PAPER • **OPEN ACCESS**

Meshfree RBF–FD methods for numerical simulation of PDE problems

Journal of Physics: Conference Series **2766** (2024) 012158 doi:10.1088/1742-6596/2766/1/012158

Meshfree RBF–FD methods for numerical simulation of PDE problems

Elisabeth Larsson[1], Boštjan Mavrič[2], Andreas Michael[1], Fatemeh Pooladi[3] and Igor Tominec[4]

[1] Department of Information Technology, Uppsala University, Uppsala, Sweden
[2] Faculty of Mechanical Engineering, University of Ljubljana, Ljubljana, Slovenia
[3] Department of Mathematics, Persian Gulf University, Bushehr, Iran
[4] Department of Mathematics, Stockholm University, Stockholm, Sweden

E-mail: elisabeth.larsson@it.uu.se

Abstract. The meshfree RBF–FD method is becoming an increasingly popular discretization method for challenging PDE problems due to its flexibility with respect to geometry and its ease of implementation. Using a combination of polyharmonic spline basis functions and polynomial basis functions has proven to be particularly successful with a convergence rate depending on the polynomial degree and good behavior at boundaries due to the contribution of the spline part. A challenge that has been observed is that derivative boundary conditions can give rise to large errors. A known remedy for this problem is to introduce one or more layers of ghost points, which improves the boundary approximations. In a recent paper, Tominec, Larsson, Heryudono (2021), it was shown that using oversampling is another effective way to handle this problem. Oversampling also allows for theoretical analysis of the method, which can then be viewed as a discretization of a continuous least-squares projection. By looking deeper into the error estimates we gain an understanding about how to best implement the method.

1. Introduction

The radial basis function-generated finite difference method (RBF–FD) is a generalization of finite difference approximations to scattered node layouts. A local RBF interpolant is formed over a stencil node set, and by differentiating the interpolant, we get the weights in the stencil approximation of local derivatives. The method first appeared in [1] and was followed by [2, 3, 4, 5, 6, 7]. In these early papers infinitely smooth RBFs like the multiquadric were typically used, while the current trend is to combine a low order polyharmonic spline $\phi(r) = r^3$ with a polynomial basis $p_j(x)$, $j = 1, \ldots, m$, for polynomials of degree up to p [8, 9, 10, 11, 12, 13].

We follow the notation in [14] and introduce a scattered data problem, where $X = \{x_k\}_{k=1}^N$ is the global node set in $\mathbb{R}^d$ and $X_k = \{x_j^{(k)}\}_{j=1}^n$ are the nodes in the local influence domain Ω_k of the stencil centered at x_k. We define the local RBF interpolant as

$$u_h(x) = \sum_{j=1}^{n} \lambda_j^{(k)} \phi(\|x - x_j^{(k)}\|_2) + \sum_{j=1}^{m} \mu_j^{(k)} p_j(x), \quad x \in \Omega_k, \tag{1}$$

with the constraint $\sum_{j=1}^n \lambda_j^{(k)} p_\ell(x_j^{(k)}) = 0$, $\ell = 1, \ldots, m$. As a rule of thumb, we choose $n \geq 2m$ in two dimensions [10]. We form a linear system to express the interpolation conditions for the

nodal values $u_h(X_k)$

$$\begin{pmatrix} A^{(k)} & P^{(k)} \\ (P^{(k)})^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \underline{\lambda}^{(k)} \\ \underline{\mu}^{(k)} \end{pmatrix} = \begin{pmatrix} u_h(X_k) \\ \underline{0} \end{pmatrix} \quad \Rightarrow \quad \begin{pmatrix} \underline{\lambda}^{(k)} \\ \underline{\mu}^{(k)} \end{pmatrix} = \begin{pmatrix} A^{(k)} & P^{(k)} \\ (P^{(k)})^T & \mathbf{0} \end{pmatrix}^{-1} \begin{pmatrix} u_h(X_k) \\ \underline{0} \end{pmatrix}, \quad (2)$$

where the matrices have elements $a_{ij} = \phi(\|x_i^{(k)} - x_j^{(k)}\|_2)$ and $p_{ij} = p_j(x_i^{(k)})$. The stencil matrix is invertible if the nodes are distinct and P has full rank. To find the stencil weights (for interpolation) we evaluate the interpolant at a specific point x

