ell_2(\mathcal{V}_{k,i})}\right)\right|}{\sum_{i=0}^{2}\|\mathcal{L}_i v_h\|^2_{L_2(\mathcal{V}_{k,i})}} \\
& \le 2C_I \frac{h_y^d}{h} \frac{v_h(X_k)^T\left(\tilde{\Psi}_k^{|\nabla\Delta|,|\Delta|} + h_y^{-1} h \tilde{\Psi}_{k,1}^{|\nabla\mathcal{L}_1|,|\mathcal{L}_1|} + h_y^{-1} h^3 \tilde{\Psi}_{k,0}^{|\nabla|,|I|}\right) v_h(X_k)}{v_h(X_k)^T\left(\tilde{\Psi}_k^{\Delta,\Delta} + h\tilde{\Psi}_{k,1}^{\mathcal{L}_1,\mathcal{L}_1} + h^3\tilde{\Psi}_{k,0}^{I,I}\right) v_h(X_k)} \\
& \le 2C_I\left(\frac{h_y^d}{h} K_\Delta^{|\nabla\Delta|,|\Delta|} + h_y^{d-1} K_\Delta^{|\nabla\mathcal{L}_1|,|\mathcal{L}_1|,1} + h_y^{d-1} h^2 K_\Delta^{|\nabla|,|I|,0}\right).
\end{aligned}$$

The first coefficient is the critical one, so if we choose

$$(4.26)\qquad h_y \le \left(\frac{\tau h}{2C_I \max\left(K_I^{|\nabla|,|I|}, K_\Delta^{|\nabla\Delta|,|\Delta|}\right)}\right)^{\frac{1}{d}},$$

then the relative error is in both cases bounded by τ locally and globally. □

5. Convergence and error estimates for RBF-FD-LS.

In this section, we derive stability, convergence, and error estimates for the RBF-FD-LS method. When solving the least squares problem numerically in the form (3.6), we have not experienced practical problems with well-posedness. However, from the theoretical perspective it simplifies the analysis to have the Dirichlet boundary conditions imposed strongly as in (3.17), which means that the error is zero at the node set $X_{\partial\Omega_0}$. When performing the analysis, we assume that the weights $\beta_i = 1$, $i = 0, 1, 2$, in (3.14). Scaling is discussed separately in subsection 5.4. Before stating the global error estimate, we prove coercivity for the continuous bilinear form in a homogeneous space, and then relate the discrete bilinear form to the continuous bilinear form.

5.1. Coercivity of the continuous bilinear form in a homogeneous space.

We investigate the coercivity of the continuous bilinear form (2.4) for functions $v \in V^0 = \{v \in W_2^2(\Omega) \mid v(y)|_{\partial\Omega_0=0}\}$. Given the smoothness assumptions on the domain Ω and the function v, a Poincaré–Friedrich inequality holds with the boundary data given on some part of the boundary [6]. This can be seen if the inequality is shown using integration along paths from points on the Dirichlet boundary to points in the domain.

$$(5.1)\qquad \|v\|^2_\Omega \le C_P^2\left(\|v\|^2_{\partial\Omega_0} + \|\nabla v\|^2_\Omega\right) = C_P^2\|\nabla v\|^2_\Omega, \quad v \in W_2^1(\Omega).$$

We also need a trace inequality that relates the solution on (any part of) the boundary to the solution in the interior. The following inequality [9, Theorem 1.6.6], [16, Theorem A.4] holds for domains with Lipschitz or smooth boundary:

$$(5.2)\quad \|v\|^2_{\partial\Omega_i} \le C_T^2\|v\|_\Omega\left(\|v\|^2_\Omega + \|\nabla v\|^2_\Omega\right)^{\frac{1}{2}} \le C_T^2 C_P\sqrt{C_P^2+1}\|\nabla v\|^2_\Omega, \quad v \in W_2^1(\Omega),$$

where (5.1) was used for the second inequality. Then we have Green's first identity that can be derived from the divergence theorem

$$(5.3)\qquad \int_\Omega \nabla u\cdot\nabla v = \int_{\partial\Omega} u\frac{\partial v}{\partial n} - \int_\Omega u\Delta v, \quad u \in W_2^1(\Omega), \quad v \in W_2^2(\Omega),$$

leading to

$$\begin{aligned}\|\nabla v\|_\Omega^2 &\le \|v\|_{\partial\Omega_0}\|\partial v/\partial n\|_{\partial\Omega_0} + \|v\|_{\partial\Omega_1}\|\partial v/\partial n\|_{\partial\Omega_1} + \|v\|_\Omega\|\Delta v\|_\Omega \\ &= \|v\|_{\partial\Omega_1}\|\partial v/\partial n\|_{\partial\Omega_1} + \|v\|_\Omega\|\Delta v\|_\Omega,\end{aligned} \tag{5.4}$$

where we separated the boundary integral into two parts due to the structure of our specific problem and used that functions in V^0 vanish on $\partial\Omega_0$.

To show coercivity, we start from (5.4), then use the trace inequality (5.2) on the first term, and use the Poincaré inequality (5.1) on the second term.

$$\begin{aligned}\|\nabla v\|_\Omega^2 &\le \|v\|_{\partial\Omega_1}\|\partial v/\partial n\|_{\partial\Omega_1} + \|v\|_\Omega\|\Delta v\|_\Omega \\ &\le C_T\sqrt{C_P}\sqrt[4]{C_P^2+1}\|\nabla v\|_\Omega\|\partial v/\partial n\|_{\partial\Omega_1} + C_P\|\nabla v\|_\Omega\|\Delta v\|_\Omega.\end{aligned} \tag{5.5}$$

Dividing through by the gradient norm, squaring the result, and using $(a+b)^2 \le 2a^2+2b^2$ leads to

$$\|\nabla v\|_\Omega^2 \le 2C_T^2C_P\sqrt{C_P^2+1}\|\partial v/\partial n\|_{\partial\Omega_1}^2 + 2C_P^2\|\Delta v\|_\Omega^2. \tag{5.6}$$

Let $C_1^2 = 2C_T^2C_P^2\sqrt{1+\frac{1}{C_P^2}}$ and $C_2^2 = 2C_P^2$. Using (5.1) one more time, we have

$$\|v\|_\Omega^2 \le C_P^2\|\nabla v\|_\Omega^2 \le C_P^2\max(C_1^2, C_2^2)(\|\partial v/\partial n\|_{\partial\Omega_1}^2 + \|\Delta v\|_\Omega^2). \tag{5.7}$$

If we finally let $C^2 = C_P^2\max(C_1^2, C_2^2)$, we have the coercivity result

$$C^2 a(v,v) \ge \|v\|_\Omega^2, \quad v \in V^0. \tag{5.8}$$

5.2. Coercivity of the discrete bilinear form in a homogeneous space.

THEOREM 5.1. *If Theorems* 4.2 *and* 4.3 *hold,* h_y *is chosen according to Theorem* 4.4*, and the continuous bilinear form is coercive, then the discrete bilinear form is coercive for* $v_h \in V_h^0$ *such that*

$$\|v_h\|_{\ell_2(\Omega)} \le C^2\frac{(1+\tau)(1+\eta_0)(1+\eta_a)}{(1-\tau)}a_h(v_h,v_h) \equiv C_h a_h(v_h,v_h). \tag{5.9}$$

Proof. Since $v = S_2^0 \in W_2^2(\Omega)$ the continuous coercivity property holds for v, showing that $a(v,v) > 0$ and hence $a^*(v_h,v_h) > 0$, and we can use (4.21), (4.11), (5.8), (4.12), and (4.22) in sequence. ◻

5.3. The global error estimate.

THEOREM 5.2. *Consider the least squares problem* (3.17) *with properties* (3.18) *and* (3.19)*. If Theorems* 4.2 *and* 4.3 *hold,* h_y *is chosen according to Theorem* 4.4*, and Theorem* 5.1 *holds for the trial space error* $e_h = u_h + u_h^0 - I_h(u) \in V_h^0$*, then the error* $e = u_h + u_h^0 - u$ *satisfies*

$$\|e\|_{\ell_2(\Omega)} \le C\left(\frac{(1+\tau)(1+\eta_0)(1+\eta_a)}{(1-\tau)}\right)^{\frac{1}{2}}\sqrt{2a_h(e_I,e_I)} + \|e_I\|_{\ell_2(\Omega)}, \tag{5.10}$$

where $e_I = I_h(u) - u$ *is the interpolation error.*

Proof. The error $e = u_h + u_h^0 - u$ does not lie in the trial space unless u lies in the trial space. Also, $u_h + u_h^0$ does not in general interpolate u. We use the interpolant $I_h(u) \in V_h$ as an auxiliary function to write $e = u_h + u_h^0 - u = (u_h + u_h^0 - I_h(u)) + (I_h(u) - u) = e_h + e_I$. The first term e_h has nodal values $e(X)$ and $e_h \in V_h^0$, since the Dirichlet condition is imposed strongly. The second term e_I is the interpolation error, which is not in the trial space. We split the error to get

$$\|e\|_{\ell_2(\Omega)} = \|e_h + e_I\|_{\ell_2(\Omega)} \leq \|e_h\|_{\ell_2(\Omega)} + \|e_I\|_{\ell_2(\Omega)}. \tag{5.11}$$

