An unfitted RBF-FD method in a least-squares setting for elliptic PDEs on complex geometries

Igor Tominec [a,*], Eva Breznik [b]

[a] *Uppsala University, Department of Information Technology, Division of Scientific Computing, Sweden*
[b] *Uppsala University, Department of Information Technology, Centre for Image Analysis, Sweden*

A R T I C L E   I N F O

A B S T R A C T

Radial basis function generated finite difference (RBF-FD) methods for PDEs require a set of interpolation points which conform to the computational domain Ω. One of the requirements leading to approximation robustness is to place the interpolation points with a locally uniform distance around the boundary of Ω. However generating interpolation points with such properties is a cumbersome problem. Instead, the interpolation points can be extended over the boundary and as such completely decoupled from the shape of Ω. In this paper we present a modification to the least-squares RBF-FD method which allows the interpolation points to be placed in a box that encapsulates Ω. This way, the node placement over a complex domain in 2D and 3D is greatly simplified. Numerical experiments on solving an elliptic model PDE over complex 2D geometries show that our approach is robust. Furthermore it performs better in terms of the approximation error and the runtime vs. error compared with the classic RBF-FD methods. It is also possible to use our approach in 3D, which we indicate by providing convergence results of a solution over a thoracic diaphragm.

© 2021 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

Most localized radial basis function (RBF) methods for computing solutions to partial differential equations (PDEs) (for example [1–3]) require a set of interpolation points that conforms to a computational domain Ω. Normally, a localized RBF method uses a collection of local interpolation problems over the subsets of interpolation points (e.g. stencils or patches) to generate the compactly supported cardinal functions which are then employed to solve a PDE. It is well known that locally non-uniform node distances between interpolation points increase the conditioning of the local interpolation problem and cause unwanted growth of the cardinal functions [4,2]. While it is trivial to place uniform points in the very interior of Ω, it is on the other hand challenging to place them in the vicinity of a boundary of Ω and at the same time maintain their uniformity. This is especially difficult in three dimensions. One possibility to avoid these difficulties is to improve the node generation techniques [5–8]. Another possibility to circumvent the aforementioned difficulties is to build a PDE discretization over a set of interpolation points decoupled from the shape of Ω (see Fig. 1): this is what we focus on in this work.

The first steps towards solving PDEs using a computational grid decoupled from the computational domain were made by Peskin in 1972 [9,10], who introduced the *Immersed boundary method* (IB) in order to simplify the simulations of fluid-
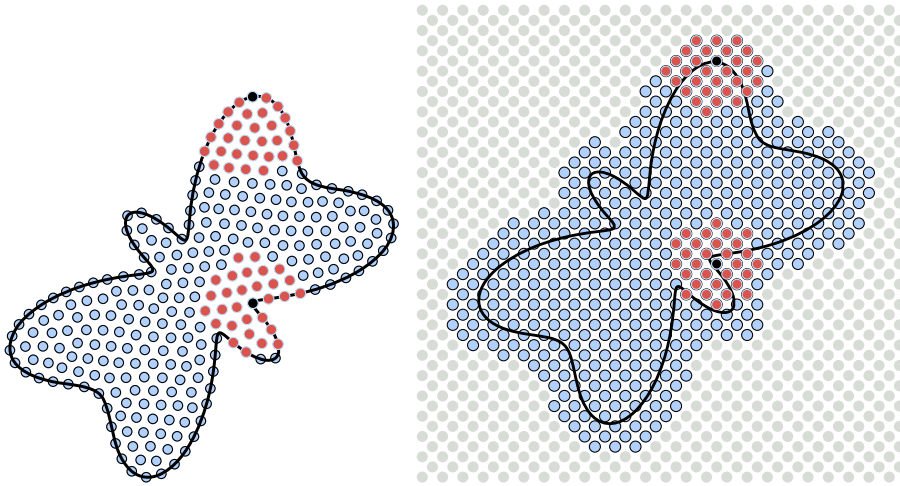

Fig. 1. Node distributions over a butterfly. Left: a classical distribution of interpolation points over Ω for RBF-FD in the least-squares setting with two skewed stencils on the boundary. Here the black point is the stencil center and the red points are the members of the stencil. Right: a node distribution over Ω for the unfitted RBF-FD method in the least-squares setting with a less skewed stencil on the boundary. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

structure interactions (FSI). The method uses a fixed, structured Eulerian grid to move the flow, and a Lagrangian grid to move a curvilinear immersed boundary. The two grids are coupled through a special forcing term that is added to the continuous PDE. A difficulty in IB is that the FSI solution quantities (pressure and stress) become discontinuous at the interface of the fluid and the solid, leading to low-order accuracy. A remedy is the *Immersed interface method* [11], which directly imposes the jump condition over the interface, and by that enables high-order accuracy. The Cartesian cut cell methods [12,13] also belong to the related work, where a finite difference method (FD) is employed on a background Cartesian grid, cut by an arbitrarily shaped immersed boundary. The interior cells are treated as in classical FD, while the cut cells get a different treatment, depending on the application.

The so called *unfitted methods* [14–16] which are a part of the finite element methods address the decoupling of Ω and a mesh: the boundary of Ω is enclosed in a box with a background mesh, where only the elements which have a non-empty intersection with Ω are taken to be active. Those methods tend to suffer from ill-conditioning in the presence of small and irregular cuts close to the boundary of Ω. This was for example addressed in [17–19]. An additional challenge is the enforcement of Dirichlet boundary conditions, which has been addressed by introducing a penalty term over the boundary elements [20,21].

Another related approach is the placement of *ghost points* (also *fictitious points*) in finite difference methods, where additional points (unknowns) are placed outside of Ω in order to enforce Neumann type boundary conditions in a more accurate way. This is a widely used concept, an example can be found in [22], where for every added point, an additional equation is generated in order to maintain a square linear system of equations. In [23,24] the authors introduce ghost points for a global radial basis function (RBF) collocation method in order to decouple interpolation points from Ω. Their computational study shows that this is a feasible approach and that the error under node refinement tends to be smaller compared with the fitted method. However the study is limited to using basis functions with a global support, and the study does not provide an insight into how many ghost points to use and how that affects the stability properties.

A partition of unity based RBF method in a least-squares setting (RBF-PUM-LS) [25] enables a decoupling of the interpolation points from Ω by placing a set of overlapping patches over Ω, where every patch contains interpolation points independent of the shape of Ω. The authors provided numerical evidence that RBF-PUM-LS is an accurate and robust method to solve an elliptic model problem, but have not studied the effects of the patches that extend outside Ω.

A recently introduced *RBF-FD method in a least-squares setting* (RBF-FD-LS) [26] was proven to be significantly more robust compared to the same method in the collocation setting (RBF-FD-C), especially in the presence of Neumann-type boundary conditions [26]. However the interpolation points are required to conform to Ω. Another study leading to a least-squares RBF-FD was introduced in [27].

In this paper we use the RBF-FD-LS method [26] and for that method introduce an approach to decouple interpolation nodes from Ω. A computational domain Ω is enclosed in a box that contains a set of regularly spaced interpolation nodes with reasonably good interpolation properties (see Fig. 1 and Fig. 2). The solution unknowns are determined by solving a system where every equation is an evaluation of a PDE in a point $y \in \bar{\Omega}$. Here every y picks the closest interpolation point which is used as a center of an approximation stencil built over a set of neighboring interpolation points in a box. The shape of such a stencil is given in Fig. 1 and Fig. 2. An additional strength of the present approach is that the approximation stencils around the boundary become less skewed which is found to reduce the magnitude of the error around the boundary, especially when the stencil size is large.
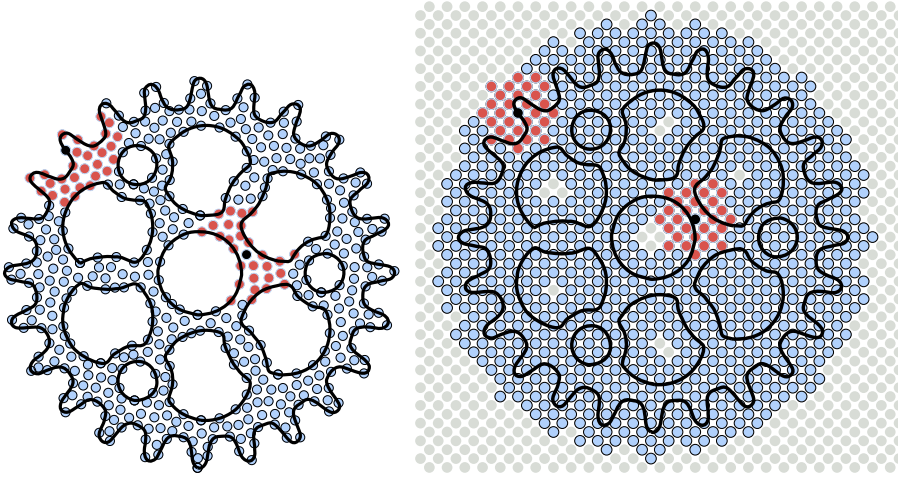
Fig. 2. Node distributions over a drilled 24-tooth sprocket. Left: a classical distribution of interpolation points over Ω for RBF-FD in the least-squares setting with two skewed stencils on the boundary. Here the black point is the stencil center and the red points are the members of the stencil. Right: a node distribution over Ω for the unfitted RBF-FD method in the least-squares setting with a less skewed stencil on the boundary.

The paper is organized as follows. In Section 2 we state the model problem. In Section 3 we provide a description of the unfitted RBF-FD-LS method, together with the formulas for computing local differentiation weights, the global differentiation matrices and the discretization of the model problem. In Section 4 we study linear independence of the cardinal functions as the interpolation points move away from the boundary of Ω and develop a heuristic criterion to keep the linear independence of cardinal functions unchanged. This is a necessary condition for the well-posedness of the discrete PDE problem. In Section 5 we study the relation between the discrete solution and the analytic solution by deriving a discrete error estimate without an a-priori bound on the stability norm. The 2D experiments are presented in Sections 6 and 7. In the former section we consider a butterfly domain and numerically investigate the behaviors of the error against true solution, stability norm and the condition numbers under node and polynomial degree refinements. The results are compared against RBF-FD-LS and RBF-FD-C. In the latter we use a drilled 24-tooth sprocket as a computational domain and study the effects of the unfitted discretization on the spatial distribution of the error for a fixed internodal distance and several polynomial degrees. In Section 8 we numerically study the convergence under node refinement in 3D, where we use a thoracic diaphragm geometry extracted from medical images. Lastly, Section 9 concludes the paper and offers directions for further work.

2. The model problem

We choose to evaluate our method by solving the Poisson equation on an open and bounded domain $\Omega \subset \mathbb{R}^d$ ($d = 2, 3$) with mixed boundary conditions.

$$\begin{aligned}
\Delta u(y) &= f_2(y), \quad y \in \Omega, \\
u(y) &= f_0(y), \quad y \in \partial\Omega_0, \\
\nabla u(y) \cdot n &= f_1(y), \quad y \in \partial\Omega_1,
\end{aligned} \tag{1}$$

where $\partial\Omega_0$ and $\partial\Omega_1$ are two disjoint parts of a smooth boundary $\partial\Omega$. The solution u is throughout the paper assumed to be smooth. In the theoretical parts of the paper we prefer to work with the following formulation of the same problem:

$$Du(y) = F(y), \tag{2}$$

where:

$$Du(y) = \begin{cases} \Delta u(y), & y \in \Omega, \\ u(y), & y \in \partial\Omega_0, \\ \nabla u(y) \cdot n(y), & y \in \partial\Omega_1, \end{cases} \quad \text{and} \quad F(y) = \begin{cases} f_2(y), & y \in \Omega, \\ f_0(y), & y \in \partial\Omega_0, \\ f_1(y), & y \in \partial\Omega_1 \end{cases} \tag{3}$$

The numerical solution is going to be sought using:

$$u_h(y) = \sum_{i=1}^{N} u_h(x_i) \Psi_i(y), \tag{4}$$

where $\Psi_i(y)$, $i = 1, .., N$ are the RBF-FD cardinal functions and $u_h(x_i)$ are the nodal values of the solution.

3. The unfitted RBF-FD method

In this section we discuss the choice of point sets that discretize the computational domain, the generation of the cardinal functions (4) using the RBF-FD method, the assembly of the evaluation and differentiation matrices and the discretization of the model problem (1).

3.1. The point sets

Two sets of computational points are distributed over Ω:

- *The interpolation point set* $X = \{x_i\}_{i=1}^N$ *for generating the cardinal functions.*
- *The evaluation point set* $Y = \{y_i\}_{j=1}^M$ *for sampling the PDE* (1).

We noted in [26] that the X points are supposed to be distributed such that the internodal distance is as uniform as possible, since the Lebesgue constants associated with the cardinal functions then stay fairly small. On the other hand the evaluation point set Y does not influence the magnitude of the Lebesgue constants but is instead important for the implicit integration that occurs when solving a discretized system of equations in the least-squares sense [26]. Thus the constraints for placing the Y points are far more forgiving: as long as Ω is overall well covered with Y points, some of them can lie very close to each other, also in the vicinity of the boundary.

We choose the interpolation point set such that it does not conform to the computational domain Ω (see Fig. 1 and Fig. 2). Throughout the paper we take X to be a tilted Cartesian grid with spacing h for 2D cases. For the 3D case we use a point set obtained using the algorithm from [28]. The motivation for that is a simplified point generation, and the benefit of a polynomial unisolvency on those points. The latter is important for the well-posedness of an interpolation problem over a stencil.