$$\begin{aligned} u_h(x) &= \begin{pmatrix} \phi(\|x - x_1^{(k)}\|_2) & \cdots & \phi(\|x - x_n^{(k)}\|_2) & p_1(x) & \cdots & p_m(x) \end{pmatrix} \begin{pmatrix} \underline{\lambda}^{(k)} \\ \underline{\mu}^{(k)} \end{pmatrix} \\ &= \begin{pmatrix} \phi(\|x - x_1^{(k)}\|_2) & \cdots & \phi(\|x - x_n^{(k)}\|_2) & p_1(x) & \cdots & p_m(x) \end{pmatrix} \begin{pmatrix} A^{(k)} & P^{(k)} \\ (P^{(k)})^T & \mathbf{0} \end{pmatrix}^{-1} \begin{pmatrix} u_h(X_k) \\ \underline{0} \end{pmatrix} \\ &\equiv \begin{pmatrix} \psi_1^{(k)}(x) & \cdots & \psi_n^{(k)}(x) & \eta_1^{(k)}(x) & \cdots & \eta_m^{(k)}(x) \end{pmatrix} \begin{pmatrix} u_h(X_k) \\ \underline{0} \end{pmatrix}. \end{aligned} \quad (3)$$

We choose to express the interpolant in terms of the cardinal functions $\psi_j^{(k)}(x)$ as

$$u_h(x) = \sum_{j=1}^{n} u_h(x_j^{(k)}) \psi_j^{(k)}(x) \equiv \sum_{j=1}^{n} u_j^{(k)} \psi_j^{(k)}(x). \quad (4)$$

Stencil weights for derivatives can then be expressed using derivatives of the cardinal functions.

When solving PDEs using RBF–FD (and other RBF methods), it is known that derivative boundary conditions may lead to large errors close to the boundary. A popular remedy is to introduce a layer of ghost points and then enforce both the PDE and the boundary condition at the boundary nodes, see, e.g., [15, 9, 10, 11, 16].

In some recent papers, this has been generalized to oversampled RBF–FD [17, 18, 19], and (oversampled) unfitted RBF–FD [20, 21]. For these methods, the number of least-squares evaluation points for the PDE and boundary conditions exceeds the number of center points, which can be placed inside the domain or both inside and outside the domain. Figure 1 illustrates the different node layouts. Unless otherwise stated, the nodes that are used are generated using

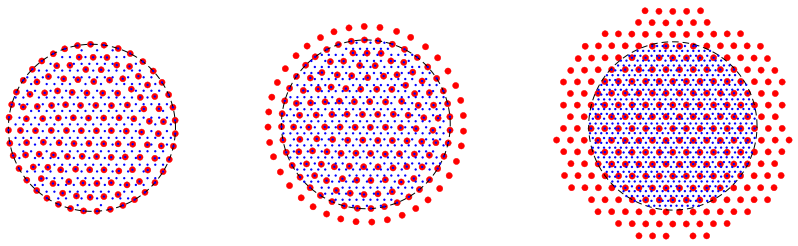

Figure 1. Nodes in a circular domain for a boundary-fitted case (left), with one layer of ghost nodes (middle), and for an unfitted case (right). The nodes are shown with large red markers. The smaller blue points are least-squares evaluation points for an oversampling factor $\alpha = 3$.

Distmesh [22], to limit the influence of node irregularities on the errors.

In this paper, we investigate different aspects of the errors in RBF–FD approximations. Related and to some extent similar investigations were made in [17, 18, 14].

2. The behavior of single cardinal functions

When a cardinal function $\psi_j^{(k)}$ is evaluated over the stencil, it shows the overall influence of the nodal value at $u_j^{(k)}$ on the local interpolant. In this section, we illustrate how a single cardinal functions and its derivative behave at different locations in the stencil. Figure 2 shows results for a cardinal function at the center of the stencil, a cardinal function one step away from the edge of the stencil, and a cardinal function at the edge of the stencil, for polynomial degree $p = 4$ in the PHS + polynomial basis.