For the trial space error, we start from Theorem 5.1, and then use (3.19), together with $|a_h(u, v)| \leq \frac{1}{2}a_h(u, u) + \frac{1}{2}a_h(v, v)$, and (3.18)

$$\begin{aligned}\frac{1}{C_h^2}\|e_h\|^2_{\ell_2(\Omega)} &\leq a_h(e_h, e_h) = a_h(e - e_I, e_h) = a_h(-e_I, e_h) = a_h(-e_I, e - e_I)\\ &= a_h(-e_I, e) + a_h(e_I, e_I) \leq 0.5a_h(e, e) + 1.5a_h(e_I, e_I)\\ &\leq 2a_h(e_I, e_I), \qquad (5.12)\end{aligned}$$

leading to

$$\|e_h\|_{\ell_2(\Omega)} \leq C_h\sqrt{2a_h(e_I, e_I)}. \tag{5.13}$$ □

Combining (5.11) and (5.13) provides the final result (5.10).

5.4. The details of the global error estimate including scaling. The components of the global error estimate are now in place, and we can discuss their properties as well as the question about scaling of the different terms in the bilinear form. Equation (4.3) for the interpolation error $e_I = I_h(u) - u$ yields

$$\|e_I\|_{\ell_2(\Omega)} \leq \alpha_0 h^{p+1}|u|_{W_\infty^{p+1}(\Omega)}, \tag{5.14}$$

and for the bilinear form applied to the interpolation error we have

$$a_h(e_I, e_I) \leq \left((\alpha_0 h^{p+1})^2 + (\alpha_1 h^p)^2 + (\alpha_2 h^{p-1})^2\right)|u|^2_{W_\infty^{p+1}(\Omega)}. \tag{5.15}$$

Noting that $a^2 + b^2 + \cdots \leq (a + b + \cdots)^2$ for positive numbers, we insert all terms in the global error estimate (5.10), to get

(5.16)
$$\|e\|_{\ell_2(\Omega)} \leq \sqrt{2}C\left(\frac{(1+\tau)(1+\eta_0)(1+\eta_a)}{(1-\tau)}\right)^{\frac{1}{2}}\left(\tilde{\alpha}_0 h^{p+1} + \alpha_1 h^p + \alpha_2 h^{p-1}\right)|u|_{W_\infty^{p+1}(\Omega)},$$

where $\tilde{\alpha}_0 = \alpha_0(1 + \frac{1}{\sqrt{2}C_h})$. The error estimate tells us that h should be chosen to resolve the solution function, while $h_y \leq h$ should be chosen to resolve the integrals of the trial space error and its derivatives.

The error expression (5.16) indicates that the error may be somewhat improved, by adjusting the scale factors β_i in (3.14). The three terms in the sum change directly with the scale factors, while the stability constant depends on the worst case, leading to

$$C_\beta = \sqrt{2}C_P^2 \max\left(\frac{C_T\sqrt[4]{1+\frac{1}{C_P^2}}}{\beta_1}, \frac{1}{\beta_2}\right) = C_P \max\left(\frac{C_1}{\beta_1}, \frac{C_2}{\beta_2}\right). \tag{5.17}$$

Choosing the scaling such that $C_1/\beta_1 = C_2/\beta_2$ can in principle reduce the overall scaled error estimate. However, since we do not know the constants a priori, we use $\beta_1 = \beta_2 = 1$ in the numerical experiments.

The scaling β_0 of the Dirichlet condition does not affect the stability constant, but it can be beneficial to increase β_0 to reduce the errors near the boundary. The largest scaling such that the order of the Dirichlet term does not dominate the order of the Neumann term is $\beta_0 = \mathcal{O}(h^{-1})$. This scaling strategy is evaluated numerically in section 6 and is shown to perform well.

6. Numerical study.

In this section, we investigate the convergence, stability, and efficiency of least squares–based RBF-FD, compared with collocation-based RBF-FD. The two methods are tested in two flavors: with additional ghost points (RBF-FD-LS-Ghost, RBF-FD-C-Ghost) and without ghost points (RBF-FD-LS, RBF-FD-C). We solve the PDE problem (2.1) using the scaling (3.14) with $\beta_0 = h^{-1}$, $\beta_1 = 1$, and $\beta_2 = 1$ on a domain with an outer boundary defined in polar coordinates as $r(\theta) = 1 + \frac{1}{10}(\sin(7\theta) + \sin(\theta))$, $\theta \in [-\pi, \pi)$, and with the Dirichlet boundary $\partial\Omega_0$ defined by $r(\theta)$, $\theta \in [-\pi, 0)$, and the Neumann boundary $\partial\Omega_1$ defined by $r(\theta)$, $\theta \in [0, \pi)$. The node sets X and Y are generated using *DistMesh* [21]. However, in order to enforce $X \subset Y$, we modify the Y node set such that for each $x_k \in X$ we find the closest point $\tilde{y}_j$ in the initial node set $\tilde{Y}$, and then let $y_j = x_k$ in the final node set Y. The Dirichlet boundary conditions are enforced exactly at $X_{\partial\Omega_0}$ according to (3.17).

The domain Ω, an example of the spatial error distribution, the node sets X and Y, and the Voronoi diagram corresponding to the node set X are shown in Figure 3. All experiments were run in MATLAB on a laptop with an Intel i7-7500U processor and 16 GB of RAM. The code that was used to generate the rectangular and square RBF-FD matrices is available in [31].

In the numerical study, we focus on the three main method parameters: the node distance h, the oversampling parameter $q = \frac{M}{N} = (h/h_y)^2$, which determines h_y, and the polynomial degree p. When nothing else is stated, we use the default value $q = 3$ for the oversampling. When applicable, the ghost points are added to the node set X

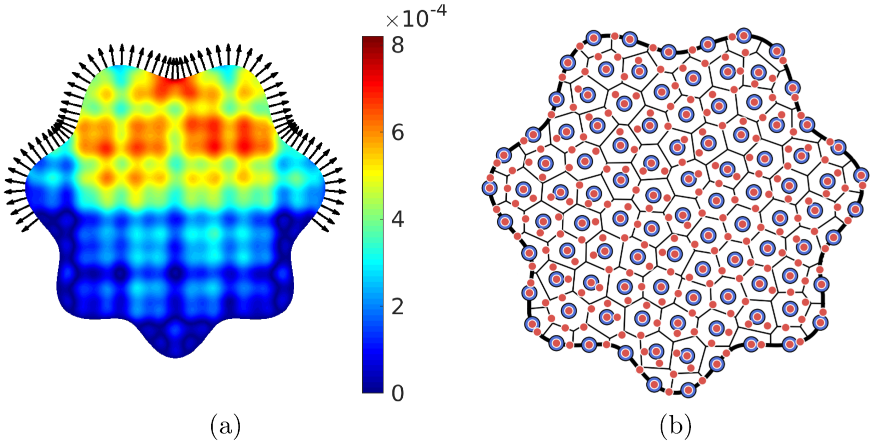

FIG. 3. (a) *A contour plot of the absolute error distribution over the domain Ω for the truncated nonanalytic solution function* (6.4) *when $h = 0.02$, $p = 3$, $q = 3$. The outward normals indicate the locations where the Neumann condition is enforced.* (b) *The X node set (large blue markers) and the Y node set (small red markers) are shown together with the Voronoi diagram for the X node set. Each Voronoi cell contains on average three Y node points.*

as an additional layer outside Ω. The layer is generated by projecting every boundary point a distance h in the normal direction. The matrix $\bar{D}_h$ and the right-hand-side vector $\bar{F}$ are then modified such that the Laplacian is also sampled on $\partial\Omega$. In the RBF-FD-LS case, the size of $\bar{D}_h$ then grows from $M \times N$ to $(M + N_g) \times (N + N_g)$ and in the RBF-FD-C case from $N \times N$ to $(N + N_g) \times (N + N_g)$, where N_g is the number of ghost points equal to the number of X points at the boundary. The stencil size in all experiments is $n = 2m$, where m is the dimension of the polynomial space. In the convergence experiments, we measure the relative ℓ_2-error

$$\|e(Y)\|_{\ell_2(\Omega)} = \frac{\|u_h(Y) - u(Y)\|_{\ell_2(\Omega)}}{\|u(Y)\|_{\ell_2(\Omega)}}. \tag{6.1}$$