The evaluation point set Y conforms to Ω. We can subdivide Y into the points on the boundary and the points in the interior of the domain. We choose the interior points such that there are q points placed in every Voronoi cell centered around each $x \in X$. Here q is the oversampling parameter. The boundary points are then placed on $\partial\Omega$ with a uniform distance that corresponds to the distance between the interior points. For a visual representation see Fig. 3. With such a relation between the X- and the Y-points it follows that the cardinality of those sets very closely matches the relation: $M \approx qN$, where M is the number of Y-points and N is the number of X-points. In [26] we studied the effect of q on the stability properties of the oversampled discretization and to the accuracy of the final approximation. We found out that in 2D, the stability and accuracy improved until $q = 3$, but the improvement after that point was not significant anymore. In our experience $q > 3$ was always a good choice of the oversampling parameter for solving the steady-state PDEs in 2D. In 3D, the oversampling can be determined by the relation $q_{3D} \approx q_{2D}\sqrt{q_{2D}}$, based on a reasonable assumption that sampling with q_{1D} points in $1D$ is equivalent to sampling with $q_{2D} = q_{1D}^2$ points in $2D$, which is in $3D$ equivalent to sampling with $q_{3D} = q_{1D}^3 = q_{1D} q_{1D}^2 = \sqrt{q_{2D}} \, q_{2D}$ points.

Note that in Fig. 3 the points placed into every Voronoi region are obtained using the algorithm introduced in [7]. It is also possible to avoid placing q points into every Voronoi region by using a global point set Y that conforms to Ω, which we tested in [26]. The only requirement of the global Y point set is that every Voronoi cell gets a reasonable sampling, with approximately q points. Halton point layout would be a good candidate for forming a global Y. In this paper, however, we use the Voronoi cell point placement technique.

3.2. The RBF-FD trial space

Let Ω_i be a subdomain holding a collection of points $X_{\Omega_i} = \{x_j\}_{j=1}^n$ that are a subset of the interpolation points placed on top of the computational domain Ω. Then every stencil is defined by a tuple (Ω_i, x_i), where $x_i \in X_{\Omega_i}$ is the stencil center point (see Fig. 1 for a visual representation of a stencil). The solution $u_h(x)$ over a stencil is spanned by a combination of cubic polyharmonic splines $\phi_l(x) = ||x - x_l||^3$ and multivariate monomials $\{\bar{p}_k\}_{k=1}^m$ of degree D_m. The relation between m and D_m is: $m = \binom{D_m + d}{d}$. The linear combination then reads:

$$u_h(x) = \sum_{l=1}^n c_l \phi_l(x) + \sum_{k=1}^m \beta_k \bar{p}_k(x), \tag{5}$$

$$\text{subject to} \sum_{l=1}^n c_l \bar{p}_k(x_l) = 0, \quad k = 1, .., m,$$

where n is the stencil size, c_l are the interpolation coefficients and β_k are the Lagrange multipliers. The approach with appending the monomial basis to the polyharmonic spline basis was studied in [29–31]. The coefficients c_l and β_k from (5)
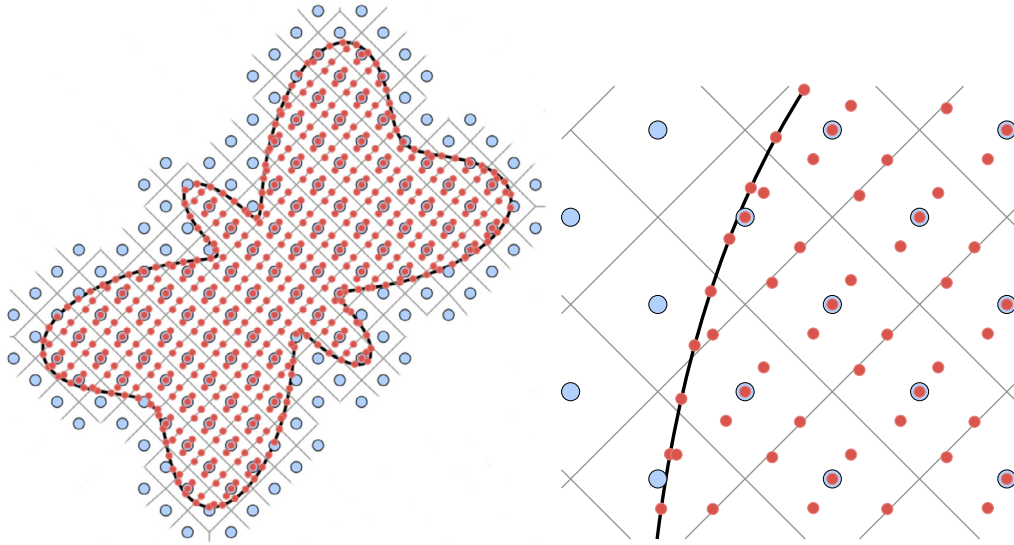
Fig. 3. The image on the left displays a butterfly domain with evaluation points (smaller red markers). Evaluation points are formed such that local templates of $q = 5$ points are placed in Voronoi cells (grey lines) centered around every interpolation point (larger blue markers). The image on the right is a closer view over the boundary where $q = 5$ points are placed in every Voronoi cell.

are computed by requiring the interpolation conditions $u_h(X_{\Omega_i}) = u(X_{\Omega_i})$, where $u(X_{\Omega_i}) := \underline{u}^{(i)}$ are the stencil data (stencil nodal values). This results in a square system of equations:

$$\underbrace{\begin{pmatrix} A^{(i)} & P^{(i)} \\ (P^{(i)})^T & 0 \end{pmatrix}}_{:=\tilde{A}^{(i)}} \begin{pmatrix} \underline{c}^{(i)} \\ \underline{\beta}^{(i)} \end{pmatrix} = \begin{pmatrix} \underline{u}^{(i)} \\ 0 \end{pmatrix}. \tag{6}$$

Here $A_{jl}^{(i)} = \phi_l(x_j)$ for indices $j, l = 1, .., n$ and $P_{jk}^{(i)} = \bar{p}_k(x_j)$ for index $k = 1, .., m$ and $u_j^{(i)} = u^{(i)}(x_j)$.

The stencil-based solution in any point $z \in \Omega_i$ is expressed by reusing the computed coefficients $\underline{c}^{(i)}$ and $\underline{\beta}^{(i)}$ from (6) within the linear combination (5):

$$u_h^{(i)}(z) = \left(\underbrace{\left(\phi_1(z), .., \phi_n(z), \quad \bar{p}_1(z), .., \bar{p}_m(z) \right)}_{:=b^{(i)}(z)} \begin{pmatrix} \underline{c}^{(i)} \\ \underline{\beta}^{(i)} \end{pmatrix} \right)_{1:n}$$

$$= \left(b^{(i)}(z)((\tilde{A}^{(i)})^{-1}) \right)_{1:n} \underline{u}^{(i)} \equiv (\psi_1^{(i)}(z) \cdots \psi_n^{(i)}(z)) \underline{u}^{(i)} \equiv \underline{w}^{(i)}(z) \underline{u}^{(i)}, \tag{7}$$

where $\psi_1^{(i)}(z), .., \psi_n^{(i)}(z)$ are the stencil-based cardinal functions with the Kronecker delta property ($\psi_k^{(i)}(z) = 1$ if $z = x_k$, otherwise 0) and $\underline{w}^{(i)}(z)$ are the local stencil weights for evaluating the stencil-based solution at a point z.

The next step is to use the formulation from (7) to represent the solution over the whole computational domain Ω. First, the inverse $(\tilde{A}^{(i)})^{-1}$ from (6) is computed for every stencil (Ω_i, x_i). Among all (N) available stencils, we now associate every evaluation point $y \in Y$ with an index of the closest stencil center point defined as:

$$\rho(y) = \arg\min_i \|y - x_i\|, \quad i = 1, .., N. \tag{8}$$

By construction, we have that $y \in \Omega_{\rho(y)}$. Thus, we can use (7) and employ the stencil selection criterion (8) to express the local solution at every $y \in Y$:

$$u_h^{\rho(y)}(y) = \underline{w}^{\rho(y)}(y) \underline{u}^{\rho(y)} = \sum_{j=1}^n w_j^{\rho(y)}(y) u^{\rho(y)}(x_j), \quad x_j \in X_{\Omega_{\rho(y)}}. \tag{9}$$

The local solution is using local indexing over the vector of weights and over the vector of the stencil nodal values. To solve a PDE, we need a global representation of the solution, that is, a linear combination of the global nodal values and the global weights (global cardinal functions). Using (9), the global solution at any $y \in \Omega$ is formally written as:

$$u_h(y) = \sum_{i=1}^{N} u(x_i)\Psi_i(y) = \sum_{i=1}^{N} u(x_i) \sum_{j=1}^{n} w_j^{\rho(y)}(y)\,\delta_{i,\Gamma(j,\rho(y))}$$

$$= \sum_{i=1}^{N} u(x_i) \sum_{j=1}^{n} \left[b^{\rho(y)}(y)\tilde{A}_{\rho(y)}^{-1} \right]_j \delta_{i,\Gamma(j,\rho(y))}, \qquad (10)$$

where Ψ_i are the global cardinal functions, $w_j^{\rho(y)}$ are the local weights over the stencil centered at $x_{\rho(y)}$. Furthermore the operator $\Gamma(j,\rho(y)) : \mathcal{I}[1,n] \to \mathcal{I}[1,N]$ is an index mapping from the $j-th$ local weight of a stencil with index $\rho(y)$ to its global equivalent. Here the sum $\sum_{j=1}^{n} w_j^{\rho(y)}(y)\,\delta_{i,\Gamma(j,\rho(y))}$ is using the Kroenecker delta function to search for one local weight $w_j^{\rho(y)}(y)$ which is multiplied only with the corresponding global nodal solution $u(x_i)$ inside the linear combination (10). In this way we are mimicking the matrix-vector product of one row of the global evaluation matrix and the vector of weights generated specifically for y. This ansatz could be used for theoretical studies of the RBF-FD method in the future. Although the global RBF-FD cardinal functions are discontinuous along the interfaces of the Voronoi regions [26], a derivative of the unknown solution is determined in a unique way: for every given y we choose a unique stencil representation (9) with an index $\rho(y)$ defined in (8). Within that stencil we are always able to smoothly evaluate a derivative. A representation of a differential operator $\mathcal{L}$ in y is then written as:

$$\mathcal{L}u_h(y) = \sum_{i=1}^{N} u(x_i)\mathcal{L}\Psi_i(y)$$

$$= \sum_{i=1}^{N} u(x_i) \sum_{j=1}^{n} \left[\mathcal{L}b^{(i)}(y)\,\tilde{A}_{\rho(y)}^{-1} \right]_j \delta_{i,\Gamma(j,\rho(y))}. \qquad (11)$$

3.3. Evaluation and differentiation matrices

Setting $y = Y$ in equation (10) we arrive at the discrete representation of a solution in the Y-points:

$$u_h(Y) = E_h(Y,X)u_h(X), \qquad (12)$$

where $E_h(Y,X)$ is a rectangular evaluation matrix interpolating $u_h(X)$ from X to Y. Its components are $(E_h)_{ik} = \Psi_i(y_k)$.

The discrete representation of a differential operator $\mathcal{L}$ is obtained by setting $y = Y$ in (11):

$$\mathcal{L}u_h(Y) = D_h^{\mathcal{L}}(Y,X)u_h(X), \qquad (13)$$

where $D_h^{\mathcal{L}}$ is a rectangular differentiation matrix with components $(D_h^{\mathcal{L}})_{ik} = \mathcal{L}\Psi_i(y_k)$. MATLAB code for generating matrices $E_h(Y,X)$ and $D_h^{\mathcal{L}}(Y,X)$ is available in [32].

3.4. The unfitted discretization of a PDE

The model problem (3) is discretized with the RBF-FD operators E_h and $D_h^{\mathcal{L}}$ given in (12) and (13) respectively. The result is the semi-discrete matrix $D_h(y,X)$ and the semi-discrete vector $F(y)$, where:

$$D_h(y,X) = \begin{cases} \beta_2 D_h^{\Delta}(y,X), & y \in \Omega \\ \beta_0 E_h(y,X), & y \in \partial\Omega_0 \\ \beta_1 D_h^{\nabla \cdot n(y)}(y,X), & y \in \partial\Omega_1 \end{cases} \text{ and } F(y) = \begin{cases} \beta_2 f_2(y), & y \in \Omega \\ \beta_0 f_0(y), & y \in \partial\Omega_0 \\ \beta_1 f_1(y), & y \in \partial\Omega_1 \end{cases}$$

$$(14)$$

Here $\beta_2, \beta_0, \beta_1$ are the scalings of the PDE and the boundary conditions. Setting $y = Y$ we obtain a rectangular linear system of size $M \times N$:

$$D_h(Y,X)u_h(X) = F(Y). \qquad (15)$$

For that system choose the scalings:

$$\beta_2 = \frac{1}{\sqrt{M_2}}, \quad \beta_0 = h^{-1}\frac{1}{\sqrt{M_0}}, \quad \beta_1 = \frac{1}{\sqrt{M_1}},$$

where h is the average node distance in the node set X and where M_2, M_0 and M_1 are the number of evaluation points placed over Ω, $\partial\Omega_0$ and $\partial\Omega_1$ respectively.

The motivation to use this scaling is two fold. Firstly, the factors $\frac{1}{\sqrt{M_2}}$, $\frac{1}{\sqrt{M_0}}$ and $\frac{1}{\sqrt{M_1}}$ relate every component of $D_h^T D_h$, which are discrete inner products, to continuous inner products plus a first order integration error [26]. Secondly, the factor h^{-1} is used to impose the Dirichlet condition in a weak sense such that the matrix $D_h^T D_h$ is nonsingular: this is a classical approach in those finite element methods which use a solution space that does not exactly satisfy the Dirichlet condition. In order to prove uniqueness of the solution in such a setup, a mesh dependent parameter h^{-1} has to be introduced via inverse inequalities, which is at the end multiplying the added Dirichlet penalty term [21]. We also note that we are not able to impose a Dirichlet condition exactly in an efficient way due to using X-points that are unfitted with respect to the boundary.