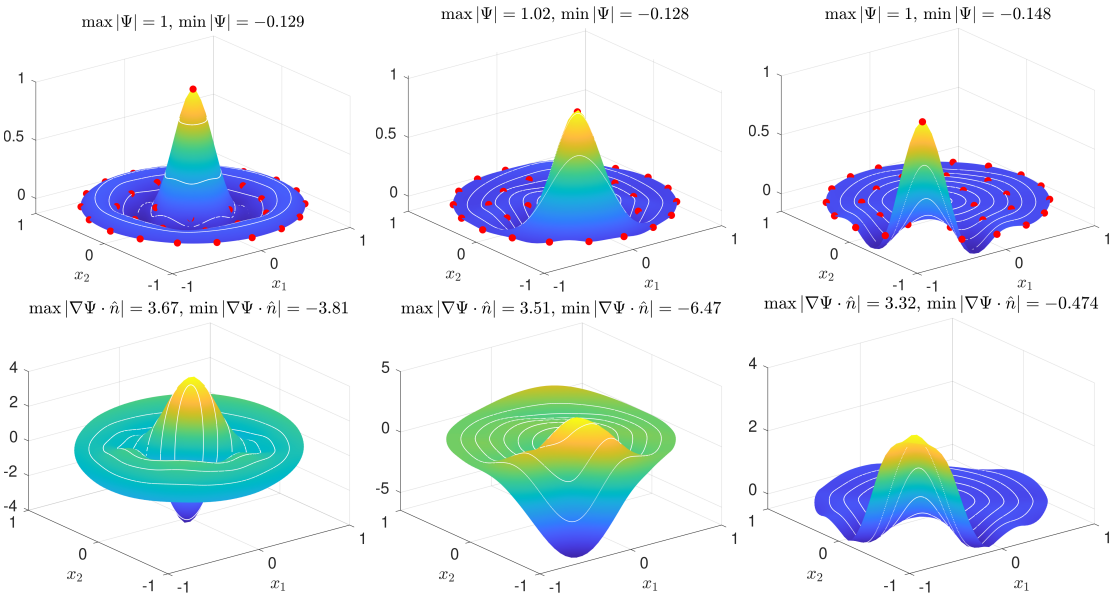

Figure 2. The cardinal functions $\Psi(x)$ (top row) and the diagonal derivative $\nabla\Psi(x) \cdot (-1, -1)$ (bottom row), for three different points, at the center (left), next to the edge (middle), and at the edge (right), in an RBF–FD stencil with $n = 35$ nodes and polynomial degree $p = 4$.

In Figure 3, the polynomial degree is increased to $p = 8$. We note that in particular, the derivative for the cardinal function next to the boundary gets a large negative value that may cause edge effects.

3. Lebesgue functions for the three choices

Here we consider interpolation over $N_c > n$ nodes in the unit disc. With a boundary fitted approximation, stencils near the boundary become skewed, which means that the cardinal functions near the edge will have a large impact. With one layer of ghost nodes, stencils centered at the boundary are less skewed, and the influence of the cardinal functions at the stencil edge is reduced. For the unfitted method, none of the stencils are skewed.

If we have exact nodal values $u(X)$, then the Lebesgue function measures the worst case error at any point in the domain such that $|u_h(x) - u(x)|_\infty \leq \Lambda(x)\|u\|_\infty$. For the RBF–FD case, we define the Lebesgue function as

$$\Lambda(x) = \sum_{j=1}^{n} |\psi_j^{(k(x))}(x)|, \tag{5}$$

where $k(x)$ is chosen as the index of the node point that is closest to x. That is, when we evaluate $u_h(x)$ we choose the nearest stencil. This implicitly defines a division of the domain into Voronoi regions, and the global RBF–FD interpolant will have small discontinuities at the