We also investigate the stability norm by defining it as the ratio of the largest singular value of $\bar{E}_h(Y, \tilde{X})$ and the smallest singular value of $\bar{D}_h(Y, \tilde{X})$. The stability norm provides a numerical value for the coercivity constant C_h in the global error estimate (5.10). Using (3.17) for the interior solution, we have

$$\begin{aligned} \|u_h(Y)\|_{\ell_2(\Omega)} &= \|\bar{E}_h(Y, \tilde{X})\, u(\tilde{X})\|_2 = \|\bar{E}_h\, \bar{D}_h^+\, \tilde{F}(Y)\|_2 \\ &\leq \|\bar{E}_h\|_2\, \|\bar{D}_h^+\|_2\, \|\tilde{F}(Y)\|_2 = \frac{\sigma_{\max}(\bar{E}_h)}{\sigma_{\min}(\bar{D}_h)} \|\tilde{F}(Y)\|_2. \end{aligned} \tag{6.2}$$

6.1. Errors and convergence tests for different functions. Three solution functions which are different in nature are used to compute the right-hand-side data of (2.1). The purpose of this test is to show the differences in the error behavior and to pick one solution function for which the method is later on tested more extensively. Additionally, we solve the PDE problem with only Dirichlet boundary data. The following functions are considered:

$$u_1(x, y) = \sqrt{x^2 + y^2}, \tag{6.3}$$

$$u_2(x, y) = \sum_{k=0}^{5} e^{-\sqrt{2^k}} \left(\cos(2^k x) + \cos(2^k y)\right), \tag{6.4}$$

$$u_3(x, y) = \sin\left(2(x - 0.1)^2\right) \cos\left((x - 0.3)^2\right) + \frac{\sin\left(2(y - 0.5)^2\right)^2}{1 + 2x^2 + y^2}, \tag{6.5}$$

which are referred to by the names distance, truncated nonanalytic, and rational sine, respectively. The polynomial degree $p = 5$ is used for computing the local interpolation matrices. The error under refinement of h is displayed in Figure 4.

The accuracy of RBF-FD-LS is better than that of RBF-FD-C for all solution functions when both the Neumann and Dirichlet conditions are present. The convergence rates for the truncated nonanalytic and rational sine functions are $k = 4.9$ and $k = 4.8$, respectively, which agrees with the error estimate (5.16) since $k \geq p - 1 = 4$. When ghost points are utilized, the accuracy is better compared to when no ghost points are utilized. Comparing RBF-FD-LS-Ghost with RBF-FD-C-Ghost we see that the accuracy of the former is generally better compared with the accuracy of the latter. Next, comparing RBF-FD-C-Ghost to RBF-FD-LS we see that RBF-FD-LS is, in this test, overall more accurate, but the gap is smaller compared with the gap between RBF-FD-LS and RBF-FD-C. The convergence rate for the distance function is $k = 0.8 < 4\}$, but that is expected since it is a C^0 function.

When only the Dirichlet condition is imposed, the accuracy of RBF-FD-LS is better for the distance and rational sine functions. This is also the case for the

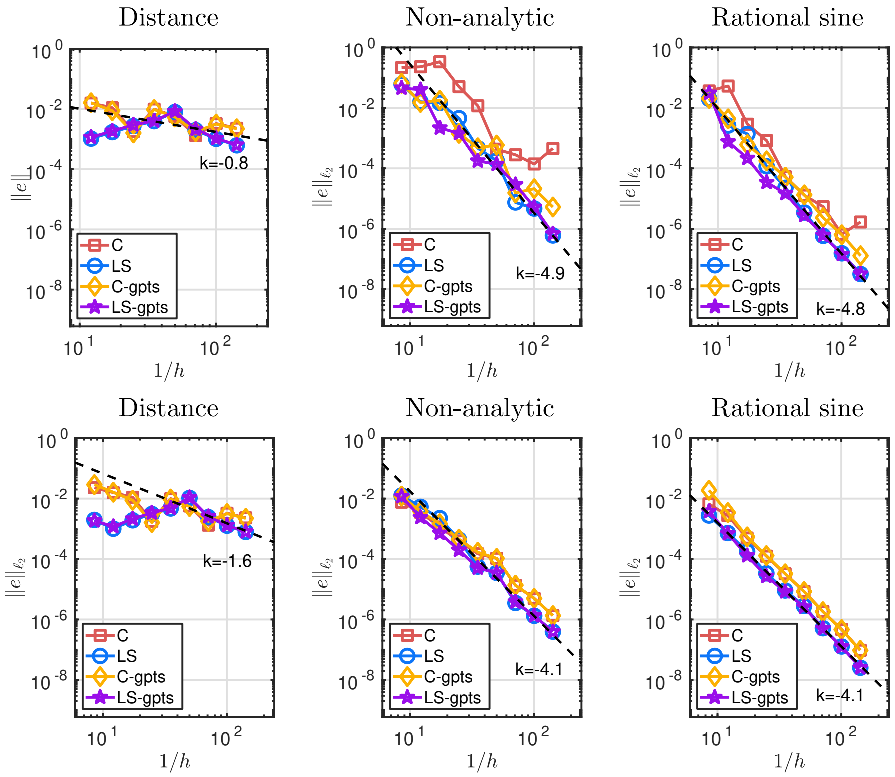

Fig. 4. *The errors of RBF-FD-LS and RBF-FD-C as a function of the inverse node distance for the three different solution functions defined in subsection* 6.1. *The first row of plots corresponds to solving* (2.1) *and the second row corresponds to solving the same problem with the Dirichlet condition on the whole boundary. The polynomial degree used to construct the interpolation matrices is* $p = 5$ *and the oversampling parameter* $q = 3$. *The number of node points in* X *ranges from* $N = 500$ *to* $N = 64000$.

truncated nonanalytic function, when h is small enough. The difference in error between RBF-FD-LS and RBF-FD-C is not as large as when both the Neumann and Dirichlet conditions are imposed. In this case the convergence rates for the truncated nonanalytic function and rational sine function are $k \geq p - 1 = 4$, similarly to the mixed conditions case. Ghost points in this case do not play a significant role in improving accuracy.

In the following subsections further experiments are made with the truncated nonanalytic solution function, which, due to its fine scale variation, is challenging to approximate.

6.2. Approximation properties under node refinement. We refine h (this increases the number of nodes N) and measure the approximation properties for different polynomial degrees in the local interpolation matrices (1.5). We denote this by h-refinement. The convergence as a function of the node distance is shown in Figure 5. We observe that the accuracy of RBF-FD-LS is better for each tested p compared with RBF-FD-C. The overall difference in the errors is larger for $p = 5$ compared with when $p = 3$ and 4. The convergence trend k of RBF-FD-LS is $k \geq p-1$ for every p. It is hard to evaluate the convergence trend of RBF-FD-C since the error behavior is unpredictable. The accuracy of RBF-FD-LS-Ghost is overall better for

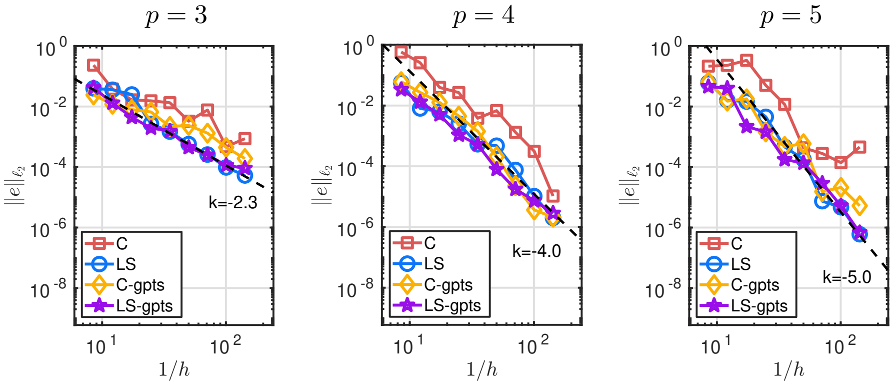

Fig. 5. *The RBF-FD-LS and RBF-FD-C methods are compared. The relative error as a function of the inverse node distance $1/h$ for a fixed oversampling parameter $q = 3$ and different polynomial degrees p used to form the local interpolation matrices is shown.*