The numerical solution is obtained by solving (15) for the solution coefficients $u_h(X)$ and then interpolating this data onto the evaluation points Y. This can be written as:

$$u_h(Y) = E_h(Y, X) \, D_h^+(Y, X) F(Y), \tag{16}$$

where $D_h^+(Y, X) = (D_h^T D_h)^{-1} D_h^T$ is a pseudoinverse, which, in practice, is computed using the QR decomposition.

Once $u_h(X)$ is computed, the residual is given by:

$$r(Y) = D_h(Y, X) u_h(X) - F(Y). \tag{17}$$

The least-squares residual is by definition orthogonal to the column-space of $D_h(Y, X)$, which implies the relation:

$$D_h^T r(Y) = 0 \quad \Rightarrow \quad D_h^+ r(Y) = (D_h^T D_h)^{-1} D_h^T r(Y) = 0. \tag{18}$$

Throughout this paper we solve (16) using the *mldivide()* function in Matlab, which uses a sparse QR decomposition as an intermediate step in order to obtain the solution to the rectangular system. In our experience this was a reasonably fast approach when the number of unknowns was – roughly speaking – smaller or equal to 10^5. For problems with larger amounts of unknowns, the reader could use a conjugate-gradient method on the normal system $D_h^T D_h = D_h^T F$. Unfortunately this approach is prone to numerical instabilities when D_h is ill-conditioned, since the already large condition number is then squared: $\kappa(D_h^T D_h) = \kappa(D_h)^2$. However, there exists a numerically more stable algorithm, called *lsqr* [33], which is a conjugate-gradient like iterative solver designed for solving large rectangular systems of equations.

4. Linear independence of cardinal functions

In this section we address the difficulties related to the linear independence of the cardinal functions which arise when using the unfitted discretization. We provide a criterion upon which a certain amount of the interpolation points that extend outside of the computational domain is removed. A similar study, but in a context of the isogeometric finite element method is performed in [19].

Throughout Section 3 we outlined that we are using a solution ansatz (10) plugged into the PDE problem (3) to then solve the discretized PDE problem (15) for the unknown nodal values $u(x_i)$, $i = 1, .., N$. It is not possible to solve (15) for these nodal values, unless the cardinal functions $\Psi_i(y)$, $i = 1, .., N$ in (10) form a basis for $\mathbb{R}^N$, i.e. unless they are linearly independent with each other. Since the columns of the matrix E_h from (12) contain the whole set of sampled cardinal functions, we investigate the linear independence of those columns by computing the smallest singular value $\sigma_{\min}(E_h)$. When $\sigma_{\min}(E_h) = 0$ we have that the columns have a nonzero nullspace and thus the sampled family of cardinal functions is linearly dependent. In this section we do not focus on studying $\sigma_{\min}(D_h)$, as this approach would in addition carry information on the well-posedness of the PDE problem. For example, if we had observed $\sigma_{\min}(D_h) = 0$, then this could well be due to using an unfitted discretization, but also other reasons such as: ill-posedness of the PDE problem, wrong imposition of boundary conditions, etc.

Whereas the RBF-FD cardinal functions in the fitted setting are indeed linearly independent when the interpolation points conform to Ω and the interpolation points X are a subset of the evaluation points Y [26], it is important to check whether this is true in the unfitted case as well.

A cardinal function $\Psi_k(y)$ has a Kronecker delta property in the X-points:

$$\Psi_k(y) = \begin{cases} 1, & y = x_k \subset X \\ 0 & y \in X \setminus x_k, \end{cases}$$

which guarantees linear independence as long as $X \subseteq Y$ and $X \subseteq \Omega$ since in this case, there is always at least one point in Ω for every cardinal function (e.g. x_k for Ψ_k) where Ψ_k is one, but all other Ψ_j for indices $k \neq j$ are 0. A problem when using the unfitted discretization can occur due to the compact support of Ψ_k. When an external x_k is placed such that its corresponding Ψ_k vanishes before it reaches the interior of Ω, then $\Psi_k(y) = 0$ for every $y \in \Omega$ and the basis function becomes linearly dependent (on Ω) with all others. Note that when $\Psi_k(y) = 0$ for every $y \in \Omega$, we also have that all derivatives of $\Psi_k(y)$ are 0 for every $y \in \Omega$. Thus the k-th column of the PDE matrix D_h, which is built upon a combination of Ψ_k and its derivatives, is a 0 vector: this implies that D_h in this case does not have a full column rank.

In Fig. 4 we can see a one-dimensional setup, where in the top plot, the X-points are placed outside of Ω such that the left-most cardinal function Ψ_{left} does not have a support (red ellipse) in Ω, which results in a singular E_h ($\sigma_{\min} = 0$). As the
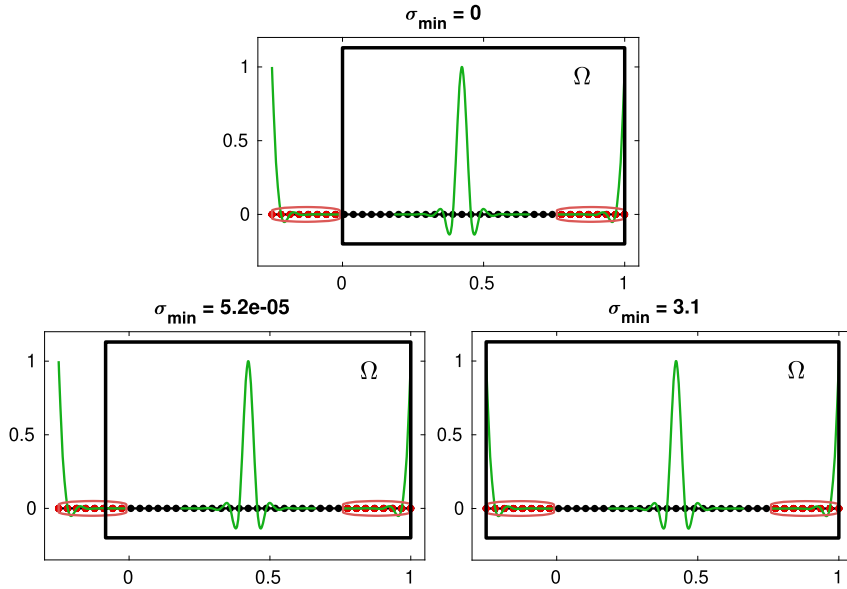
Fig. 4. Setup for measuring the smallest singular value of the evaluation matrix E_h as the size of the computational domain Ω grows. The green lines represent three cardinal functions. The encircled red points represent the support of the two outmost cardinal functions.
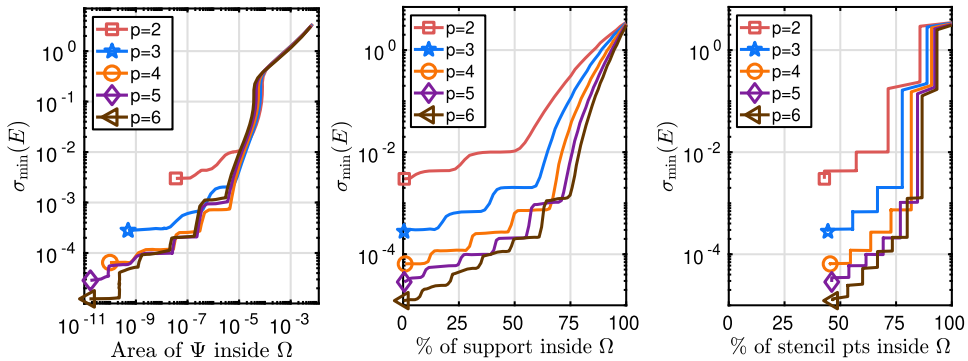

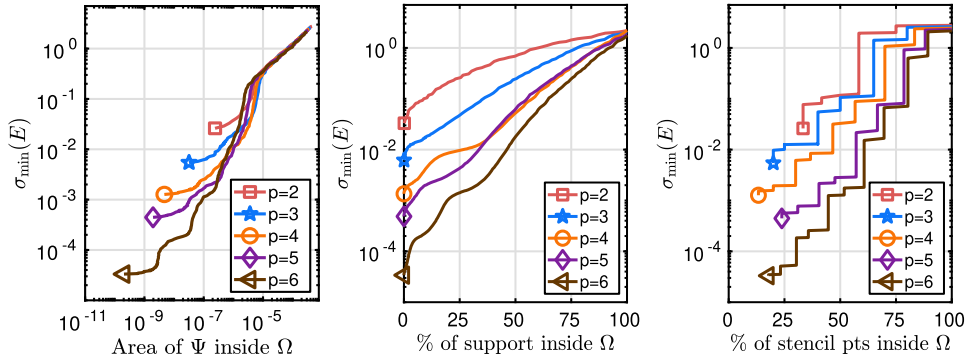**Fig. 5.** One-dimensional case: The smallest singular value as a function of (i) the approximate area of the outmost cardinal function Ψ_{outmost} that penetrates inside Ω, (ii) the support of Ψ_{outmost} inside Ω, (iii) the percentage of stencil points inside Ω, which are a part of the outmost stencil.

support of Ψ_{left} enters Ω, then E_h becomes non-singular ($\sigma_{\mathrm{min}} = 5.2 \cdot 10^{-5}$), and when the support is fully contained inside Ω then the smallest singular value gets considerably larger ($\sigma_{\mathrm{min}} = 3.1$).

The same setup is used in Fig. 5, where for different polynomial degrees p, $\sigma_{\mathrm{min}}(E_h)$ is computed as a function of:

- the approximate area under Ψ_{left} inside Ω,
- the percentage of the compact support of Ψ_{left} inside Ω,
- the percentage of stencil points of the left-most stencil inside Ω.

From Fig. 5 we can see that as long as the percentage of support inside Ω is larger than 0, σ_{min} is also larger than 0. When the percentage of support is gradually increased σ_{min} is also increased, approximately in the same way as the area of Ψ_{left} inside Ω. In the third plot we can see that when the percentage of the stencil points inside Ω is more than 50%, E_h is always non-singular given that the cardinal functions (columns of E_h) are well sampled. We note that any cardinal function Ψ_k centered in x_k is not generated by a single stencil, but by several stencils which have x_k as a neighboring point. In this sense the compact support of Ψ_k is decoupled from the support of one stencil centered around x_k. This is the reason why the measurements of σ_{min} do not coincide when considered as a function of a compact support inside Ω and stencil points inside Ω.

Fig. 6 shows analogous results to Fig. 5 but for a two-dimensional case, where the computational domain is a square of size $[-1, 1] \times [-1, 1]$. Similar results can be observed as for the one-dimensional case in terms of the area under Ψ_{left} and the percentage of its support inside Ω, however, the percentage of stencil points inside Ω is allowed to be smaller in the

Fig. 6. Two-dimensional case: The smallest singular value as a function of (i) the approximate area of the outmost cardinal function Ψ_{outmost} that penetrates inside Ω, (ii) the support of Ψ_{outmost} inside Ω, (iii) the percentage of stencil points inside Ω, which are a part of the outmost stencil.

2-dimensional case. This indicates that the relation between the support of a stencil and the compact support of a cardinal function is in this case tighter.

In our experience the smallest percentage of stencil points inside Ω is the criterion which is the easiest to implement prior to computing RBF-FD differentiation matrices and it is therefore our choice for all further experiments. We remove some of the initial interpolation points (placed in the box around Ω) such that at least 50% of points of each stencil are contained inside Ω. Once we decide on a stencil size n and the initial X-points and Y-points are computed, the criterion can be used by invoking one command in Matlab.

```
X = X(unique(knnsearch(X,Y,'k',ceil(0.5*n))), :);
```

5. Analysis of the approximation error under node refinement

In this section we develop an understanding of the behavior of the error e between the true solution $u(Y)$ and the approximate solution $u_h(Y)$, both restricted to the evaluation points.

5.1. Preliminaries I: a norm for measuring the error

Our choice of a vector norm that measures the error $e(Y) = u(Y) - u_h(Y)$ is given by:

$$\|e(Y)\|_{\ell_2}^2 = \frac{1}{M} \sum_{j=1}^{M} e(y_i)^2 = \frac{1}{M} \|e(Y)\|_2^2. \tag{19}$$

The ℓ_2 norm is a good choice for the discrete least-squares problems since it is up to $\mathcal{O}(h_y)$ (h_y is the average spacing between the Y-points) equivalent to the L_2 norm which is a natural norm for the continuous least-squares problem [26]. In this sense the stability properties can carry over to the discrete formulation, under some assumptions [26].

An analogous norm for a matrix $A \in \mathbb{R}^{M \times N}$ is:

$$\|A\|_{\ell_2}^2 = \frac{1}{M} \sup_{x \neq 0} \frac{\|Ax\|_2^2}{\|x\|_2^2} = \frac{1}{M} \|A\|_2^2. \tag{20}$$

Furthermore we have the relation $\|e_h(Y)\|_{\ell_2} \leq \|e_h(Y)\|_\infty$ due to:

$$\|e(Y)\|_{\ell_2}^2 = \frac{1}{M} \sum_{j=1}^{M} e(y_i)^2 \leq \frac{1}{M} M \max_i e(y_i)^2 = \|e(Y)\|_\infty^2 \tag{21}$$

which is later used to bound the consistency terms.

