



High order difference schemes using the Local Anisotropic Basis Function Method

DOI:
[10.1016/j.jcp.2020.109549](https://doi.org/10.1016/j.jcp.2020.109549)

Document Version
Accepted author manuscript

[Link to publication record in Manchester Research Explorer](#)

Citation for published version (APA):
King, J. R. C., Lind, S. J., & Nasar, A. (2020). High order difference schemes using the Local Anisotropic Basis Function Method. *Journal of Computational Physics*, 109549. <https://doi.org/10.1016/j.jcp.2020.109549>

Published in:
Journal of Computational Physics

Citing this paper
Please note that where the full-text provided on Manchester Research Explorer is the Author Accepted Manuscript or Proof version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version.

General rights
Copyright and moral rights for the publications made accessible in the Research Explorer are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Takedown policy
If you believe that this document breaches copyright please refer to the University of Manchester's Takedown Procedures [<http://man.ac.uk/04Y6Bo>] or contact uml.scholarlycommunications@manchester.ac.uk providing relevant details, so we can investigate your claim.



High order difference schemes using the Local Anisotropic Basis Function Method

J. R. C. King,* S. J. Lind, and A. Nasar

Department of Mechanical, Aerospace and Civil Engineering, The University of Manchester, Manchester, UK

(Dated: May 12, 2020)

Mesh-free methods have significant potential for simulations in complex geometries, as the time consuming process of mesh-generation is avoided. Smoothed Particle Hydrodynamics (SPH) is the most widely used mesh-free method, but suffers from a lack of consistency. High order, consistent, and local (using compact computational stencils) mesh-free methods are particularly desirable.

Here we present a novel framework for generating local high order difference operators for *arbitrary* node distributions, referred to as the Local Anisotropic Basis Function Method (LABFM). Weights are constructed from linear sums of anisotropic basis functions (ABFs), chosen to ensure exact reproduction of polynomial fields up to a given order. The ABFs are based on a fundamental Radial Basis Function (RBF), and the choice of fundamental RBF has small effect on accuracy, but influences stability. LABFM is able to generate high order difference operators with compact computational stencils (4^{th} order with $\mathcal{N} \approx 25$ nodes, 8^{th} order with $\mathcal{N} \approx 60$ nodes in two dimensions). At domain boundaries (with incomplete support) LABFM automatically provides one-sided differences of the same order as the internal scheme, up to 4^{th} order. We use the method to solve elliptic, parabolic and mixed hyperbolic-parabolic partial differential equations (PDEs), showing up to 8^{th} order convergence. The inclusion of hyperviscosity is straightforward, and can effectively provide stability when solving hyperbolic problems.

LABFM is a promising new mesh-free method for the numerical solution of PDEs in complex geometries. The method is highly scalable, and for Eulerian schemes, the computational efficiency is competitive with RBF-FD for a given accuracy. A particularly attractive feature is that in the low order limit, LABFM collapses to Smoothed Particle Hydrodynamics (SPH), and there is potential for Arbitrary Lagrangian-Eulerian schemes with natural adaptivity of resolution and accuracy.

I. INTRODUCTION

The numerical solution of partial differential equations (PDEs) is key to many branches of science, and the calculation of the spatial derivatives of a field, based on the knowledge of the value of that field at a discrete set of points, is key. More accurate approximations lead to more accurate results. Approximations which converge at high order are attractive, yielding equivalent accuracy at lower resolution, and hence lower computational cost. Mesh-based methods have long dominated numerical simulations, and for simple problem geometries, can be made extremely accurate (e.g. tenth order finite differences (FD) [1], or exponentially convergent spectral methods [2]). For complex geometries, mesh-based methods suffer two main drawbacks. Firstly, the symmetries which enable efficient and highly accurate approximations no longer apply. Global spectral methods are no longer possible, and high order finite differences are only possible where an orthogonal mesh can be fitted to the problem. Secondly, the process of mesh generation is complex and time-consuming, and accuracy can become highly dependent on the quality of the mesh. Mesh-free methods have significant potential, as discretisation of the domain becomes extremely simple, and automation of the process to create a node distribution with the desired properties (e.g. fitting the boundaries and satisfying resolution criteria) is relatively straightforward. There is no need to generate and store information on inter-node connectivity. In the context of computational fluid dynamics, mesh-free methods also have the advantage that topological changes in the solution (i.e. wave overtopping) which would lead to singularities in mesh based methods, may be handled easily, provided the method is based in an appropriate frame of reference. A number of mesh-free methods have gained traction in recent decades, and for a broad overview of mesh-free methods, we refer the reader to [3, 4]. Below we provide a brief review of the methods most relevant to the present work. In the following, we use h to denote the characteristic length scale of a computational stencil, \mathcal{N} to denote the typical number of nodes or particles in a stencil, and N to denote the total number of computational nodes.