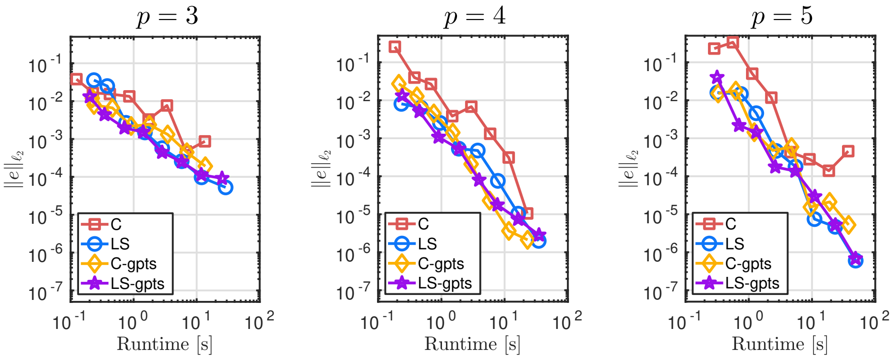

Fig. 6. *The RBF-FD-LS and RBF-FD-C methods are compared. The relative error as a function of the initialization runtime (R_1) measured in seconds for a fixed oversampling parameter $q = 3$ and different polynomial degrees p used to form the local interpolation matrices.*

all p compared with RBF-FD-C-Ghost. The gap between RBF-FD-LS and RBF-C-Ghost is smaller compared with the gap between RBF-FD-LS and RBF-FD-C, and in some points, RBF-FD-C-Ghost is more accurate than RBF-FD-LS.

Next, the relation between the error and the computational time (runtime) is investigated. It is important to note that a method with a smaller error/runtime ratio is more efficient. The runtime is divided into two steps shown in Figures 6 and 7:

R_1 : The closest neighbor search, forming and inverting the local interpolation matrices (1.5), and forming the evaluation and differentiation weights (3.3).

R_2 : Assembly of the PDE operator (3.4), and solution of the system of equations using *mldivide()* in MATLAB.

The node generation is considered as a preprocessing step and is therefore not included in the measurement.

We observe that the efficiency of RBF-FD-LS is better than that of RBF-FD-C for all considered p and both efficiency measurements: R_1 and R_2. When $p = 5$, the difference in the efficiency is larger. The oversampling parameter q does not have a

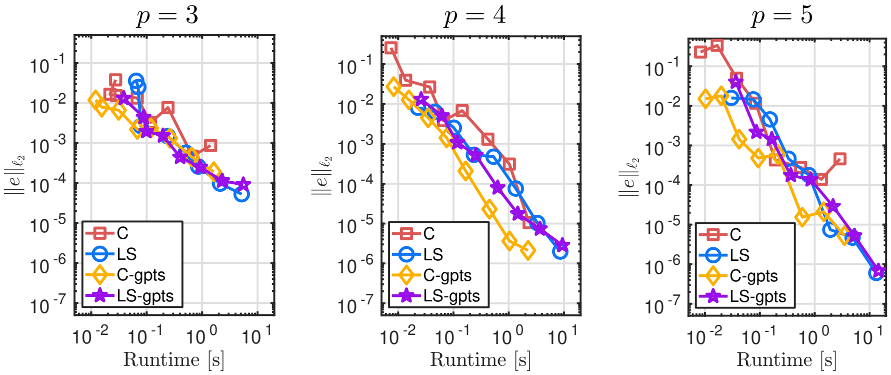

FIG. 7. *The RBF-FD-LS and RBF-FD-C methods are compared. The relative error as a function of the solving runtime (R_2) measured in seconds for a fixed oversampling parameter $q = 3$ and different polynomial degrees p used to form the local interpolation matrices.*

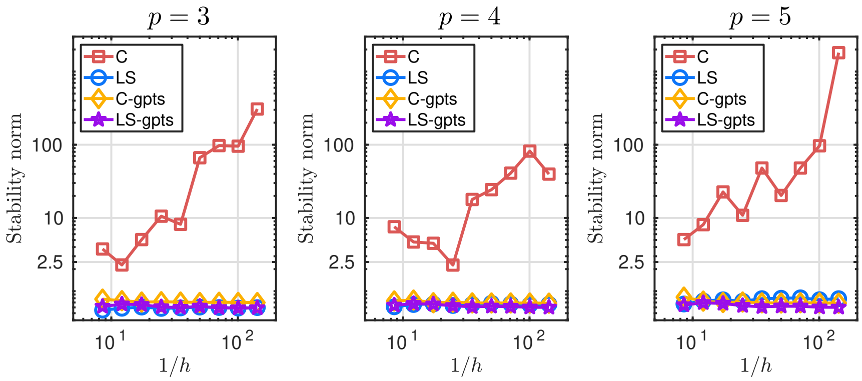

FIG. 8. *The stability norm* (6.2) *as a function of* $1/h$ *when the oversampling parameter is* $q = 3$ *for different polynomial degrees* p.

decisive role when it comes to efficiency. We expect the runtime to be dominated by the solution of the overdetermined linear system. For a dense matrix, the cost grows linearly with $M = qN$ for a fixed N. We expect a similar behavior for our sparse system. The added cost is compensated for by the improved accuracy. Figure 10 shows the error improvement with q.

RBF-FD-C-Ghost behaves similarly to RBF-FD-LS and RBF-FD-LS-Ghost concerning the efficiency R_1. In the R_2 case, RBF-FD-C-Ghost outperforms all other methods; however, the magnitude of the runtime in R_2 is at least one order smaller compared with R_1. By observing the efficiency as a function of the total runtime $R_1 + R_2$, the result of R_1 is dominating. Thus, the three methods overall behave similarly in terms of efficiency.

The stability norm (6.2) as a function of $1/h$ is studied in Figure 8. We observe that the stability norm of RBF-FD-LS is almost constant for all polynomial degrees p which we considered. This corresponds with the error estimate (5.16). When $h_y = h/\sqrt{q}$, the integration error τ goes to zero as h/q, and the factor $\frac{\sqrt{1+\tau}}{\sqrt{1-\tau}}$ in the stability constant C_h approaches 1. The stability norm of the RBF-FD collocation method does not follow a pattern for the given PDE, parameters, and node sets. Here

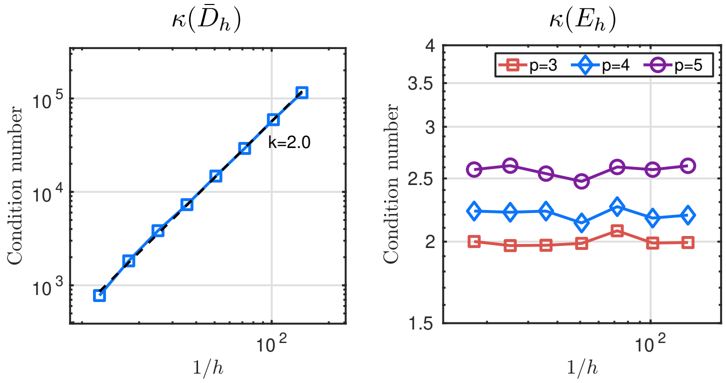

FIG. 9. *Left: The condition number of the PDE matrix $\bar{D}_h$ as a function of the inverse node distance when the polynomial degree used for representing the trial space is $p = 3$. Right: The condition number of the evaluation matrix E_h when the polynomial degrees used for representing the trial space are $p = 3$, 4, and 5.*

we emphasize that this behavior is not caused by the RBF-FD trial space but rather by the collocation formulation in which the PDE is solved. An interesting behavior is observed in the RBF-FD-C-Ghost case, where the stability norm is constant. While it is possible to attribute that to the imposition of ghost points, we argue that this occurs only due to the imposition of the extra Laplacian condition on the boundary points, which in turn introduces a stronger control over $\|\Delta u_h\|_{\ell_2}$: a key factor when it comes to the invertibility of $\bar{D}_h$ and thus the stability norm. To confirm this claim, we made a side experiment. The extra Laplacian condition was enforced at the boundary points, but no ghost points were used and no oversampling was employed. The resulting system of equations was rectangular only due to the imposition of the extra Laplacian conditions. The stability norm remained constant.

The condition number of a rectangular or square matrix A is defined by $\kappa(A) = \|A\|_2 \, \|A^+\|_2 = \sigma_{max}(A)/\sigma_{min}(A)$. In Figure 9 we show the condition numbers for the two matrices involved in RBF-FD-LS: $\bar{D}_h$ and E_h. We observe that $\kappa(\bar{D}_h)$ grows with $\frac{1}{h^2}$ for $p = 3$. The results are almost identical for $p = 4$ and 5. This is an expected (optimal) growth, since $\bar{D}_h$ is a numerical second-order differentiation operator, which has an inverse quadratic dependence on h when the stencil size is kept constant. On the other hand $\kappa(E_h)$ is constant with respect to h for all p, which is also an expected result, since E_h is a numerical interpolation operator, which does not by itself yield a dependence on h when the stencil size is kept constant.