5.2. Preliminaries II: consistency estimates

Evaluate a function $\hat{u} : \hat{\Omega} \to \mathbb{R}$ in the X-points and denote this data by $\hat{u}(X)$. Then we can, for any $y \in \hat{\Omega}$, where $\hat{\Omega}$ contains Ω, construct a finite dimensional representation $u_{\text{ext}}(y)$ that interpolates the data $\hat{u}(X)$:

$$u_{\text{ext}}(y) = \sum_{i=1}^{N} \hat{u}(x_i) \Psi_i(y), \tag{22}$$

where $\Psi_i(y)$ are the RBF-FD cardinal functions defined in (10). The equivalent matrix-vector formulation is:

$$u_{\text{ext}}(y) = E_h(y, X)\hat{u}(X). \tag{23}$$

The interpolation error is estimated by [30,26]:

$$\|\hat{u}(y) - u_{\text{ext}}(y)\|_\infty \le C_I h^{p+1} |\hat{u}|_{W_\infty^{p+1}}, \tag{24}$$

where and $|\hat{u}|_{W_\infty^{p+1}}$ is a Sobolev semi-norm defined as $|\hat{u}|_{W_\infty^{p+1}} = \max_{y \in \hat{\Omega}} |\mathcal{D}^{p+1}\hat{u}(y)|$, where $\mathcal{D}^{p+1}$ is a partial derivative of degree $p+1$. The constant C_I depends on p and the quality of X point set $c_q = \frac{h}{h_s}$, where h is the fill distance in the point set X and h_s is the separation distance in the point set X. The two distances are defined as:

$$h = \sup_{x \in \Omega} \min_{x_j \in X} \|x - x_j\|_2 \le h_s = \frac{1}{2} \min_{j \ne k, x_j, x_k \in X} \|x_j - x_k\|_2.$$

An application of a PDE operator D from (3) to (22) gives:

$$Du_{\text{ext}}(y) = \sum_{i=1}^N \hat{u}(x_i) D\Psi_i(y).$$

The (semi-discrete) matrix-vector equivalent is:

$$Du_{\text{ext}}(y) = D_h(y, X)\hat{u}(X). \tag{25}$$

There are three parts involved in D, see (3), the Laplacian, the normal derivative and the Dirichlet condition. Each of them has its own consistency estimate depending on the order of the derivative. The overall consistency [26] is bounded by:

$$\begin{aligned}\|D\hat{u}(y) - Du_{\text{ext}}(y)\|_\infty &\le \left(C_2 h^{p-1} + C_1 h^p + C_0 h^{p+1}\right) |\hat{u}|_{W_\infty^{p+1}} \\ &\le C h^{p-1} |\hat{u}|_{W_\infty^{p+1}}.\end{aligned} \tag{26}$$

5.3. Preliminaries III: extension of the true solution

Generally speaking the error estimation for the unfitted RBF-FD requires a special treatment since an intermediate numerical solution $u_h(X)$, see (16), also lives in the exterior of the computational domain. In order to be able to compare the solution with the true solution we have to define an extension of the true solution on some extended open domain. Let $u = u(y)$ be a true unique solution of the PDE problem (1), where $y \in \bar{\Omega}$ (the closure of Ω) and let $\hat{\Omega} = (\bar{\Omega} + \tilde{\Omega}) \subset \mathbb{R}^2$ be an extended domain. The extended smooth solution $\hat{u} = \hat{u}(\hat{y})$, where $\hat{y} \in (\bar{\Omega} + \tilde{\Omega})$ is then defined such that:

$$\hat{u}|_{\bar{\Omega}} = u. \tag{27}$$

This definition makes it possible to bound the error in terms of the stability and consistency terms, where the latter then depends on the size of the partial derivative of degree $p+1$. There exists an extension lemma for smooth functions that we can use, which is stated below.

Lemma 1 *([34, Lemma 4.1]). Suppose M is a smooth manifold with or without boundary, $A \subseteq M$ is a closed subset, and $f : A \to \mathbb{R}^k$ is a smooth function. For any open subset U containing A, there exists a smooth function $\tilde{f} : M \to \mathbb{R}^k$ such that $\tilde{f}|_A = f$ and supp $\tilde{f} \subseteq U$.*

The lemma from above is directly applicable to extending $u : \bar{\Omega} \to \mathbb{R}$, since $\bar{\Omega}$ is a closed domain and since our u is a smooth function. We can therefore conclude that the extension defined in (27) exists, and that a discrete implication is the relation:

$$\hat{u}(Y) = u(Y), \tag{28}$$

where Y is an evaluation point set which conforms to Ω.

5.4. The PDE error estimate

Now we estimate the error $e = u_h(Y) - u(Y)$ between the numerical solution $u_h(Y)$ and the true solution $u(Y)$:

$$\|u(Y) - u_h(Y)\|_{\ell_2} = \|u(Y) - u_h(Y) + u_{\text{ext}}(Y) - u_{\text{ext}}(Y)\|_{\ell_2}$$
$$\leq \|u_h(Y) - u_{\text{ext}}(Y)\|_{\ell_2} + \|u(Y) - u_{\text{ext}}(Y)\|_{\ell_2} \tag{29}$$

where we first added and subtracted $u_{\text{ext}}(Y)$, then used the triangle inequality to make a split into the PDE error $\|u_h(Y) - u_{\text{ext}}(Y)\|_{\ell_2}$ and the interpolation error $\|u(Y) - u_{\text{ext}}(Y)\|_{\ell_2}$. The latter is then measured and trivially bounded by (24).

The term $\|u_h(Y) - u_{\text{ext}}(Y)\|_{\ell_2}$ remains to be estimated. We first use the definition of $u_h(Y)$ from (12) and the definition of $u_{\text{ext}}(Y)$ from (23) and then multiply with $D_h^+ D_h = I$ where D_h defined in (15) is assumed to have a full column rank.

$$\|u_h(Y) - u_{\text{ext}}(Y)\|_{\ell_2} = \|E_h u_h(X) - E_h \hat{u}(X)\|_{\ell_2}$$
$$= \|E_h D_h^+ D_h \left(u_h(X) - \hat{u}(X) \right)\|_{\ell_2}$$
$$= \|E_h D_h^+ \left(D_h u_h(X) - D_h \hat{u}(X) \right)\|_{\ell_2}.$$

Using the relation $D_h u_h(X) = F(Y) + r_h(Y)$ from (17) and the fact that $D_h^+ r_h(Y) = 0$ from (18) we then obtain:

$$\|u_h(Y) - u_{\text{ext}}(Y)\|_{\ell_2} = \|E_h D_h^+ \left(F(Y) + r_h(Y) - D_h \hat{u}(X) \right)\|_{\ell_2}$$
$$= \|E_h D_h^+ \left(F(Y) - D_h \hat{u}(X) \right)\|_{\ell_2}.$$

After that we use that $F(Y) = Du(Y)$ by (2), $u(Y) = \hat{u}(Y)$ by (27), $\|E_h D_h^+\|_{\ell_2} = \frac{1}{\sqrt{M}} \|E_h D_h^+\|_2$ by the matrix norm relation from (20) and use the submultiplicative property of the 2-norm to bound the matrix norm $\|E_h D_h^+\|_2$, and arrive to:

$$\|u_h(Y) - u_{\text{ext}}(Y)\|_{\ell_2} \leq \frac{1}{\sqrt{M}} \|E_h\|_2 \|D_h^+\|_2 \|D\hat{u}(Y) - D_h \hat{u}(X)\|_{\ell_2} \tag{30}$$

It now remains to insert the estimate (30) into (29) and then also combine this with consistency estimates (24), (26) to arrive at the final error estimate:

$$\|u(Y) - u_h(Y)\|_{\ell_2} \leq \frac{1}{\sqrt{M}} \|E_h\|_2 \|D_h^+\|_2 \, C_D \, h^{p-1} \, \max|D^{p+1}\hat{u}| + C_E \, h^{p+1} \, \max|D^{p+1}\hat{u}|. \tag{31}$$

The term $\frac{1}{\sqrt{M}} \|E_h\|_2 \|D_h^+\|_2$ is what we call the stability norm, which should remain constant so that the error overall decays with at least order $p - 1$. We numerically test that in the following section.

6. Detailed numerical experiments on a 2D butterfly domain

In this section we perform computational experiments to further explore the numerical properties of the unfitted RBF-FD method when solving (1) and compare them to the classical RBF-FD method in the least-squares (RBF-FD-LS) and collocation settings (RBF-FD-C). The involved parameters are (the internodal distance of the X-points), p (the polynomial degree used to form the interpolant over a stencil) and q (the oversampling parameter). More precisely, we compute h as the average distance between all pairs of the neighboring interpolation points:

$$h = \frac{1}{N} \sum_{i,j=1}^{N} \min_{x_j \neq x_i} \|x_j - x_i\|_2.$$

The relation between the stencil size n and the polynomial degree p is [31]:

$$n = 2 \binom{p+d}{d}.$$

Throughout the section we compute the relative error $\|e\|$ as:

$$\|e\| = \frac{\|u_h(Y) - u(Y)\|_2}{\|u(Y)\|_2},$$

where $u_h(Y)$ and $u(Y)$ are the numerical and exact solutions sampled in the Y-points.

All of the computations are performed in Matlab on a laptop with an Intel i7-7500U processor and 16 GB of RAM.
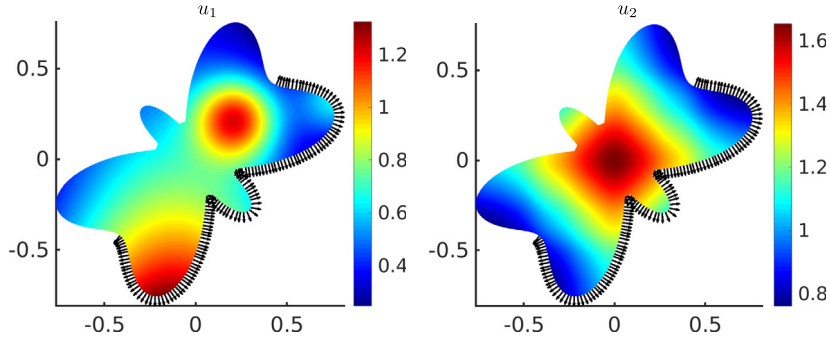
Fig. 7. Solution functions on a butterfly domain. Franke function over a butterfly (u_1), and the Truncated non-analytic function over a butterfly (u_2). The black outward normals indicate the locations of the Neumann condition. The Dirichlet condition is enforced at locations where there are no normals displayed over the boundary.

6.1. Domain Ω

The boundary of a computational domain has a butterfly-like shape (see Fig. 1) and is prescribed using a polar function:

$$r(\theta) = \frac{1}{4}\left(2 + \sin(2t) - 0.01\cos\left(5t - \frac{\pi}{2}\right) + 0.63\sin(6t - 0.1)\right),$$

where r is a radial coordinate and $\theta \in [0, 2\pi]$ the angle.

6.2. Point sets

When using the unfitted RBF-FD-LS method, the point sets X and Y are computed according to the description in Section 3, where for constructing Y we use the oversampling parameter $q = 5$. For a more detailed study of the oversampling parameter and its impact on the approximation error and the stability norm, we refer the reader to [26]. When using the RBF-FD-LS method the point set X is computed using the DistMesh algorithm [35], while the point set Y is computed in the same way as for the unfitted RBF-FD-LS method. For the RBF-FD-C method, the point set X is also computed using the DistMesh algorithm, note that in the collocation case we have $Y = X$.

6.3. Solution functions

We pick two solution functions to compute the right-hand-sides of (1). Those are:

$$u_1(x, y) = \frac{3}{4}e^{-\frac{1}{4}((9x-2)^2+(9y-2)^2)} + \frac{3}{4}e^{-(\frac{1}{49}(9x+1)^2+\frac{1}{10}(9y+1)^2)} \tag{32}$$

$$\ldots + \frac{1}{2}e^{-\frac{1}{4}((9x-7)^2+(9y-3)^2)} - \frac{1}{5}e^{-((9x-4)^2+(9y-7)^2)},$$

$$u_2(x, y) = \sum_{k=0}^{5} e^{-\sqrt{2^k}}\left(\cos(2^k x) + \cos(2^k y)\right), \tag{33}$$

where u_1 is the Franke function, a commonly used infinitely smooth test function for benchmarking multivariate approximations. Function u_2 is a truncated series of an infinitely smooth function that is at the same time not analytic: we refer to u_2 as the Truncated non-analytic function. Both functions over the butterfly domain are displayed in Fig. 7.

6.4. Convergence under node refinement with Dirichlet condition

First we use the Dirichlet boundary condition on all of $\partial\Omega$ and compute the error as a function of $1/h$ for three choices of polynomial degrees: $p = 2$, $p = 4$ and $p = 6$. The results for the Franke and Truncated non-analytic functions are given in Fig. 8 and Fig. 9, respectively. We observe that for the Franke function, the error behavior is similar for all three methods, while for the Truncated non-analytic function the error is smaller for the unfitted RBF-FD method when the stencil sizes are larger ($p = 4$ and $p = 6$). We note that the Dirichlet condition is enforced exactly in RBF-FD-C and RBF-FD-LS, but weakly in the unfitted RBF-FD-LS method. This could lead a reader to intuitively presume that the overall error could behave in favor of RBF-FD-C and RBF-FD-LS.