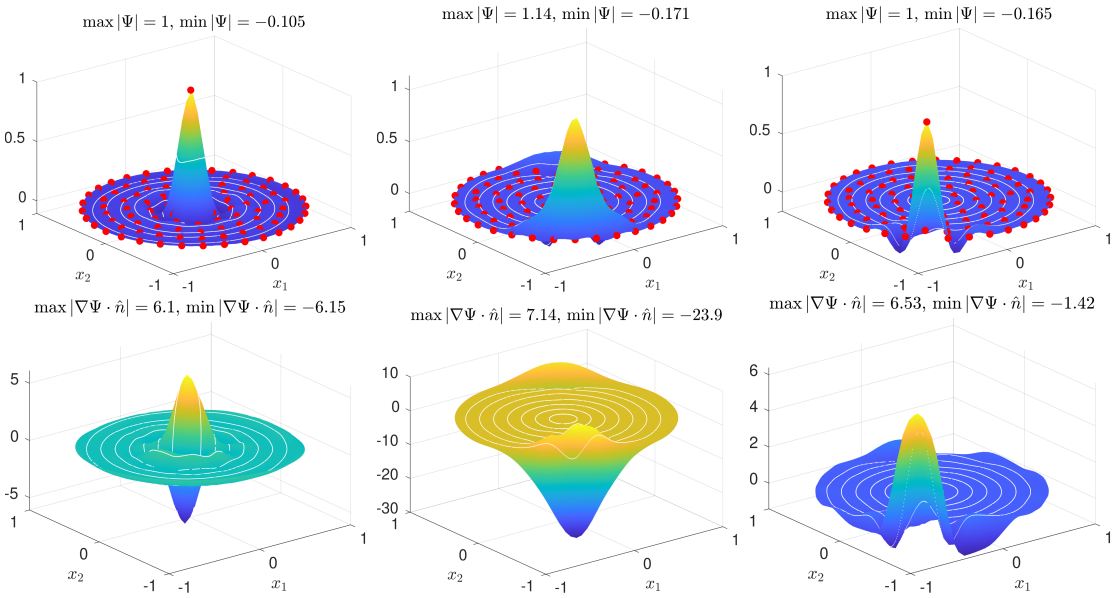

Figure 3. The cardinal functions and their derivatives for three different points, at the center, next to the edge, and at the edge, in an RBF–FD stencil with $n = 99$ nodes and polynomial degree $p = 8$.

edges of these regions [17, 18]. Figures 4 and 5 show the Lebesgue functions for the three different types of approximations and for polynomial degrees $p = 4$ and $p = 8$. For the lower order approximation, the boundary effects are pronounced for the fitted node case. This is even more so for the higher order approximation, but then also one layer of ghost nodes is insufficient to suppress the boundary effects, and only the unfitted approximation has a uniform behavior.

4. Global error estimates for RBF–FD

In [17], the following global error estimate for oversampled RBF–FD applied to a Poisson problem with mixed boundary conditions was derived:

$$\|u_h - u\|_{\ell_2(\Omega)} \leq \sqrt{2}C \left(\frac{(1+\tau_0)(1+\eta_0)(1+\eta_a)}{(1-\tau_a)} \right)^{\frac{1}{2}} \left(c_0 h^{p+1} + c_1 h^p + c_2 h^{p-1} \right) |u|_{W_\infty^{p+1}(\Omega)}, \quad (6)$$

where C and c_i are constants, τ_i are integration errors and η_i are smoothing errors. These errors were defined in [17], and further investigated in [14], where it was also shown how to evaluate the errors locally using generalized eigenvalue problems. Here we apply the same technique to the full problem, to evaluate the factors that contribute to the error for the three approximation modes. In Figure 6, we show the integration errors, which indicate how well the oversampling reproduces the continuous norm, and the smoothing errors that measure how far the RBF–FD solution and its derivatives are from a smooth function. The oversampling factor α is the ratio of the number of evaluation points and the number of node points. To compute the smoothed function that u_h is compared with, we applied compactly supported partition of unity weight functions to each stencil result. The radius of the support is given by $R_0(1+\delta)$, where R_0 is the smallest radius that encloses the local Voronoi region and δ is the relative overlap parameter.

5. Discussion

The experiments we performed show that approximation errors at the boundary are greatly reduced when moving nodes outside the domain. For higher order methods more nodes outside