6.3. Approximation properties as the oversampling is increased. In this section we build an understanding of the error and stability behavior for different choices of $h_y = h/\sqrt{q}$, when h is fixed at $h = 0.08$ (underresolved case) and at $h = 0.02$ (well-resolved case). Three polynomial degrees $p = 3$, 4, and 5 are used for the local interpolation matrices (1.5). The exact solution is chosen to be the truncated nonanalytic function (6.4). The convergence study for RBF-FD-LS is displayed in Figure 10. For both the underresolved and well-resolved cases, the error decays and then levels out as h_y becomes small enough. This behavior matches the error estimate (5.16) for the case when h is fixed. As $h_y \to 0$ the term $\left(\frac{1+\tau}{1-\tau}\right)^{\frac{1}{2}} \to 1$ from a larger value, and therefore $\|e\|_{\ell_2(\Omega)}$ levels out. The stability norm behavior is shown in Figure 11, from which we observe that in both the well-resolved and underresolved

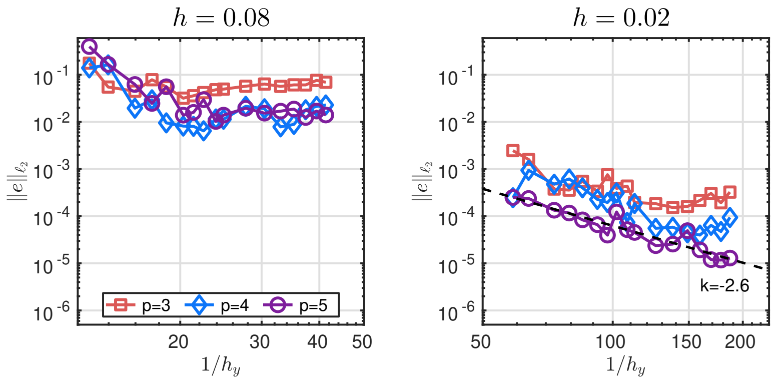

FIG. 10. *The error against* $1/h_y$ *which is the average distance in the* Y *node set for different choices of the polynomial degree* p *used to form the trial space. The average internodal distance in the node set* X *was for the left plot fixed at* $h = 0.08$ *and for the right plot fixed and* $h = 0.02$*. The values of* h_y *were computed from* $q = (1.1,\ 1.3,\ 1.7,\ 2,\ 2.3,\ 2.7,\ 3,\ 3.3,\ 3.7,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9,\ 10,\ 11)$.

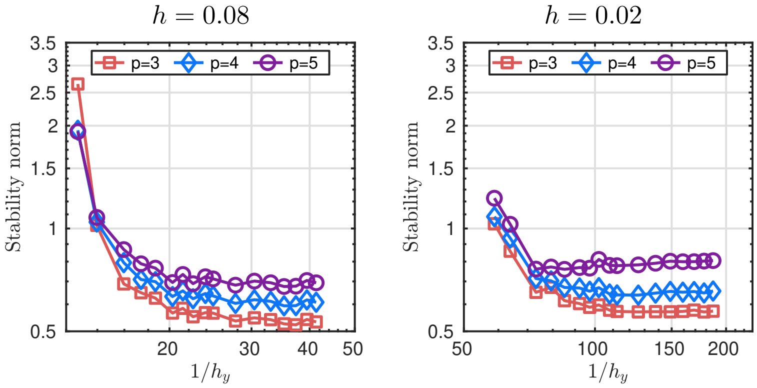

FIG. 11. *Both plots show the stability norm* (6.2) *as a function of* $1/h_y$*, the average inverse node distance in the point set* Y*. Different choices of the polynomial degree* p *were used to form the trial space. The average internodal distance* h *in the node set* X *was for the left plot fixed at* $h = 0.08$ *and for the right plot fixed at* $h = 0.02$*, illustrating the underresolved and the well-resolved case. The values of* h_y *were computed from* $q = (1.1,\ 1.3,\ 1.7,\ 2,\ 2.3,\ 2.7,\ 3,\ 3.3,\ 3.7,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9,\ 10,\ 11)$.

cases, the norm first rapidly decays and then flattens out when h_y is small enough. The approximate point when the stability norm starts to flatten out is at $1/h_y \approx 28$ (corresponding to $q = 3.7$) for the underresolved case and at $1/h_y \approx 70$ ($q = 3.3$) for the well-resolved case.

6.4. Eigenvalue spectrum as the oversampling is increased. In the case of the least squares matrices, we use the relation $u(Y) = E_h u(X)$ to arrive at $E_h^+ u(Y) = u(X)$, which is then used in the discretized PDE $\bar{D}_h u(X) = \bar{F}(Y)$ to arrive at

$$\bar{D}_h E_h^+ u(Y) = \bar{F}(Y).$$

We then investigate the eigenvalues of the square $M \times M$ matrix $\bar{D}_h E_h^+$. Both matrices $\bar{D}_h$ and E_h are rectangular of size $M \times N$, thus the rank of each is at most N. The

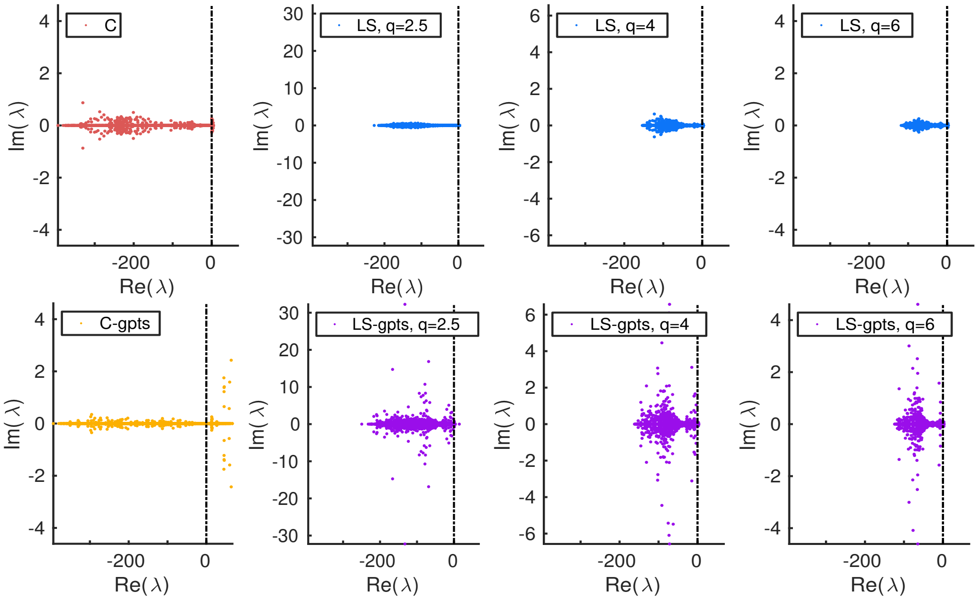

FIG. 12. *Eigenvalues of $E\bar{D}^+$ as the oversampling parameter q is increased, where $\bar{D}$ discretizes the Poisson equation with Dirichlet and Neumann boundary conditions imposed on two disjoint parts of the boundary of the domain. In this test the number of nodes is $N = 1000$ and the polynomial degree used to construct the local approximations is $p = 4$.*

rank of $\bar{D}_h E_h^+$ can then not be larger than N, implying that there will always exist a nullspace of size $M - N$. This is not a problem when we use the method for solving PDEs as we always solve for the unique N-dimensional least squares solution.

In Figure 12, we display the eigenvalue spectrum of the PDE matrix for $N = 1000$ and an increasing oversampling parameter q. We observe that the real part of the spectra in the least squares case shrinks as q is increased. The spectra of the collocation matrices have a larger negative real part compared to the spectra of the least squares matrices. In addition, in Figure 13, we also display the eigenvalue spectra of the discretized first order operator $-g \cdot \nabla u$, where $g = [0, 1]$. We only impose the Dirichlet boundary condition at the location of the polar angle $\theta \in [0, \pi]$. In this case, the least squares cases have less distinct behavior. However, we can observe that the negative and positive real parts of the spectrum are moving slightly toward the imaginary axis as the oversampling is increased. The increase in oversampling improves the eigenvalue spectra of the rectangular differentiation matrices in both eigenvalue examples.