The smaller error in the unfitted variant can be attributed to the smaller skewness of the stencils which are placed in the interior, but still touch the boundary of Ω. Supporting discussion is available in Appendix A.
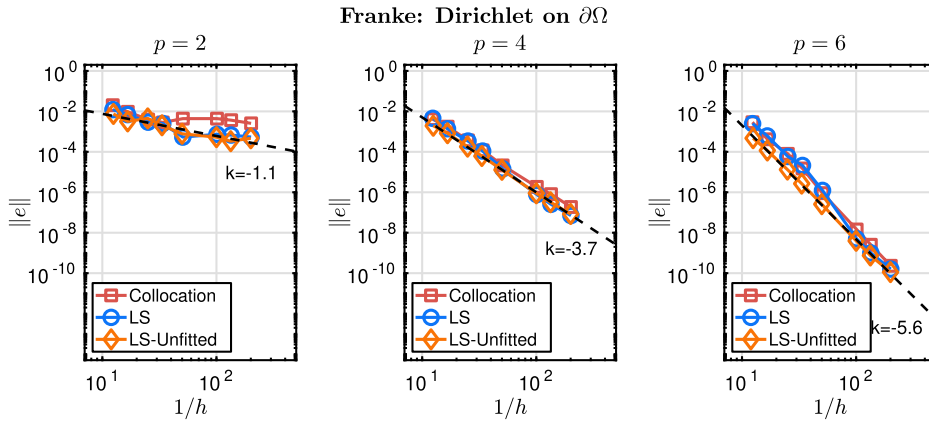
Fig. 8. Error as a function of the inverse internodal distance $1/h$ for different polynomial degrees p. In this case the Franke function is used to manufacture the right-hand-sides of the Poisson equation with the Dirichlet boundary condition.
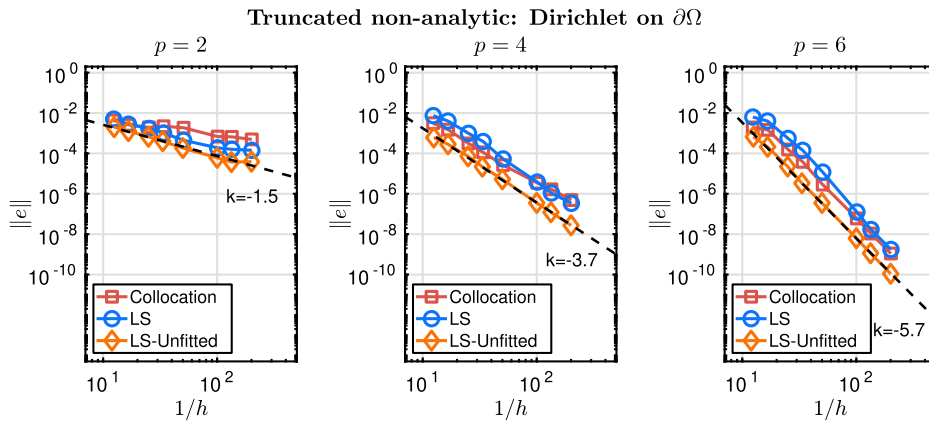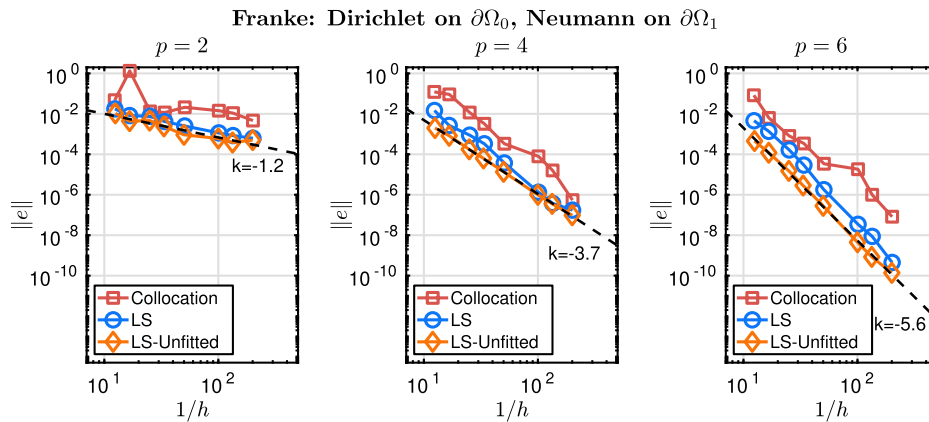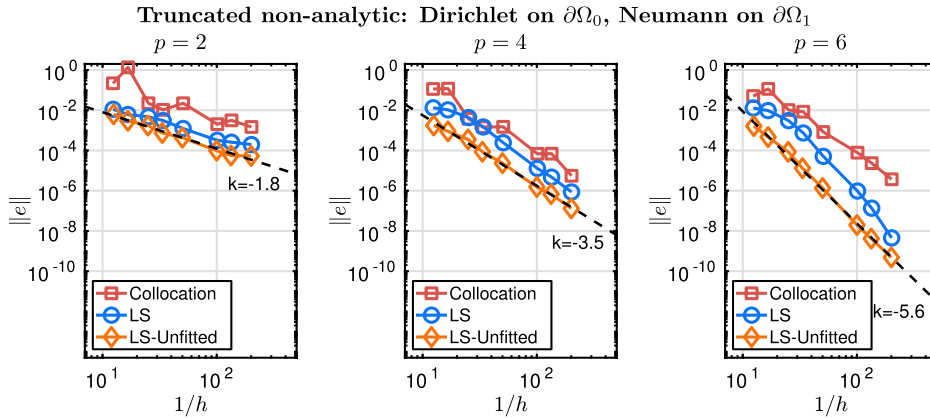

Fig. 9. Error as a function of the inverse internodal distance $1/h$ for different polynomial degrees p. In this case the Truncated non-analytic function is used to manufacture the right-hand-sides of the Poisson equation with the Dirichlet boundary condition.

Fig. 10. Error as a function of the internodal distance h for different polynomial degrees p. In this case the Franke function is used to manufacture the right-hand-sides of the Poisson equation with mixed boundary conditions.

6.5. Convergence under node refinement with Dirichlet and Neumann conditions

Here we follow the formulation from (1) in the sense that the Dirichlet and Neumann boundary conditions are used on two disjoint parts of the domain. The error is again computed as a function of $1/h$ for $p = 2$, $p = 4$ and $p = 6$. The results for the Franke and the Truncated non-analytic functions are given in Fig. 10 and Fig. 11 respectively.

Fig. 11. Error as a function of the internodal distance h for different polynomial degrees p. In this case the Truncated non-analytic function is used to manufacture the right-hand-sides of the Poisson equation with mixed boundary conditions.

We observe that the collocation setting has larger errors than the least-squares counterparts, which is expected [26]. For all p, the unfitted RBF-FD-LS method has a slightly smaller error compared to RBF-FD-LS in the case of the Franke function. The difference between the errors is more pronounced in the Truncated non-analytic case, especially when $p = 4$ and $p = 6$, where the error of the unfitted RBF-FD-LS is 10-times smaller throughout the refinement.

We know that the Franke function does not oscillate around the boundary and that the Truncated non-analytic function is highly oscillatory on a fine scale over the whole Ω. In the first case the less skewed stencils are then not expected to have a significant impact, while in the second case the less skewed stencils are expected to significantly contribute towards a smaller PDE error. This can be accounted to smaller Lebesgue constants which enable a better approximation error see Appendix A for a more detailed explanation.

6.6. Error as a function of runtime with Dirichlet and Neumann conditions

In the subsections above we confirmed that the approximation error is smaller for the unfitted RBF-FD-LS method compared with RBF-FD-LS and RBF-FD-C, for all the cases that were considered. The computational time for both RBF-FD-LS methods is expected to be slightly larger than for RBF-FD-C. In the RBF-FD-LS case, the system (15) is rectangular and requires a sparse QR decomposition as the most expensive intermediate step, so that the solution is obtained. In the RBF-FD-C case, the system (15) is square and requires a sparse LU decomposition. The sparse QR decomposition is slightly more expensive compared to the sparse LU decomposition. In addition, the computational cost of the unfitted RBF-FD-LS method is expected to be slightly larger than for RBF-FD-LS and RBF-FD-C due to the additional degrees of freedom that extend over the boundary of Ω and give rise to an increased number of columns in the matrices D_h and E_h from (16). The question that we address in this subsection is whether the error for the unfitted RBF-FD-LS method is small enough to compensate for the larger computational cost. For this reason we measure the code runtime vs. the obtained accuracy. The runtime is a sum of the following execution times.

- The closest neighbor search for obtaining the stencil neighbor points around every $x_i \in X$.
- Forming and inverting a set of local interpolation matrices (6).
- Forming the evaluation and differentiation weights used in (7).
- Assembly of the PDE matrix used in (15).
- Solving the linear system (15) using *mldivide()* in Matlab.

In Fig. 12 we can observe that in the Franke case error vs. runtime is comparable for all three settings when $p = 2$ and $p = 4$. When $p = 6$ the ratio is in favor of the unfitted RBF-FD method. In Fig. 13 we see that in the Truncated non-analytic case the ratio is in favor of the unfitted RBF-FD method for $p = 4$ and $p = 6$, while it is tied between the three settings for $p = 2$.

The conclusion is that the unfitted RBF-FD method does not lag behind in error vs. runtime compared to the two fitted settings, despite a larger amount of degrees of freedom. Moreover, in cases when stencils are large and the approximated solution oscillatory around the boundaries (the Truncated non-analytic case) it performs significantly better.

6.7. Stability norms and condition numbers under node refinement

Here we measure the condition number of the PDE matrix D_h given in (15). In addition we also measure the stability norm $||E_h||_2 ||D_h^+||_2$ which can be understood as the well-posedness constant that multiplies the consistency term in the
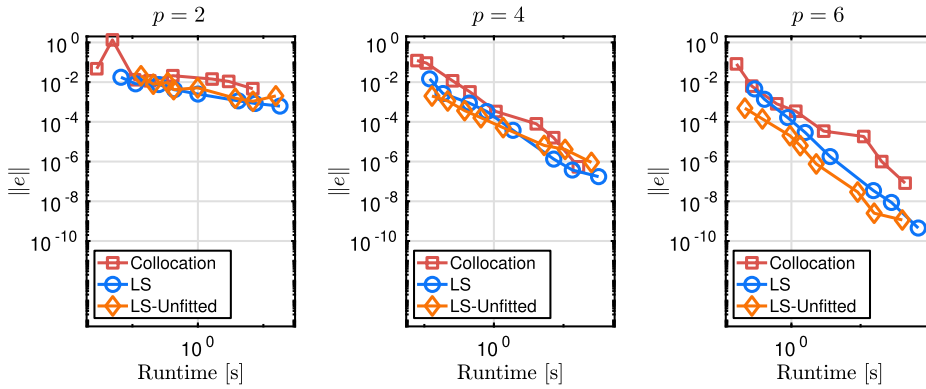
Fig. 12. Error versus runtime for three polynomial degrees p when the Franke function is used as the exact solution.
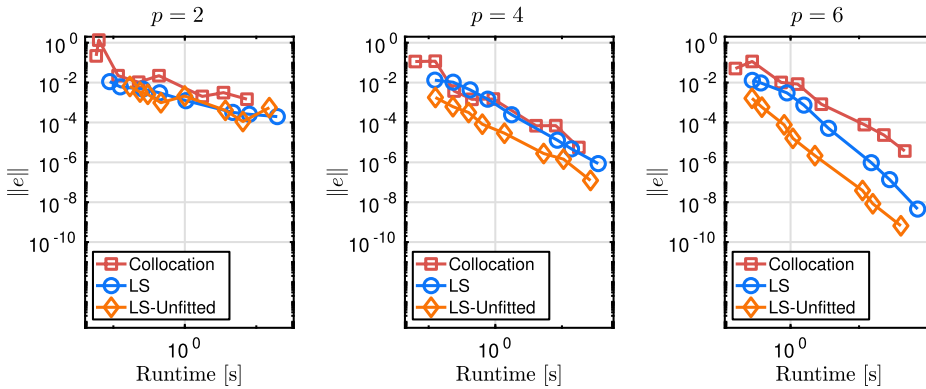

Fig. 13. Error versus runtime for three polynomial degrees p when the Truncated non-analytic function is used as the exact solution.

error estimate (31). In both cases, we impose Neumann and Dirichlet boundary conditions in the same way as in Section 6.5. The condition number of (for example) a matrix D_h is computed by:

$$\kappa(D_h) = ||D_h||_2 \, ||D_h^+||_2 = \frac{\sigma_{\max}(D_h)}{\sigma_{\min}(D_h)},$$

and the stability norm involved in the error estimate (31) by:

$$\frac{1}{\sqrt{M}} ||E_h||_2 \, ||D_h^+||_2 = \frac{1}{\sqrt{M}} \frac{\sigma_{\max}(E_h)}{\sigma_{\min}(D_h)}.$$

Note that when using the two relations stated above in the RBF-FD-C case, the evaluation and interpolation point sets are the same ($Y = X$), which implies $M = N$. Furthermore, in the RBF-FD-C case E_h is a square identity matrix.

In Fig. 14 we see that the stability norm of the unfitted RBF-FD-LS method is constant for $p = 2$. For $p = 4$ and $p = 6$ it is partially decaying until it levels out at a value slightly larger than the stability norm of fitted RBF-FD-LS. This effect is in line with the behavior of the exterior interpolation points, which move closer to the boundary as $\frac{1}{h}$ gets larger. In this way the node layout of the unfitted RBF-FD-LS method is getting increasingly similar to the node layout of the fitted RBF-FD-LS method.

The condition numbers specific to the collocation, the least-squares and the unfitted least-squares setups are referred to by κ_C, κ_{LS} and $\kappa_{U\text{-}LS}$ respectively. In Fig. 15 we see that when $p = 2$ the condition numbers κ_{LS} and $\kappa_{U\text{-}LS}$ behave as h^{-2} while κ_C does not follow any specific pattern (already observed in [26]). Growth with h^{-2} is expected, since it is known that the conditioning of any discretization involving the Laplacian operator scales with at least that rate. In the $p = 4$ and $p = 6$ cases we see a different behavior of $\kappa_{U\text{-}LS}$ compared with κ_{LS}: here $\kappa_{U\text{-}LS}$ is at first large and remains constant until it coincides with κ_{LS} and starts growing with approximately h^{-2}. This effect is conceptually very similar to the behavior of the stability norm of the unfitted RBF-FD-LS method which approaches the stability norm of RBF-FD-LS as $h \to 0$.