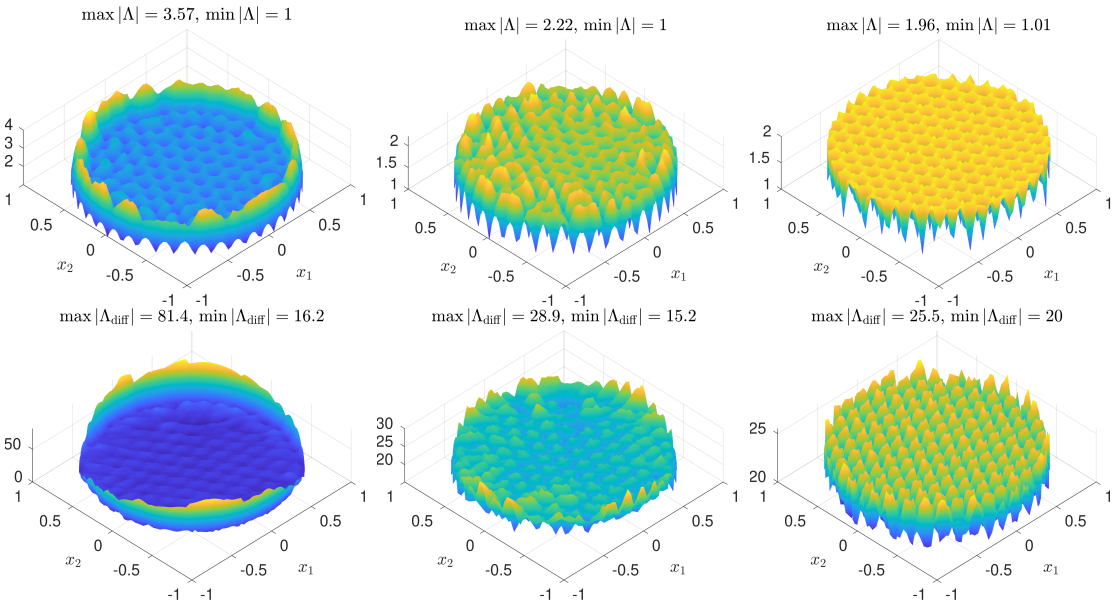

Figure 4. The Lebesgue functions for interpolation (top row) and for the diagonal derivative (bottom row) for the boundary-fitted node set (left), with one layer of ghost nodes (middle), and for the unfitted node set (right). RBF–FD stencils with $n = 30$ nodes and polynomial degree $p = 4$ where used. The total number of centers inside the unit disc domain is $N_c = 139$ in the first two cases and $N_c = 138$ in the unfitted case.

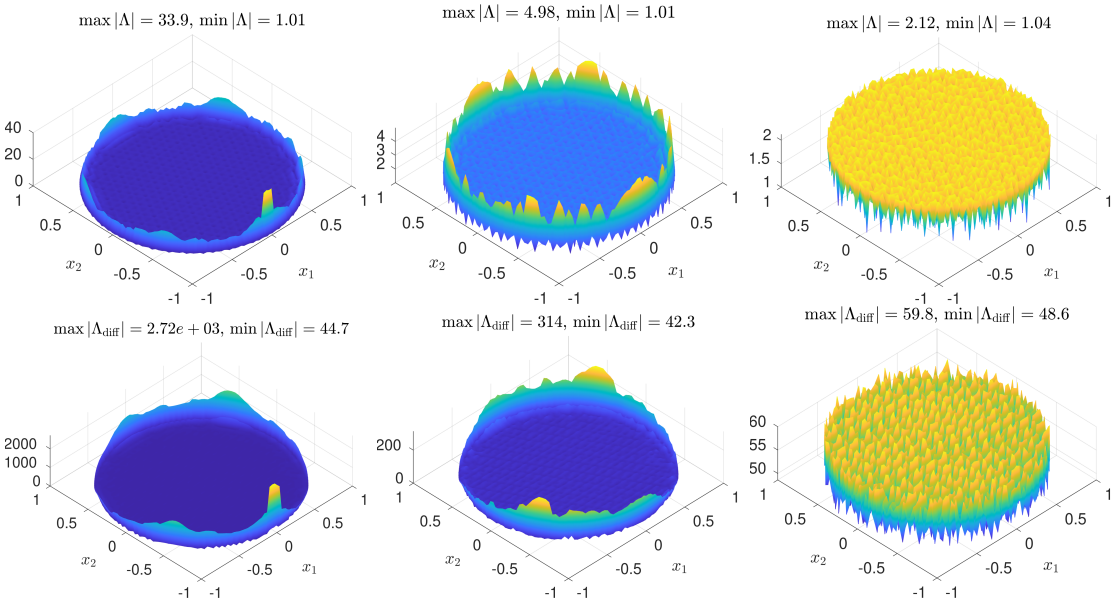

Figure 5. The Lebesgue functions for interpolation (top row) and for the diagonal derivative (bottom row) for the boundary-fitted node set (left), with one layer of ghost nodes (middle), and for the unfitted node set (right). RBF–FD stencils with $n = 90$ nodes and polynomial degree $p = 8$ where used. The total number of centers inside the unit disc domain is $N_c = 643$ in the first two cases and $N_c = 652$ in the unfitted case.