6.5. Approximation properties as the polynomial degree is increased. Here we increase the number of points per stencil, together with increasing the polynomial degree p used to form the stencil-based interpolant (1.5), while the distance h between the stencil points is kept the same. This is denoted by p-refinement. We consider polynomial degrees up to $p = 12$ in order to test the limits of the method. Two different solution functions are considered, the truncated nonanalytic function (6.4) and the rational sine function (6.5). The methods RBF-FD-LS, RBF-FD-LS-Ghost, RBF-FD-C, and RBF-FD-C-Ghost are compared when h is fixed at $h = 0.08$ (underresolved case) and at $h = 0.02$ (well-resolved case). In Figure 14, we see that the error for both resolutions and both manufactured solutions is smaller for RBF-FD-LS

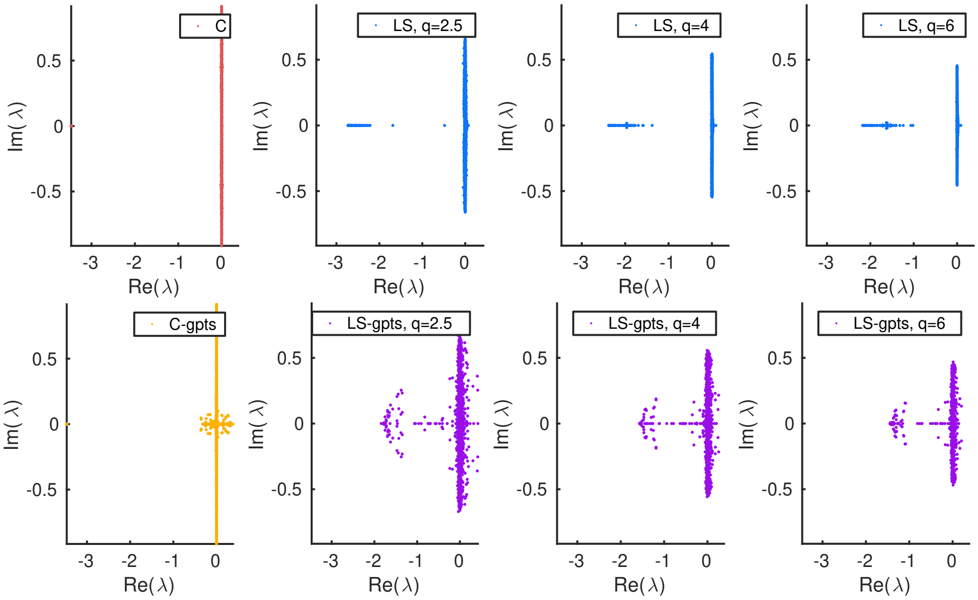

FIG. 13. *Eigenvalues of* $E\bar{D}^+$ *as the oversampling parameter* q *is increased, where* $\bar{D}$ *discretizes the advection equation with Dirichlet boundary condition at the inflow boundary of a domain* Ω. *In this test the number of nodes is* $N = 1000$ *and the polynomial degree used to construct the local approximations is* $p = 4$.

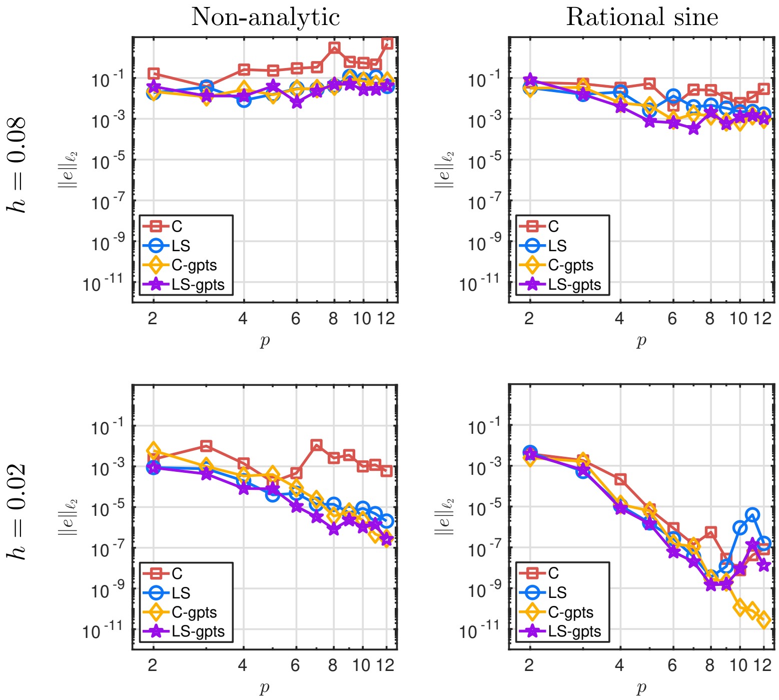

FIG. 14. *The relative error in the p-refinement mode: the polynomial degree used to form the stencil-based interpolation matrix* (1.5) *is increased, while the internodal distance between the stencil points is fixed at* $h = 0.08$ *and* $h = 0.02$.

compared with RBF-FD-C. For the underresolved case ($h = 0.08$)), there is some improvement of the error when increasing p for the rational sine function. The results are worse for the truncated nonanalytic function that has large derivatives and requires higher resolution. In this case, the error increases for $p > 4$. In the well-resolved case ($h = 0.02$), we observe convergence with p in all cases except RBF-FD-C for the nonanalytic function. For $p \geq 8$, round-off errors prevent further convergence for RBF-FD-LS. The convergence trend for RBF-FD-C levels out earlier than for RBF-FD-LS. Comparing RBF-FD-LS-Ghost with RBF-FD-C-Ghost we can see that RBF-FD-LS-Ghost is more accurate in the case that we are considering.

The stability norm as a function of p is shown in Figure 15. It has an increasing trend for all methods. Based on (4.8) and Theorem 4.2, we expect the bounds to grow with the stencil size, which depends on p. In the ideal case, the maximum value for any cardinal function is one, but here, there is an exponential growth of these functions with p, especially for cardinal functions close to the boundary. The largest weight for $p = 12$ has $|w| \approx 3.2$. Cardinal functions for different values of p are illustrated in Figure 16. The condition numbers of the local interpolation matrices $\tilde{A}_k$ also grow exponentially with p, and for $p \gtrsim 12$ prevent accurate numerical evaluation of the weights. The results also show that both methods with ghost points have a smaller growth in the stability norm for an increasing p, compared to the methods which do not use ghost points. Since all of the stencils around the boundary are less skewed when using ghost points, it is expected that the stability improves.

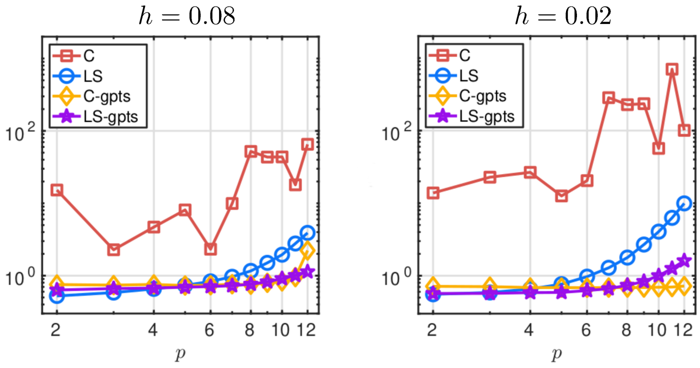

FIG. 15. *The stability norm in the p-refinement mode: the polynomial degree used to form the stencil-based interpolation matrix* (1.5) *is increased, while the internodal distance between the stencil points h is fixed at $h = 0.08$ and $h = 0.02$.*

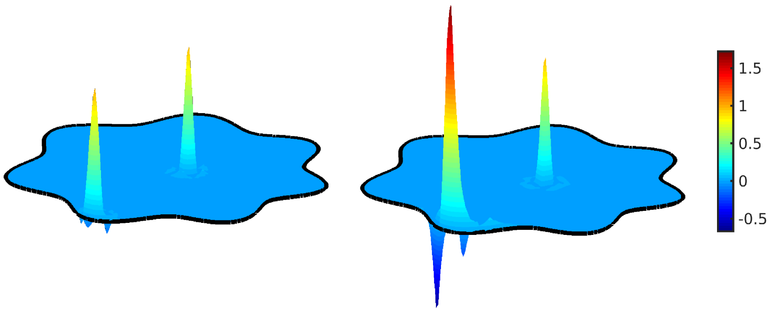

FIG. 16. *Example of two cardinal functions placed on the boundary of Ω and in the interior of Ω for $p = 3$ (left) and $p = 12$ (right). For both cases the internodal distance is $h = 0.06$.*

6.6. Approximation properties under node refinement in three dimensions. In this section we solve the PDE problem (2.1) in the same way as in the previous sections, but now in three dimensions. We compare RBF-FD-LS and RBF-FD-C with ghost points and without ghost points. The stencil sizes are in all cases $n = 2m$, where m is the dimension of the polynomial space.