6.8. Approximation properties as the polynomial degree are increased

This test gives an insight into the approximation error when the stencil size is increased together with the polynomial degree, while the internodal distance h is fixed. We choose to work with two fixed values of h, namely $h = 0.05$ and 0.015,
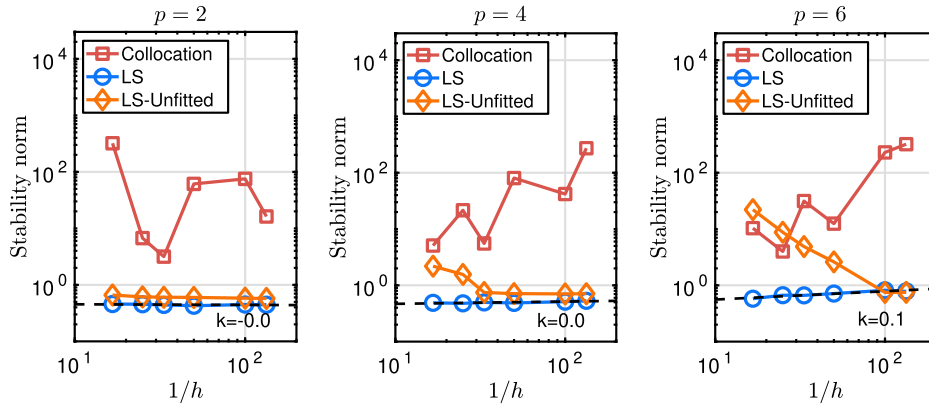
Fig. 14. Stability norm as a function of the inverse internodal distance $1/h$ for three different polynomial degrees p.
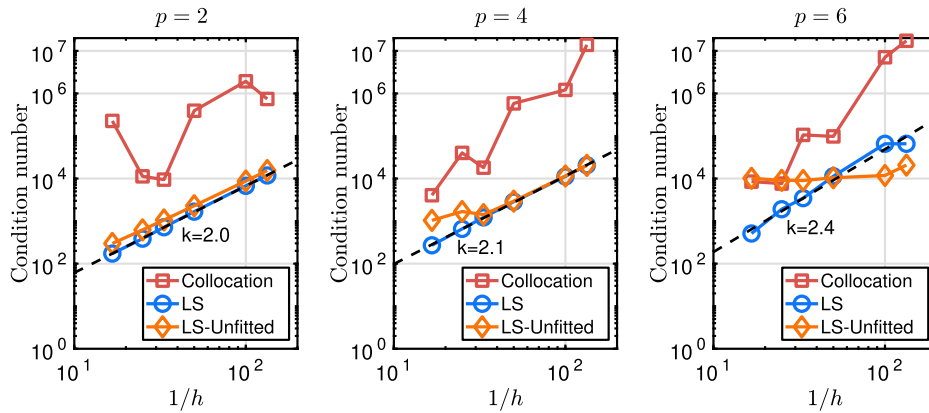

Fig. 15. Conditioning of the PDE matrix D_h as a function of the inverse internodal distance $1/h$ for three different polynomial degrees p.
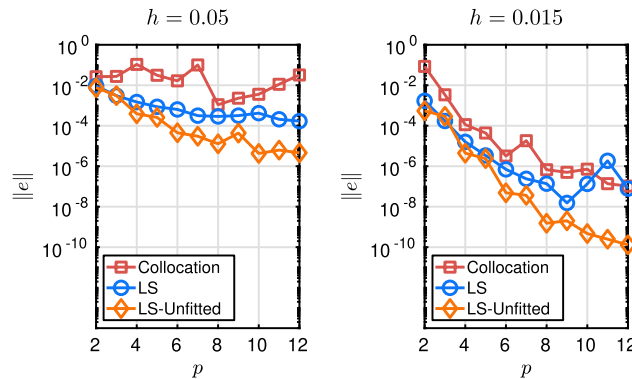

Fig. 16. Error as a function of the polynomial degree p for an under-resolved case ($h = 0.05$) and a well-resolved case ($h = 0.015$) when the Franke function is chosen as the exact solution.

where the former corresponds to a case where the approximated solution is unresolved and the latter when the approximated solution is well resolved. We use Neumann and Dirichlet boundary conditions in the same way as in Section 6.5. Results for the Franke and Truncated non-analytic functions in the role of exact solutions are given in Fig. 16 and Fig. 17 respectively. In both figures the unfitted RBF-FD-LS method is superior in error to RBF-FD-LS and RBF-FD-C, especially when p is large. This is expected: when the stencil size is increasing, the stencils on the boundary get increasingly more skewed in both fitted setups, while in the unfitted RBF-FD-LS method the stencils remain fairly unskewed. For a discussion about the behavior of the interpolation error when a stencil is skewed see Appendix A.
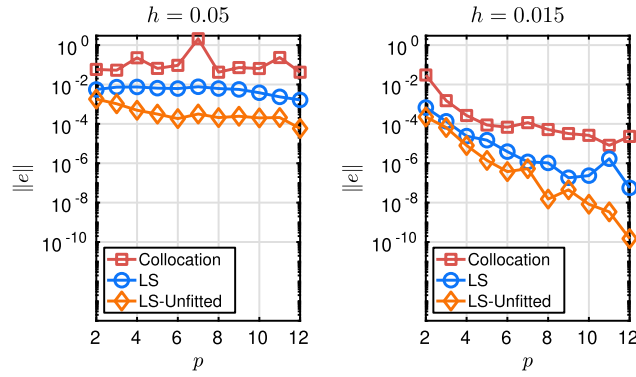
Fig. 17. Error as a function of the polynomial degree p for an under-resolved case ($h = 0.05$) and a well-resolved case ($h = 0.015$) when the Truncated non-analytic function is chosen as the exact solution.
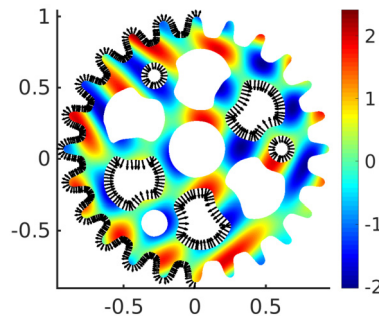

Fig. 18. Solution function u_3 on a sprocket. The black outward normals indicate the locations of the Neumann condition. The locations over the boundary which do not include an arrow indicate the Dirichlet condition.

7. Experiments on a 2D drilled sprocket domain with interior boundary conditions

In this section we consider the drilled 24-tooth sprocket from Fig. 2 as our computational domain. We solve (1), but in addition to the exterior mixed boundary conditions we also introduce interior Dirichlet and Neumann boundary conditions and compare the unfitted RBF-FD-LS method, RBF-FD-LS and RBF-FD-C with a focus on the approximation error under node refinement measured in different norms, and the spatial distribution of the error. The point sets for all methods are chosen in the same way as in Section 6.2.

7.1. Domain Ω

The points of the domain were acquired from a simple monochrome sprocket drawing. By using tools from mathematical morphology we extracted thin borders of the object and applied the Harris feature detector to produce a set of points of interest (e.g. corners) adequately representing the object shape. Each connected component was then parametrized separately via linear arc-length and resampled equidistantly (using the one-dimensional RBF-FD method for interpolation).

7.2. Solution function

Inspired by [31] the solution function used to compute the right-hand-sides of (1) is set to:

$$u_3 = \sin\left(3\pi y^2 + 4.5\pi x\right) - \cos\left(4\pi y - 3\pi x^2\right).$$

Its visual representation is given in Fig. 18. The setup of boundary conditions that we use in the experiments can be seen in Fig. 18, where the black normals over the boundaries indicate the locations of the Neumann boundary conditions and the Dirichlet boundary conditions are employed where there is no marker.

7.3. Convergence under node refinement

We numerically verify that the solution in all three setups (RBF-FD-C, RBF-FD-LS and unfitted RBF-FD-LS) converges for polynomial degrees $p = 2$, $p = 4$, $p = 6$ as $h \to 0$. We compute the relative errors in the following three norms:

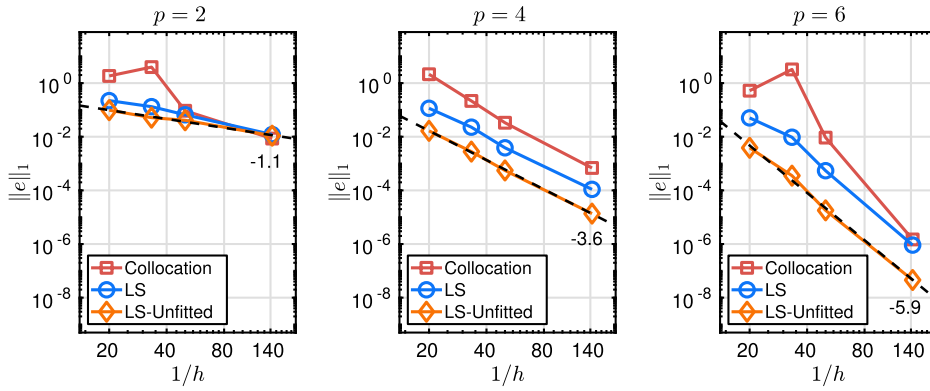**Fig. 19.** Sprocket case. Error in 1-norm as a function of the inverse internodal distance $1/h$ for three different polynomial degrees p.
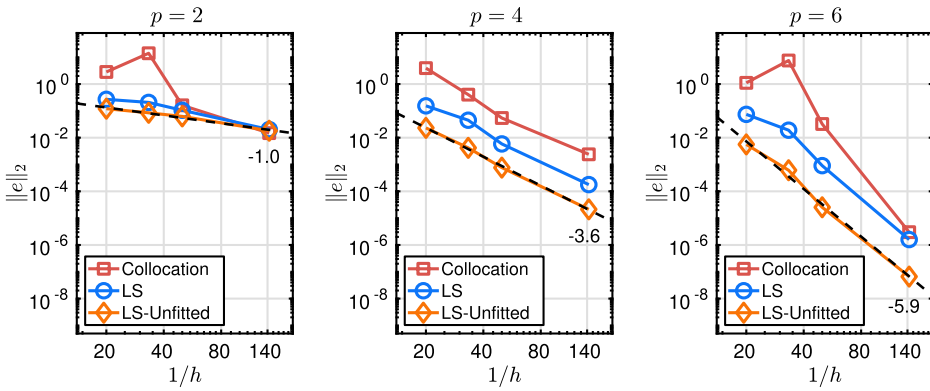

Fig. 20. Sprocket case. Error in 2-norm as a function of the inverse internodal distance $1/h$ for three different polynomial degrees p.
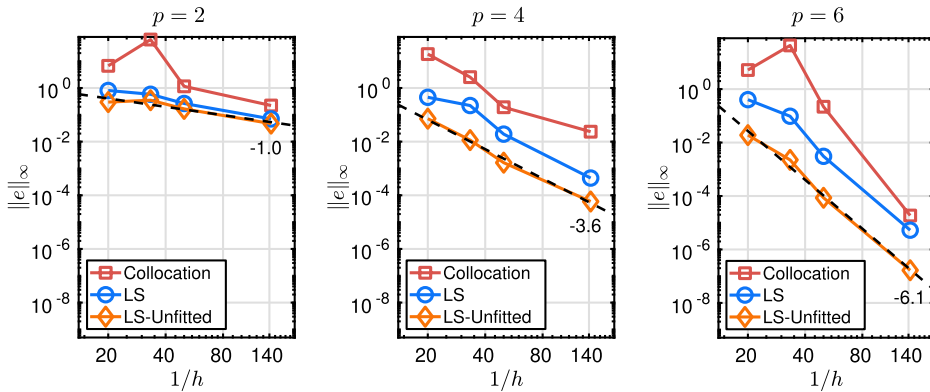

Fig. 21. Sprocket case. Error in ∞-norm as a function of the inverse internodal distance $1/h$ for three different polynomial degrees p.

$$\|e\|_1 = \frac{\|u_h(Y) - u(Y)\|_1}{\|u(Y)\|_1}, \quad \|e\|_2 = \frac{\|u_h(Y) - u(Y)\|_2}{\|u(Y)\|_2}, \quad \|e\|_\infty = \frac{\|u_h(Y) - u(Y)\|_\infty}{\|u(Y)\|_\infty}, \tag{34}$$

where $u_h(Y)$ and $u(Y)$ are the approximate solution and the exact solution consecutively, both sampled in Y points.

The oversampling parameter is set to $q = 5$. For a more detailed study of the oversampling parameter and its impact on the approximation error and the stability norm, we refer the reader to [26].

The convergence results measured in 1-norm, 2-norm and ∞-norm are presented in Fig. 19, 20 and Fig. 21 consecutively. Here we see that for the considered problem, the unfitted RBF-FD-LS method is the most accurate among all three setups, in all three different norms.
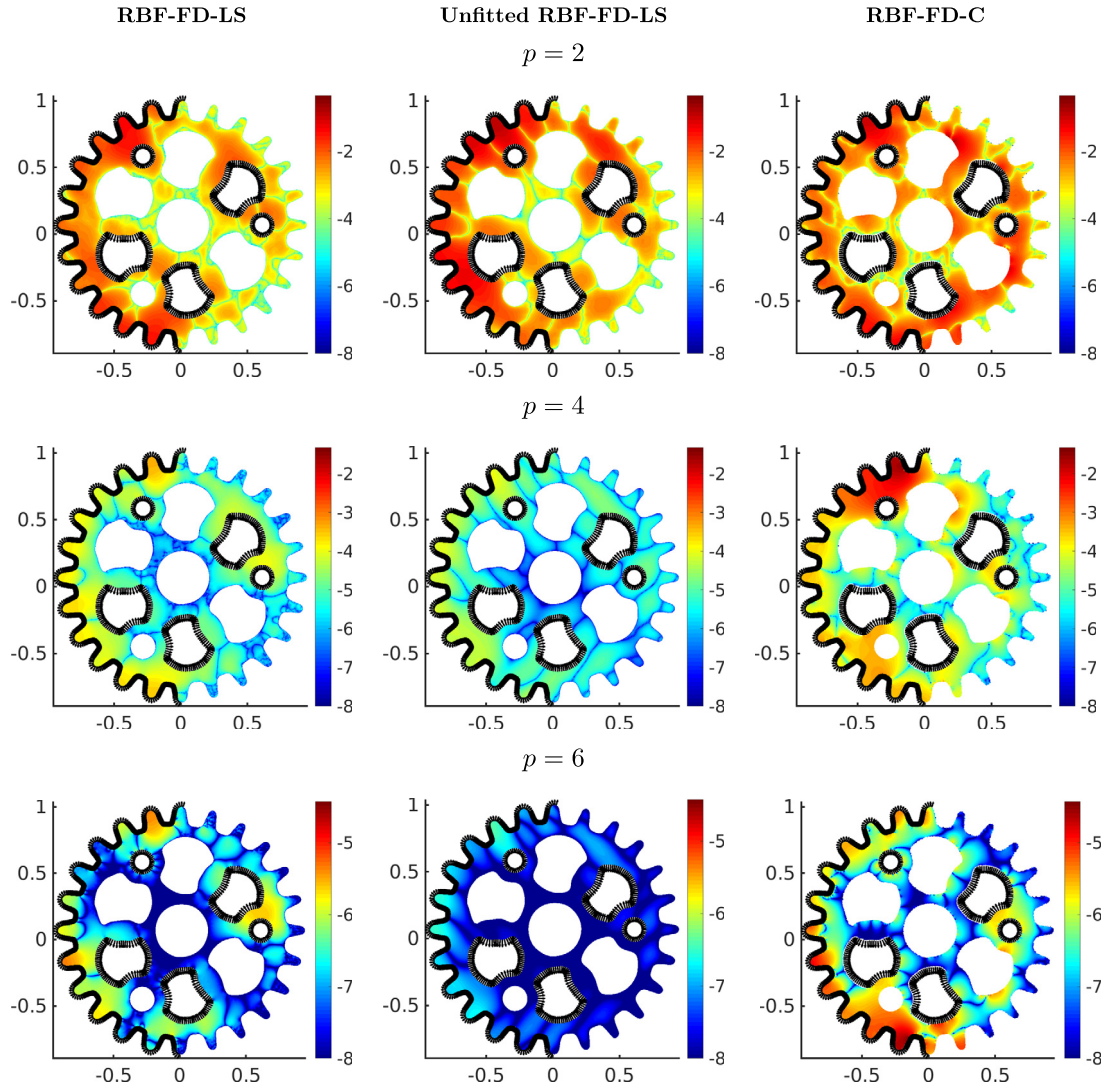
Fig. 22. A comparison of the error distribution (logarithmic scale) over a drilled 24-tooth sprocket. The outward normals indicate the parts of $\partial\Omega$ where the Neumann condition is imposed. The internodal distance is set to $h = 0.007$, the oversampling parameter to $q = 5$ and the polynomial degree to $p = 2$ (first row), $p = 4$ (second row), and $p = 6$ (third row).