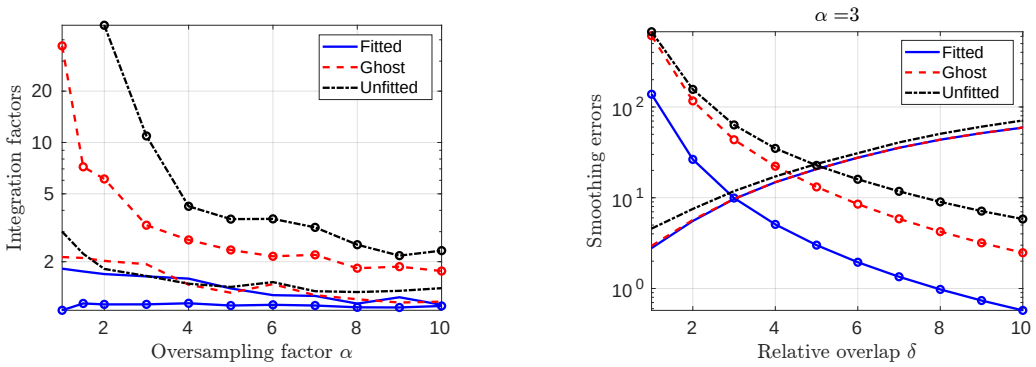

Figure 6. To the left, we show the integration error factors $(1+\tau_0)^{1/2}$ and $1/(1-\tau_a)^{1/2}$ (marked with ○), and to the right, we show the smoothing error factors $(1+\eta_0)^{1/2}$ and $(1+\eta_a)^{1/2}$ (marked with ○) for the three types of approximations.

are needed to suppress errors. On the other hand, the stability factors in the error estimate become larger when nodes are placed outside [20], since we do not control the outside solution. The smoothing error can be reduced by increasing the stencil size [18], but is otherwise fixed, while the integration error is reduced (at a cost) by increasing the oversampling. It was shown in [17] that boundary-fitted oversampled RBF–FD performed better than collocated boundary-fitted RBF–FD in terms of time to achieve a given accuracy.

Acknowledgments

Larsson and Michael were supported by the Swedish Research Council under grant number 2020-03488.

References

[1] Tolstykh A I 2002 *Proc. 16th IMACS World Congress on Scientific Computation, Applied Mathematics and Simulation* (Lausanne, Switzerland)
[2] Lee C K, Liu X and Fan S C 2003 *Comput. Mech.* **30** 396–409
[3] Shu C, Ding H and Yeo K 2003 *Comput. Methods Appl. Mech. Engrg.* **192** 941–954
[4] Tolstykh A I and Shirobokov D A 2003 *Comput. Mech.* **33** 68–79
[5] Šarler B and Vertnik R 2006 *Comput. Math. Appl.* **51** 1269–1282
[6] Vertnik R and Šarler B 2006 *Internat. J. Numer. Methods Heat Fluid Flow* **16** 617–640
[7] Wright G B and Fornberg B 2006 *J. Comput. Phys.* **212** 99–123
[8] Fornberg B and Flyer N 2015 *A primer on radial basis functions with applications to the geosciences* (Philadelphia, PA: SIAM)
[9] Flyer N, Barnett G A and Wicker L J 2016 *J. Comput. Phys.* **316** 39–62
[10] Flyer N, Fornberg B, Bayona V and Barnett G A 2016 *J. Comput. Phys.* **321** 21–38
[11] Bayona V, Flyer N, Fornberg B and Barnett G A 2017 *J. Comput. Phys.* **332** 257–273
[12] Bayona V, Flyer N and Fornberg B 2019 *J. Comput. Phys.* **380** 378–399
[13] Bayona V 2019 *Comput. Math. Appl.* **77** 2337–2353
[14] Larsson E, Mavrič B, Michael A and Pooladi F 2022 *Dolomites Res. Notes Approx.* **15** 78–95
[15] Chinchapatnam P P, Djidjeli K, Nair P B and Tan M 2009 *Proc. Inst. Mech. Eng. M J. Eng. Marit. Environ.* **223** 275–290
[16] Shankar V and Fogelson A L 2018 *J. Comput. Phys.* **372** 616–639
[17] Tominec I, Larsson E and Heryudono A 2021 *SIAM J. Sci. Comput.* **43** A1441–A1471
[18] Tominec I, Nazarov M and Larsson E 2021 Stability estimates for radial basis function methods applied to time-dependent hyperbolic PDEs (*Preprint* `2110.14548`)
[19] Tominec I and Nazarov M 2023 *J. Sci. Comput.* **94** 14, 31
[20] Tominec I and Breznik E 2021 *J. Comput. Phys.* **436** 110283, 24
[21] Tominec I, Villard P F, Larsson E, Bayona V and Cacciani N 2022 *J. Comput. Phys.* **469** 111496, 17
[22] Persson P O and Strang G 2004 *SIAM Rev.* **46** 329–345