The norm scaling for the least squares methods is in the 3D case chosen as $\beta = h_y^{3/2}$ for the Laplacian equations and $\beta = h_y$ for both boundary conditions, according to the relation (3.14). The equation scaling β_0, β_1, and β_2 for the Dirichlet boundary, the Neumann boundary, and the Laplacian condition is given by $\beta_0 = 1/h$, $\beta_1 = 1$, $\beta_2 = 1$, which is the same scaling as used in the 2D cases.

The 3D domain in spherical coordinates is given by

$$r(\theta,\phi) = \left(1 + \sin(2\sin(\phi)\sin(\theta))^2\ \sin(2\sin(\phi)\cos(\theta))^2\ \sin(2\cos(\phi))^2\right)^{\frac{1}{2}},$$

where θ is the longitude angle $\theta \in [-\pi, \pi)$ and ϕ is a latitude angle $\phi \in [-\frac{\pi}{2}, \frac{\pi}{2})$. The right-hand sides of the PDE (2.1) are computed based on a solution function:

$$u = \sin(3\pi\, x\, y\, z).$$

The coordinates (x, y, z) of the mixed boundary conditions are given by

$$(x, y, z)|_{\partial\Omega_0} = \{(x, y, z)|_{\partial\Omega}\,|\, z < 0.7\}, \quad (x, y, z)|_{\partial\Omega_1} = \{(x, y, z)|_{\partial\Omega}\,|\, z \geq 0.7\},$$

where the first set corresponds to the location of the Dirichlet boundary condition and the second to the location of the Neumann boundary condition.

In the previous experiments in two dimensions we mostly used the oversampling parameter $q = 3$. An equivalent sampling in three dimensions is motivated by knowing that 10 points in one dimension sample a fixed domain $[\Omega]^1$ equivalently well as 100 points sample $[\Omega]^2$ in two dimensions and 1000 points sample $[\Omega]^3$ in three dimensions. It follows that the relation $q_3 = q_2\sqrt{q_2}$ holds, where q_2 is a sampling in two dimensions and q_3 is an equivalent sampling in three dimensions. All of our 3D computations are therefore based on a choice of an oversampling parameter $q = \lceil 3\sqrt{3}\rceil = 6$. An instance of a solution function, together with a relative absolute error in the logarithmic scale, is given in Figure 17. We can observe that the largest errors appear close to the boundary $\partial\Omega_1$, where the Neumann condition is imposed. The error at

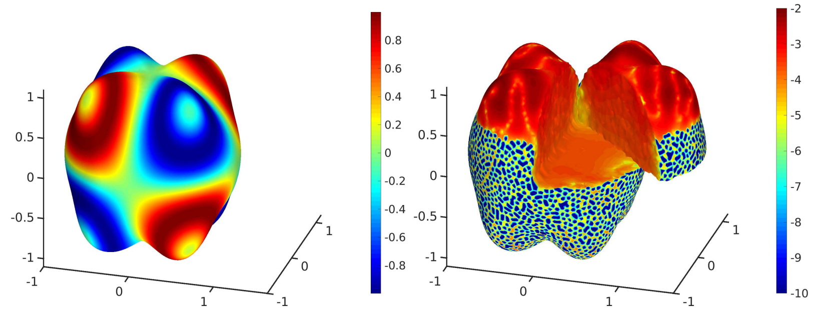

FIG. 17. *Solution function (left) and spatial distribution of the relative absolute error in the logarithmic scale (right) when h is chosen such that it corresponds to $N = 24000$ points placed over Ω. The polynomial degree chosen to construct local approximations is $p = 5$ and the oversampling parameter is $q = 6$.*

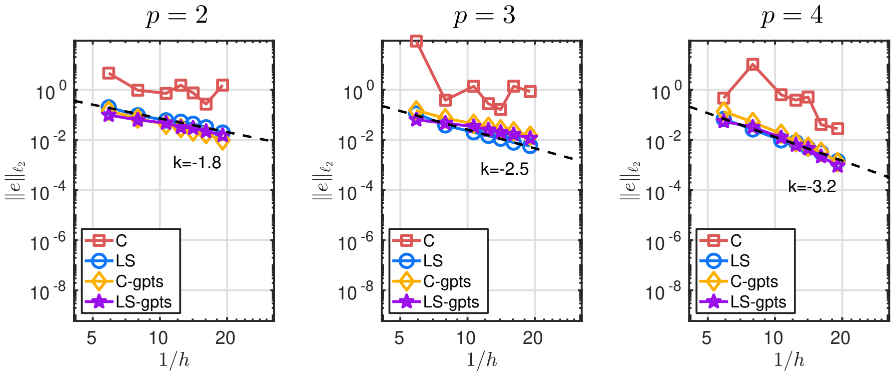

FIG. 18. *Error under nodal refinement in a 3D case for polynomial degrees $p = 3$, 4, and 5. The oversampling parameter is $q = 6$ and the chosen $1/h$ corresponds to $N = 1000$, 2000, 4000, 6000, 8000, 12000, and 24000 points spread over a 3D domain.*

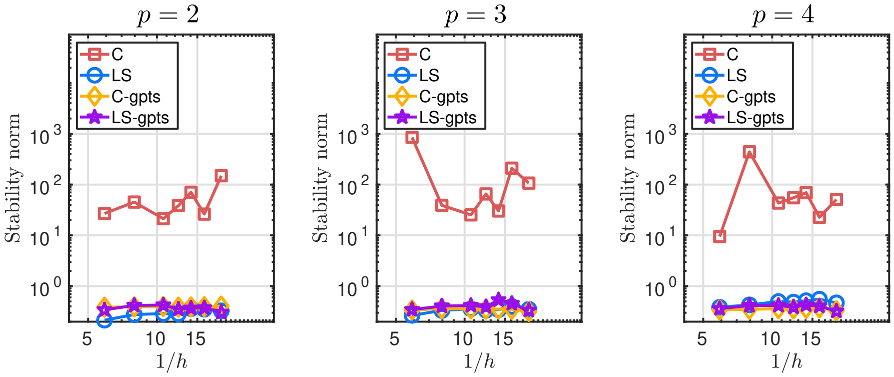

FIG. 19. *Stability norm under nodal refinement in a 3D case for polynomial degrees $p = 3$, 4, and 5. The oversampling parameter is $q = 6$ and the chosen $1/h$ corresponds to $N = 1000$, 2000, 4000, 6000, 8000, 12000, and 24000 points in a 3D domain.*

the locations of the Dirichlet boundary $\partial\Omega_0$ is very small in some points, since those are the points where this condition is enforced exactly. The error in other points at the same boundary is larger, since in those points the Dirichlet condition is not satisfied exactly. In Figure 18 we compute the error as the internodal distance h is decreased and the polynomial degrees $p = 2$, 3, and 4 are used to construct the local approximations. Our numerical results show that RBF-FD-LS, RBF-FD-LS-Ghost, and RBF-FD-C-Ghost methods share a similar accuracy, while RBF-FD-C behaves unpredictably.

A study related to the stability norm in three dimensions is given in Figure 19. The experiment confirms that the numerical (and theoretical) observations in two dimensions generalize to three dimensions as well, since the stability norm of RBF-FD-LS does not have unpredictable behavior.

7. Final remarks.

In this paper we introduced an enhancement of the collocation-based RBF-FD method where we instead use a least squares approach. The main method parameters are the node distance h, the evaluation node distance h_y, and the polynomial degree p used to form the stencil approximations. The least squares formulation led us to characterize the RBF-FD trial space as a piecewise continuous space with jumps that vanish together with the local approximation error, and to understand that $\bar{D}_h^T \bar{D}_h u = \bar{D}_h^T f$ reproduces the L_2 inner-products of the continuous least squares problem up to an error governed by h_y. This allowed us to prove well-posedness (stability) of RBF-FD-LS for an elliptic problem when h_y is small enough in relation to h. We also derived an error estimate in terms of the node distance, where the error decays with no less than order $p-1$ for the Poisson problem with Dirichlet and Neumann boundary conditions.

The experiments confirmed the theoretical observations in terms of the convergence trend as a sequence of h gets increasingly small. We also confirmed that as h is fixed at a small value, the stability norm and the error are improved as $h_y \to 0$, until both level out. This happens when the effect of the numerical integration becomes negligible.