7.4. Spatial distribution of the error

Next, we examine the spatial distribution of the error. In Fig. 22 we display the spatial distribution of error in logarithmic scale (the negative integers k in the colorbar are the exponents 10^k) for $p = 2$, $p = 4$, $p = 6$. When the stencils are small (the case $p = 2$) the error distributions for all three settings look fairly similar. A slight increase in the stencil size (the case $p = 4$) causes RBF-FD-C to accumulate more error around the boundaries compared to the other two setups. Increasing the stencil size even further (the case $p = 6$) reveals that RBF-FD-C again collects more error around the boundaries than the other two setups, but also that RBF-FD-LS has larger errors compared with the unfitted RBF-FD-LS method. This is ascribed to the effect of using the extended interpolation points which make the stencils at the domain boundary less skewed (see Appendix A.)

8. Experiments on a 3D diaphragm domain

To show that the previously introduced results also generalize to a realistic scenario in three dimensions we now consider a thoracic diaphragm of a human being in the role of a computational domain Ω. The diaphragm is displayed in Fig. 23 from three different angles. The diaphragm is a thin and non-convex geometry, of which the thickness is approximately 100-times smaller than the largest circumference over its surface. Interpolation points that conform to the diaphragm are thus hard to obtain, which is a good motivation to use the unfitted RBF-FD-LS method for computing a solution to a PDE. In
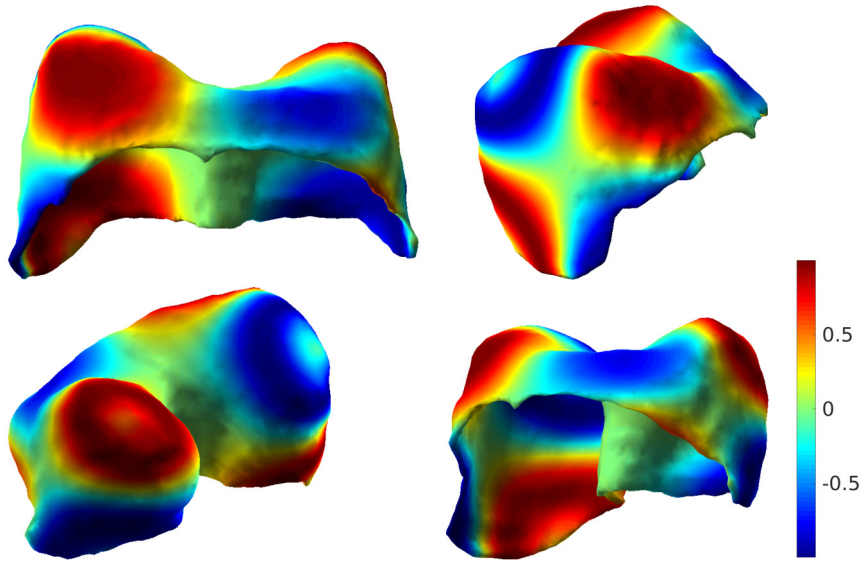
Fig. 23. The solution function $u_4 = sin(6\pi xyz)$ over a diaphragm viewed from four different angles.

this section we solve the same problem as in (1), but this time in three dimensions and only using the unfitted RBF-FD-LS method.

8.1. Solution function

The solution function used to compute the right-hand-sides of a 3D equivalent of (1) is given by:

$$u_4(x, y, z) = \sin(6\pi xyz).$$

The function u_4 over the diaphragm is drawn in Fig. 23. The Dirichlet boundary condition is enforced over the whole top surface of the diaphragm, and the Neumann boundary condition is enforced over the whole bottom surface of the diaphragm.

8.2. Point sets

The interpolation and evaluation points are – conceptually speaking – constructed in the same way as in the two-dimensional cases. We start by using a surface point-cloud of the diaphragm [36] in place of the boundary evaluation points, which are placed in a three-dimensional box that contains interpolation points computed using an algorithm from [28]. Then we enforce $q = 10$ evaluation points (again computed by an algorithm from [28]) around each interpolation point and after that remove those evaluation points which are placed outside of the diaphragm. At last we remove the interpolation points according to the criterion given in Section 4. An instance of the resulting two point sets can be observed in the left image of Fig. 24.

8.3. Convergence under node refinement

We test the convergence of the solution under node refinement for different polynomial degrees used to form the stencil approximations. The oversampling parameter is always fixed to $q = 10$. The spatial distribution of the magnitude of the relative error in logarithmic scale when $h = 0.066$ and $p = 5$ is given in Fig. 24. The convergence results can be observed in Fig. 25. We can see that the numerical solution converges for all polynomial degrees p with at least $\mathcal{O}(h^{p-1})$. This is an expected result according to the error estimate (31) and also according to the numerical experiments previously made for the two-dimensional cases in Section 6 and Section 7.

9. Final remarks

In this paper we presented the unfitted version of the RBF-FD method in the least-squares setting and in this way simplified handling of complex 2D and 3D geometries.

We developed a criterion that numerically establishes the linear independence of the cardinal functions when the interpolation points are placed in the exterior of Ω.
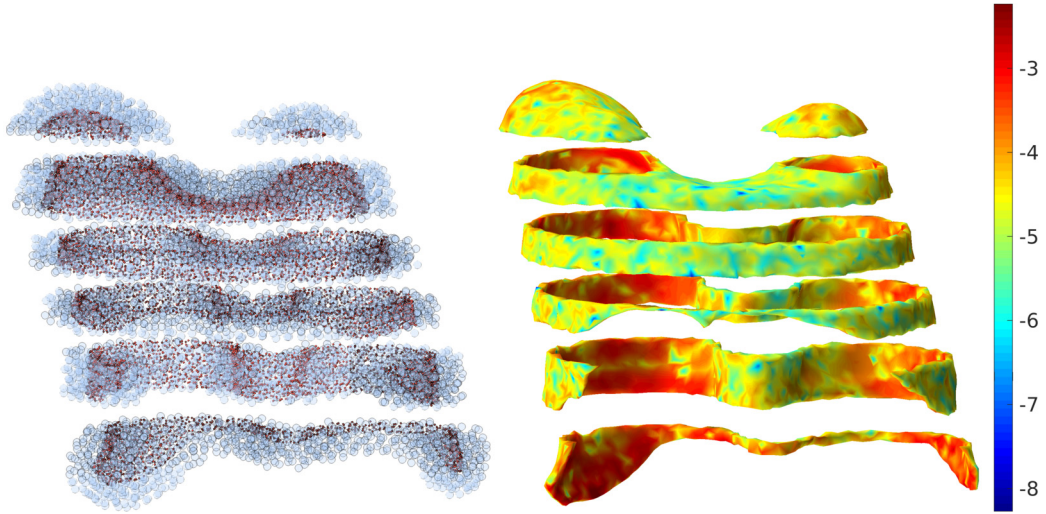
Fig. 24. Both pictures show a diaphragm which is sliced for visualization purposes only. Left: interpolation points (blue markers) are placed over the evaluation points (smaller red markers). The evaluation points discretize the diaphragm. Right: The spatial distribution of the magnitude of the relative error in the logarithmic scale when $h = 0.066$, $p = 5$ and $q = 10$.
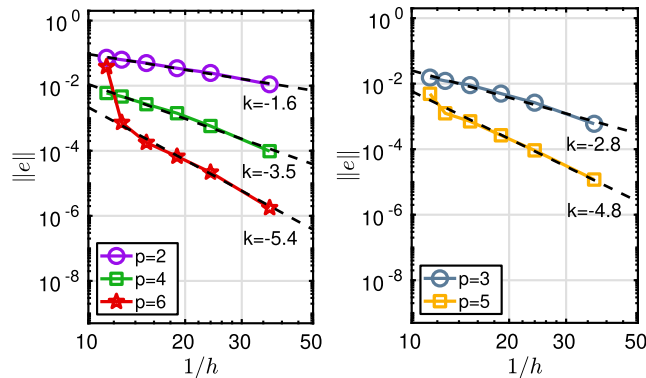

Fig. 25. Error as a function of the inverse internodal distance $1/h$ when the oversampling parameter is set to $q = 10$, for polynomial degrees $p = 2$, $p = 4$, $p = 6$ (left image) and polynomial degrees $p = 3$, $p = 5$ (right image).

Next, we numerically verified on a two-dimensional butterfly domain that the presented method is stable and that the stability properties closely follow the properties of the fitted RBF-FD-LS method. Moreover, the experiments confirmed that the error under node refinement decays as $\mathcal{O}(h^{p-1})$, which was also indicated in our theoretical work. The error was in the majority of the cases found to be smaller compared with the error RBF-FD-LS and RBF-FD-C.

Through an example of a drilled sprocket we numerically demonstrated that the less skewed boundary stencils allow the unfitted RBF-FD-LS method to outperform RBF-FD-LS and RBF-FD-C in the sense of the spatial error distribution, when the discretization is built upon large stencils.

Lastly, using a thoracic diaphragm of a human being as a computational domain, we demonstrated that the unfitted RBF-FD-LS method can also be used in a three dimensional setup, where the observed convergence trends follow $\mathcal{O}(h^{p-1})$, in the same way as in the two-dimensional cases.

Future work includes using the unfitted RBF-FD-LS method for other elliptic PDEs and investigating a formulation for time-dependent PDEs. A potentially interesting topic is also to find a way to form the unfitted RBF-FD-C method, and then compare it to the unfitted RBF-FD-LS method.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.
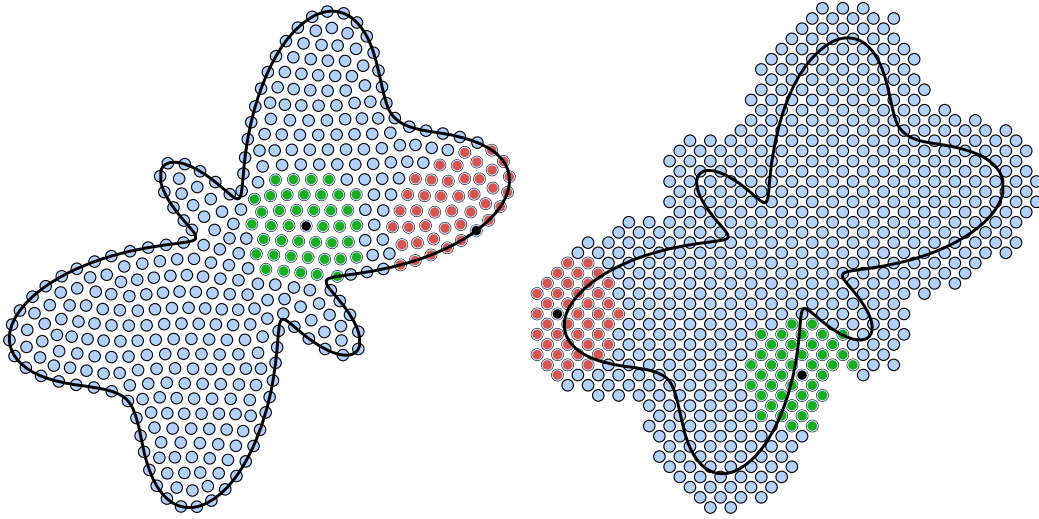
Fig. 26. The left image displays two stencils over a set of interpolation points X for the fitted RBF-FD methods, where the red stencil indicates that its interpolation matrix has the largest norm $\|\tilde{A}\|_\infty$ among all stencils over X. The green stencil is the stencil with the smallest norm $\|\tilde{A}\|_\infty$. The right image shows the red stencil and the green stencil chosen according to the same criterion, but for the unfitted RBF-FD-LS method.

Acknowledgement

Appendix A. Skewness of the stencils influences the size of the interpolation error bound

In this section we provide an insight into the interpolation error behavior over the highly skewed boundary stencils.