The Generalized Finite Difference Method (GFDM) [5] is a mesh-free extension of the finite difference method, for arbitrary node distributions. A linear system is solved for each node to obtain a moving least squares (MLS) approximation of a function and its derivatives, based on a compact stencil, from which weights for a local discrete gradient operator are calculated. The condition of the linear systems has a dependence on the node distribution [6], and generally the method is applied only up to second order [7, 8]. Recently Trask et al. [9] has developed a generalised

* jack.king@manchester.ac.uk

moving least squares (GMLS) method, in which a polynomial reconstruction is used to obtain high order approximate derivatives. The method has been extended to include variable resolution, and applied to Stokes flow in complex geometries [10], demonstrating fourth order convergence.

Smoothed Particle Hydrodynamics is a meshfree method, originally developed for astrophysical simulations [11, 12], and adapted with substantial success to simulations of incompressible flows, free surface flows, fluid structure interactions, and solid mechanics [13]. Fluid properties and their spatial derivatives are approximated based on weighted sums of the properties at neighbouring particles, with weights determined from a smoothing kernel, and its derivatives. In [11] smoothing kernels had a Gaussian form, although in modern SPH they are typically polynomials with compact support [14]. The continuous form of SPH has a theoretical convergence rate of 2 (in h) [15], provided \mathcal{N} is held constant (for typical smoothing kernels), but in practice this drops to typically ~ 1 due to particle disorder [16]. In fact, a simple error analysis of discrete SPH shows that derivative approximations are truly zero order in h (and Laplacians even divergent, depending on the formulation), although there is generally a range of resolutions and particle distributions for which convergence can be demonstrated [15]. The convergence rate of SPH is dependent on \mathcal{N} , and this dependence is influenced by the choice of smoothing kernel. For the widely used Wendland C2 kernel (e.g. [17]), the method approaches the theoretical convergence rate (in h) of 2 with $\mathcal{N}^{-5/2}$ (i.e. the total error is $\mathcal{O}(h^2) + \mathcal{O}(\mathcal{N}^{-5/2})$). A number of corrections to SPH have been devised, including by Bonet and Lok [18], whose relatively cheap correction (which requires inversion of a square matrix of rank μ , where μ is the number of spatial dimensions) provides first order consistency.