An experimental comparison of RBF-FD-LS and RBF-FD-C with ghost points revealed that both methods are comparable in robustness. However, we believe that RBF-FD-LS has an advantage due to a better theoretical understanding (at least at the present moment) compared to RBF-FD-C with ghost points.

Overall, the numerical experiments indicated that RBF-FD-LS (no ghost points) for our model problem performs better than RBF-FD-C (no ghost points) in terms of

- the error against the exact solution for p-refinement and h-refinement,
- the stability properties,
- the efficiency.

The most important strength of the least squares formulation is the robustness of the numerical solution as h is decreased, which, according to our experience, is often lacking in the collocation formulation, especially in the presence of Neumann boundary conditions.

Acknowledgments. For valuable discussions we thank Murtazo Nazarov, Gunilla Kreiss, and Eva Breznik from Uppsala University and Axel Målqvist from Chalmers University of Technology. The third author thanks the Department of Information Technology, Uppsala University, for hosting his sabbatical visit in spring 2019.

REFERENCES

[1] C. Aistleitner, F. Pausinger, A. M. Svane, and R. F. Tichy, *On functions of bounded variation*, Math. Proc. Cambridge Philos. Soc., 162 (2017), pp. 405–418, https://doi.org/10.1017/S0305004116000633.

[2] G. A. Barnett, *A Robust RBF-FD Formulation based on Polyharmonic Splines and Polynomials*, Ph.D. thesis, Department of Applied Mathematics, University of Colorado at Boulder, Boulder, 2015.

[3] V. Bayona, *An insight into RBF-FD approximations augmented with polynomials*, Comput. Math. Appl., 77 (2019), pp. 2337–2353, https://doi.org//10.1016/j.camwa.2018.12.029.

[4] 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), pp. 378–399, https://doi.org/10.1016/j.jcp.2018.12.013.

[5] 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), pp. 257–273, https://doi.org/10.1016/j.jcp.2016.12.008.

[6] A. Boulkhemair and A. Chakib, *On the uniform Poincaré inequality*, Comm. Partial Differential Equations, 32 (2007), pp. 1439–1447, https://doi.org/10.1080/03605300600910241.

[7] L. Brandolini, L. Colzani, G. Gigante, and G. Travaglini, *A Koksma-Hlawka inequality for simplices*, in Trends in Harmonic Analysis, Springer INdAM Ser. 3, Springer, New York, 2013, pp. 33–46, https://doi.org/10.1007/978-88-470-2853-1_3.

[8] L. Brandolini, L. Colzani, G. Gigante, and G. Travaglini, *On the Koksma-Hlawka inequality*, J. Complexity, 29 (2013), pp. 158–172, https://doi.org/10.1016/j.jco.2012.10.003.

[9] S. C. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods*, 3rd ed., Texts in Appl. Math. 15, Springer, New York, 2008, https://doi.org/10.1007/978-0-387-75934-0.

[10] O. Davydov, *Error Bounds for a Least Squares Meshless Finite Difference Method on Closed Manifolds*, https://arxiv.org/abs/1910.03359, 2019.

[11] 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), pp. 39–62, https://doi.org/10.1016/j.jcp.2016.02.078.

[12] 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), pp. 21–38, https://doi.org/10.1016/j.jcp.2016.05.026.

[13] B. Fornberg and N. Flyer, *Fast generation of* 2*-D node distributions for mesh-free PDE discretizations*, Comput. Math. Appl., 69 (2015), pp. 531–544, https://doi.org/10.1016/j.camwa.2015.01.009.

[14] B. Fornberg, N. Flyer, and J. M. Russell, *Comparisons between pseudospectral and radial basis function derivative approximations*, IMA J. Numer. Anal., 30 (2010), pp. 149–172, https://doi.org/10.1093/imanum/drn064.

[15] E. Larsson, V. Shcherbakov, and A. Heryudono, *A least squares radial basis function partition of unity method for solving PDEs*, SIAM J. Sci. Comput., 39 (2017), pp. A2538–A2563, https://doi.org/10.1137/17M1118087.

[16] S. Larsson and V. Thomée, *Partial Differential Equations with Numerical Methods*, Texts Appl. Math. 45, Springer-Verlag, Berlin, 2003.

[17] G. R. Liu, B. B. T. Kee, and L. Chun, *A stabilized least-squares radial point collocation method (LS-RPCM) for adaptive analysis*, Comput. Methods Appl. Mech. Engrg., 195 (2006), pp. 4843–4861, https://doi.org/10.1016/j.cma.2005.11.015.

[18] C. A. Micchelli, *Interpolation of scattered data: Distance matrices and conditionally positive definite functions*, Constr. Approx., 2 (1986), pp. 11–22, https://doi.org/10.1007/BF01893414.

[19] S. Milovanović and L. von Sydow, *A high order method for pricing of financial derivatives using radial basis function generated finite differences*, Math. Comput. Simulation, 174 (2020), pp. 205–217, https://doi.org/10.1016/j.matcom.2020.02.005.

[20] D. Mirzaei, *The direct radial basis function partition of unity (D-RBF-PU) method for solving PDEs*, SIAM J. Sci. Comput., 43 (2021), pp. A54–A83, https://doi.org/10.1137/19M128911X.

[21] P.-O. Persson and G. Strang, *A simple mesh generator in Matlab*, SIAM Rev., 46 (2004), pp. 329–345, https://doi.org/10.1137/S0036144503429121.

[22] A. Petras, L. Ling, C. Piret, and S. J. Ruuth, *A least-squares implicit RBF-FD closest point method and applications to PDEs on moving surfaces*, J. Comput. Phys., 381 (2019), pp. 146–161, https://doi.org/10.1016/j.jcp.2018.12.031.

[23] R. B. Platte and T. A. Driscoll, *Eigenvalue stability of radial basis function discretizations for time-dependent problems*, Comput. Math. Appl., 51 (2006), pp. 1251–1268, https://doi.org/10.1016/j.camwa.2006.04.007.

[24] R. Schaback, *All well-posed problems have uniformly stable and convergent discretizations*, Numer. Math., 132 (2016), pp. 597–630, https://doi.org/10.1007/s00211-015-0731-8.

[25] I. J. Schoenberg, *Metric spaces and completely monotone functions*, Ann. of Math. (2), 39 (1938), pp. 811–841, https://doi.org/10.2307/1968466.

[26] V. Shankar, *The overlapped radial basis function-finite difference (RBF-FD) method: A generalization of RBF-FD*, J. Comput. Phys., 342 (2017), pp. 211–228, https://doi.org/10.1016/j.jcp.2017.04.037.

[27] V. Shankar, R. M. Kirby, and A. L. Fogelson, *Robust node generation for mesh-free discretizations on irregular domains and surfaces*, SIAM J. Sci. Comput., 40 (2018), pp. A2584–A2608, https://doi.org/10.1137/17M114090X.

[28] C. Shu, H. Ding, and K. Yeo, *Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier–Stokes equations*, Comput. Methods Appl. Mech. Engrg., 192 (2003), pp. 941–954, https://doi.org/10.1016/S0045-7825(02)00618-7.

[29] J. Slak and G. Kosec, *On generation of node distributions for meshless PDE discretizations*, SIAM J. Sci. Comput., 41 (2019), pp. A3202–A3229, https://doi.org/10.1137/18M1231456.

[30] A. I. Tolstykh, *On using RBF-based differencing formulas for unstructured and mixed structured-unstructured grid calculations*, in Proceedings of the 16th IMACS World Congress on Scientific Computation, Applied Mathematics and Simulation, Lausanne, Switzerland, 2002.

[31] I. Tominec, *Rectangular and Square RBF-FD Matrices in MATLAB*, https://github.com/IgorTo/rbf-fd, 2021, https://doi.org/10.5281/zenodo.4525551.

[32] K. van der Sande and B. Fornberg, *Fast Variable Density 3-D Node Generation*, SIAM J. Sci. Comput., 43 (2021), pp. A242–A257, https://doi.org/10.1137/20M1337016.

[33] H. Wendland, *Fast evaluation of radial basis functions: Methods based on partition of unity*, in Approximation Theory, X (St. Louis, MO, 2001), Innov. Appl. Math., Vanderbilt University Press, Nashville, TN, 2002, pp. 473–483.

[34] H. Wendland, *Scattered Data Approximation*, Cambridge Monogr. Appl. Comput. Math. 17, Cambridge University Press, Cambridge, UK, 2005.

[35] G. B. Wright and B. Fornberg, *Scattered node compact finite difference-type formulas generated from radial basis functions*, J. Comput. Phys., 212 (2006), pp. 99–123, https://doi.org/10.1016/j.jcp.2005.05.030.