We start by making an experiment. We spread interpolation points X over a butterfly domain in the same way as in Section 6.2, and fix $p = 5$, $h = 0.05$. Then we measure the norm of the inverse interpolation matrix of every stencil and pick out the stencils with the smallest and the largest norms:

$$\xi_{\min} = \min_i \|\tilde{A}_{(i)}^{-1}\|_\infty, \quad \xi_{\max} = \max_i \|\tilde{A}_{(i)}^{-1}\|_\infty, \quad i = 1, .., N.$$

We draw the shapes of these two stencils for the fitted RBF-FD-LS method and the unfitted RBF-FD-LS method in Fig. 26, where the stencil corresponding to $\xi_{\min}$ is colored with green color and the stencil corresponding to $\xi_{\max}$ with red color. The rounded values of norms for the fitted RBF-FD-LS method are $\xi_{\min} = 2.3 \cdot 10^2$ and $\xi_{\max} = 2.4 \cdot 10^4$. In this case ξ_{max} is around two orders larger in magnitude and corresponds to the highly skewed stencil in Fig. 26.

For the unfitted RBF-FD-LS method we have $\xi_{\min} = 3.7 \cdot 10^2$ and $\xi_{\max} = 7.2 \cdot 10^2$, where both norms are similar in magnitude, and both stencils are unskewed, despite both being positioned close to the boundary. Thus, we experimentally established that the skewness of stencils is indeed reflected in the size of $\|\tilde{A}^{-1}\|_\infty$. Note that we measured $\xi_{\min}$ and $\xi_{\max}$ in 2-norm and 1-norm, and the outcome did not change in the interpretation. Another perspective is given in Fig. 27, where we plot the $\log_{10}$ spatial distribution of the norm $\|\tilde{A}_{(i)}^{-1}\|_\infty$ for every corresponding stencil center $x_i \in X$. In the fitted RBF-FD case we can see that many boundary stencils have a much larger $\|\tilde{A}_{(i)}^{-1}\|_\infty$ compared to the interior stencils, while in the unfitted RBF-FD-LS case we can see that the matrix norm distribution is much more evenly distributed.

Now we focus on the interpolation over the i-th stencil (7). We know that the interpolation error when using a polynomial interpolant in the Lagrange form [37,38] leads to an error estimate:

$$\|u_h^{(i)}(z) - u^{(i)}(z)\|_\infty \le (1 + \Lambda^{(i)}) \|u^{(i)}(z) - p_*^{(i)}(z)\|_\infty, \tag{35}$$

where $u^{(i)}(z)$ is an exact function value over the i-th stencil, $p_*^{(i)}(z)$ is the best interpolating polynomial over the i-th stencil, and:

$$\Lambda^{(i)} = \max_z \sum_{k=1}^n |\psi_k^{(i)}(z)| = \max_z \|\underline{\psi}^{(i)}(z)\|_1, \tag{36}$$
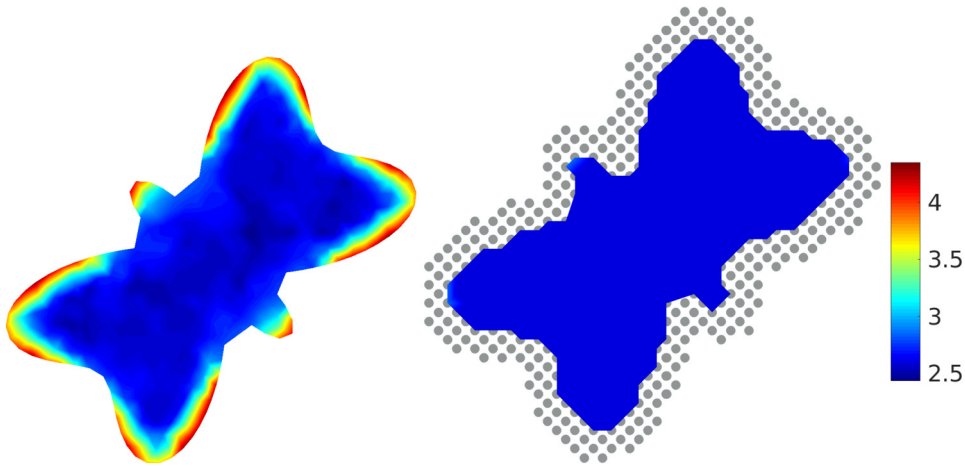
Fig. 27. Both images show a spatial distribution of the matrix norm $\|\tilde{A}_{(i)}^{-1}\|_\infty$ intrinsic to the corresponding stencil center $x_i \in X$, in $\log_{10}$ scale. Left image is showing the distribution for the fitted RBF-FD methods, while the right image is showing the distribution for the unfitted RBF-FD-LS method.

is a Lebesgue constant over the i-th stencil. Here $\underline{\psi}^{(i)}(z)$ is a vector of local cardinal functions over the i-th stencil. For a fixed $u^{(i)}(z)$, the term $p_*^{(i)}(z)$ is fixed and thus the upper limit of the error (35) depends only on $\Lambda^{(i)}$. In order to bound $\Lambda^{(i)}$ we plug the definition of $\underline{\psi}^{(i)} = \{\psi_k(z)\}_{k=1}^n$ from (7) into (36):

$$
\begin{aligned}
\Lambda^{(i)} = \max_z \|\underline{\psi}^{(i)}(z)\|_1 &= \max_z \|\underline{w}^{(i)}(z)\|_1 = \max_z \|b^{(i)}(z)\,\tilde{A}_{(i)}^{-1}\|_1 \\
&\le \max_z \|b^{(i)}(z)\|_1 \, \|\tilde{A}_{(i)}^{-1}\|_1 \\
&\le \sqrt{n}\, \max_z \|b^{(i)}(z)\|_1 \, \|\tilde{A}_{(i)}^{-1}\|_\infty
\end{aligned}
\tag{37}
$$

From the bound and the experiment in which we found out that the worst $\|\tilde{A}^{-1}\|_\infty$ among all stencils belong to the skewed stencil, we have that for a fixed p, n, h, the Lebesgue constant has a larger upper limit for the skewed stencil compared to the less skewed stencil and thus, the interpolation error also has a larger upper limit over the skewed stencil.

Finally, we have to understand how (37) relates to the global solution defined in (10). We restrict the global solution $u_h(y)$ to one Voronoi cell $\mathcal{V}_i$, where the cell is centered around a point $x_i \in X$. This restriction is denoted by $u_h(y)|_{\mathcal{V}_i}$. In the fitted/unfitted RBF-FD-LS method every $u_h(y)|_{\mathcal{V}_i}$ is approximated using an interpolant over the i-th stencil, where x_i has a role of the stencil center point. The interpolation error estimate (35) for the i-th stencil then applies to every interpolant $u_h(y)|_{\mathcal{V}_i}$: from an interpolation perspective, $u_h(y)|_{\mathcal{V}_i}$ will contain an error dominated by the Lebesgue constant over the i-th stencil. It is therefore beneficial to keep $\|\tilde{A}_{(i)}^{-1}\|_\infty$ as small as possible, i.e., to keep the stencils skewed as little as possible.

References

[1] 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.

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

[3] S. Milovanović, L. von Sydow, Radial basis function generated finite differences for option pricing problems, Comput. Math. Appl. 75 (4) (2018) 1462–1481, https://doi.org/10.1016/j.camwa.2017.11.015.

[4] B. Fornberg, N. Flyer, J.M. Russell, Comparisons between pseudospectral and radial basis function derivative approximations, IMA J. Numer. Anal. 30 (1) (2010) 149–172, https://doi.org/10.1093/imanum/drn064.

[5] J. Slak, G. Kosec, On generation of node distributions for meshless PDE discretizations, SIAM J. Sci. Comput. 41 (5) (2019) A3202–A3229, https://doi.org/10.1137/18M1231456.

[6] V. Shankar, R.M. Kirby, A.L. Fogelson, Robust node generation for mesh-free discretizations on irregular domains and surfaces, SIAM J. Sci. Comput. 40 (4) (2018) A2584–A2608, https://doi.org/10.1137/17M114090X.

[7] B. Fornberg, N. Flyer, Fast generation of 2-D node distributions for mesh-free PDE discretizations, Comput. Math. Appl. 69 (7) (2015) 531–544, https://doi.org/10.1016/j.camwa.2015.01.009.

[8] R. Zamolo, E. Nobile, Two algorithms for fast 2D node generation: application to RBF meshless discretization of diffusion problems and image halftoning, Comput. Math. Appl. 75 (12) (2018) 4305–4321, https://doi.org/10.1016/j.camwa.2018.03.031.

[9] C.S. Peskin, Flow patterns around heart valves, in: Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics (Univ. Paris VI and XI, Paris, 1972), vol. II, in: Lecture Notes in Phys., vol. 19, 1973, pp. 214–221.

[10] C.S. Peskin, The immersed boundary method, Acta Numer. 11 (2002) 479–517, https://doi.org/10.1017/S0962492902000077.

[11] R.J. LeVeque, Z.L. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM J. Numer. Anal. 31 (4) (1994) 1019–1044, https://doi.org/10.1137/0731014.

[12] D.M. Ingram, D.M. Causon, C.G. Mingham, Developments in Cartesian cut cell methods, in: MODELLING 2001 (Pilsen), Math. Comput. Simul. 61 (3–6) (2003) 561–572, https://doi.org/10.1016/S0378-4754(02)00107-6.

[13] G. Yang, D. Causon, D. Ingram, R. Saunders, P. Battent, A cartesian cut cell method for compressible flows part A: static body problems, Aeronaut. J. 101 (1002) (1997) 47–56.

[14] J.W. Barrett, C.M. Elliott, Fitted and unfitted finite-element methods for elliptic equations with smooth interfaces, IMA J. Numer. Anal. 7 (3) (1987) 283–300, https://doi.org/10.1093/imanum/7.3.283.

[15] A. Hansbo, P. Hansbo, An unfitted finite element method, based on Nitsche's method, for elliptic interface problems, Comput. Methods Appl. Mech. Eng. 191 (47–48) (2002) 5537–5552, https://doi.org/10.1016/S0045-7825(02)00524-8.

[16] E. Burman, S. Claus, P. Hansbo, M.G. Larson, A. Massing, CutFEM: discretizing geometry and partial differential equations, Int. J. Numer. Methods Eng. 104 (7) (2015) 472–501, https://doi.org/10.1002/nme.4823.

[17] S. Sticko, G. Kreiss, Higher order cut finite elements for the wave equation, J. Sci. Comput. 80 (3) (2019) 1867–1887, https://doi.org/10.1007/s10915-019-01004-2.

[18] S. Sticko, G. Ludvigsson, G. Kreiss, High-order cut finite elements for the elastic wave equation, Adv. Comput. Math. 46 (3) (2020) 45, https://doi.org/10.1007/s10444-020-09785-z.

[19] D. Elfverson, M.G. Larson, K. Larsson, Cutiga with basis function removal, Adv. Model. Simul. Eng. Sci. 5 (1) (2018) 1–19.

[20] J. Nitsche, Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind, Abh. Math. Semin. Univ. Hamb. 36 (1971) 9–15, https://doi.org/10.1007/BF02995904.

[21] I. Babuška, The finite element method with penalty, Math. Comput. 27 (1973) 221–228, https://doi.org/10.2307/2005611.

[22] B. Fornberg, A pseudospectral fictitious point method for high order initial-boundary value problems, SIAM J. Sci. Comput. 28 (5) (2006) 1716–1729, https://doi.org/10.1137/040611252.

[23] C.S. Chen, A. Karageorghis, F. Dou, A novel RBF collocation method using fictitious centres, Appl. Math. Lett. 101 (2020) 106069, https://doi.org/10.1016/j.aml.2019.106069.

[24] J. Lin, Y. Zhao, D. Watson, C.S. Chen, The radial basis function differential quadrature method with ghost points, Math. Comput. Simul. 173 (2020) 105–114, https://doi.org/10.1016/j.matcom.2020.01.006.

[25] E. Larsson, V. Shcherbakov, A. Heryudono, A least squares radial basis function partition of unity method for solving PDEs, SIAM J. Sci. Comput. 39 (6) (2017) A2538–A2563, https://doi.org/10.1137/17M1118087.

[26] I. Tominec, E. Larsson, A. Heryudono, A least squares radial basis function finite difference method with improved stability properties, arXiv:2003.03132, 2020.

[27] O. Davydov, Error bounds for a least squares meshless finite difference method on closed manifolds, arXiv:1910.03359, 2019.

[28] K. van der Sande, B. Fornberg, Fast variable density 3-D node generation, arXiv:1906.00636, 2019.

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

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

[31] V. Bayona, N. Flyer, B. Fornberg, 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, https://doi.org/10.1016/j.jcp.2016.12.008.

[32] I. Tominec, Rectangular and square RBF-FD matrices in MATLAB, https://github.com/IgorTo/rbf-fd, 2021, https://doi.org/10.5281/zenodo.4525550.

[33] C.C. Paige, M.A. Saunders, LSQR: an algorithm for sparse linear equations and sparse least squares, ACM Trans. Math. Softw. 8 (1) (1982) 43–71, https://doi.org/10.1145/355984.355989.

[34] J.M. Lee, Introduction to Smooth Manifolds, 2nd edition, Graduate Texts in Mathematics, vol. 218, Springer, New York, 2013.

[35] P.-O. Persson, G. Strang, A simple mesh generator in Matlab, SIAM Rev. 46 (2) (2004) 329–345, https://doi.org/10.1137/S0036144503429121.

[36] E. Larsson, P.-F. Villard, I. Tominec, U. Sundin, N. Cacciani, A radial basis function partition of unity method for geometry reconstruction and PDE solution in thin structures, Manuscript in preparation.

[37] M.J.D. Powell, Approximation Theory and Methods, Cambridge University Press, Cambridge-New York, 1981.

[38] B.A. Ibrahimoglu, Lebesgue functions and Lebesgue constants in polynomial interpolation, J. Inequal. Appl. (2016) 93, https://doi.org/10.1186/s13660-016-1030-3.