Higher order consistency corrections for SPH have been devised, generally based on eliminating errors via the solution of N small linear systems, and many of these corrections have become separate methods in their own right. The Reproducing Kernel Particle Method (RKPM) [19, 20], and corrective SPH (CSPH) [3] improve the accuracy of SPH approximations by introducing a polynomial factor in the kernel function chosen to ensure consistency. RKPM has gained significant popularity in solid mechanics [3], and is most commonly employed in a Galerkin formulation. Chen et al. [21] and Chen and Beraun [22] developed the Corrective Smoothed Particle Method (CSPM), in which a Taylor series expansion of the unknown function was used to derive first order consistent interpolations and derivatives approximations. Zhang and Batra [23] and Liu et al. [24] independently developed a correction to SPH interpolation, termed the Finite Particle Method (FPM), in which higher order was achieved by including second and higher order kernel derivatives, and solving a linear system to obtain approximations to a function and its gradient. Results were presented for one- and two-dimensional problems, up to 3rd order, although the consistency of the approximation deteriorated in the vicinity of boundaries. Liu and Liu [25] proposed a variation of CSPM, where a function and all its derivatives (to a given order) are calculated simultaneously. This method provides consistency at boundaries, where computational stencils lack complete support. Asprone et al. [26] developed a modified FPM, with the kernel modified by multiplication with a polynomial to eliminate certain moments, and in [27] extended the idea by replacing the kernel and its derivatives with a series of monomials, demonstrating second order convergence in two dimensions. Building on [25], Sibilla [28] presented a second order correction, involving the solution of a rank 5 (in two dimensions) linear system, to eliminate low order errors. The method of Sibilla [28] provides a consistency correction for both gradient and Laplacian operators, yielding convergence rates which do not deteriorate at very fine resolutions: the order collapses only when machine precision errors start to dominate. Recent work by Lind and Stansby [29] has shown that higher than second order in h may be achieved in SPH by modifying the SPH kernel to eliminate certain moments (in the continuous form). Convergence up to sixth order has been demonstrated for Eulerian (with the particles fixed in space) schemes with these high order kernels, and the approach has been successfully extended to semi-Eulerian schemes [30]. Whilst this approach is promising, zero-order (in h) error terms remain for disordered particle distributions, although in practice there exists a range of resolutions for which high order convergence may be observed. This range depends on how well the discrete sums over neighbour particles approximate continuous integrals. Extending the range of resolutions for which high order is observed may be achieved either by enforcing a certain degree of uniformity and isotropy on the particle distribution, or by use of a modified Gaussian kernel combined with an extremely large computational stencil. With $\mathcal{N} \approx 450$ nodes in two dimensions, corresponding to 12 particles across the support radius, the discretisation error is $< 10^{-16}$ for Gaussian kernels [31]. In practice, the range of resolutions for which high order is observed is dependent on (for a 4th order kernel) the ratio of the third and fifth discrete moments of the kernel, and the solution itself. The computational cost per particle per time-step is at best $\mathcal{O}(\mathcal{N})$ FLOPs (Floating Point Operations), and so whilst the latter approach can yield convergence to machine precision, it increases the cost over compactly supported kernels, which typically require $\mathcal{N} \leq 50$ in two dimensions.

A feature shared by GFDM, GMLS and the above mentioned SPH-derived methods is that interpolation involves summation over all \mathcal{N} nodes in the computational stencil of a local kernel function. Radial Basis Function (RBF) methods, which originated in the field of cartography as an interpolation method for multivariate scattered node data [32], and were first applied to the solution of PDEs in [33, 34], provide a different approach. An RBF is located at each computational node, with each RBF weighted such that the sum of *all* N RBFs provides exact interpolation at the nodes. Early RBF methods were global, with weights obtained by the solution of a dense global linear system

(of size $N \times N$). Global RBF methods can achieve exponential convergence and spectral accuracy [35], although the cost of the global approach is significant, at $\mathcal{O}(N^2)$ FLOPs per time-step [36]. The linear systems in global RBF methods also suffer from poor conditioning. To overcome these shortfalls a local version of the RBF method - the RBF finite difference (RBF-FD) method - was developed independently by Shu et al. [37], Cecil et al. [38], Wright [39], where each node only interacts with the nearest \mathcal{N} nodes (and generally $\mathcal{N} \ll N$). In RBF-FD, the approximations of a function and its derivative are constructed from a set of weights obtained by solving a small linear system at every node. The method is extremely fast, scalable and accurate, although the exponential convergence and spectral accuracy of the global method is lost. Provided $\mathcal{N} \ll N$, the cost is $\mathcal{O}(N)$ FLOPs per time-step, with the costs of assembling and solving the local linear systems relegated to the preprocessing stage. In RBF-FD, the order of convergence is dependent on the stencil size, and (in two-dimensions) approximately $\sqrt{\mathcal{N}}$. Recent work has shown that the addition of polynomials to the local linear systems in RBF-FD can control the order of convergence, and improve the conditioning of the linear systems [40, 41]. A good review of both global RBF and RBF-FD methods is given by Fornberg and Flyer [35].

In this paper we present a novel technique for calculating high order difference approximations, which we term the Local Anisotropic Basis Function method (LABFM). Local discrete operators approximating the spatial derivatives are constructed from a linear combination of anisotropic basis functions (ABFs). The weights for the linear sum of ABFs are determined to enforce specific error terms to be zero, ensuring exact reproduction of polynomial fields up to the desired order, on arbitrary node distributions. In this work we demonstrate up to 8th order convergence on disordered node distributions.

The remainder of the paper is set out as follows. In Section II we introduce our notation, a general discrete operator, and an expression for the error therein. In Section III we present our method for calculating weights for high order difference operators, based on a series of Anisotropic Basis Functions (ABFs), derived from a fundamental RBF. In Section IV we present numerical results using our method focusing on the accuracy and stability. In Section V we demonstrate the ability of our method to solve a range of prototype PDEs. Section VI is a summary of conclusions. In Appendix A we provide details of the ABFs used in this work.

II. A GENERAL DISCRETE OPERATOR

Consider a set of N nodes distributed in a domain Ω , with $\mathbf{r}_i = (x_i, y_i)$ the position vector of node i . Throughout this paper we consider two dimensional problems for simplicity of exposition, although the method could be easily extended to higher dimensional problems (at computational cost). *Note that the analysis in this section is for arbitrary node distributions.* A characteristic length scale of the distribution is δr , where typically $N\delta r^2 = V_\Omega$, with V_Ω the volume of the domain. The value of some field (\cdot) at node i is $(\cdot)_i$. When calculating derivatives, we use stencils with finite size with characteristic width h (in the present work, all stencils have a radius of $2h$). For any node i , we refer to all nodes j within the stencil of i as neighbours of i . \mathcal{N}_i is the number of neighbours of i . Sums over j (e.g. \sum_j) are over all $j \in \mathcal{N}_i$. In this paper, we consider the case where $h/\delta r$ is uniform and constant, and hence also \mathcal{N}_i is approximately uniform and constant. We use H as a characteristic length scale of the domain, or of some scalar or vector field, and then h/H is a non-dimensional measure of the effective resolution. We denote $(\cdot)_{ij} = (\cdot)_i - (\cdot)_j = -(\cdot)_{ji}$, and where the subscript ji appears after a function, the difference is applied to the function arguments. We define a vector of monomials

$$\mathbf{X} = \left[x, y, \frac{x^2}{2}, xy, \frac{y^2}{2}, \frac{x^3}{6}, \frac{x^2y}{2}, \frac{xy^2}{2}, \frac{y^3}{6}, \frac{x^4}{24}, \dots \right]^T, \quad (1)$$

noting that $\mathbf{X}_{ji} = [x_{ji}, y_{ji}, x_{ji}^2/2, \dots]^T$ (i.e. $\mathbf{X}_{ji} \neq \mathbf{X}_j - \mathbf{X}_i$). We also define a vector operator of partial derivatives as

$$\mathbf{D}(\cdot) = \left[\frac{\partial(\cdot)}{\partial x}, \frac{\partial(\cdot)}{\partial y}, \frac{\partial^2(\cdot)}{\partial x^2}, \frac{\partial^2(\cdot)}{\partial x \partial y}, \frac{\partial^2(\cdot)}{\partial y^2}, \frac{\partial^3(\cdot)}{\partial x^3}, \frac{\partial^3(\cdot)}{\partial x^2 \partial y}, \frac{\partial^3(\cdot)}{\partial x \partial y^2}, \frac{\partial^3(\cdot)}{\partial y^3}, \frac{\partial^4(\cdot)}{\partial x^4}, \dots \right]^T. \quad (2)$$

With this notation, the multi-dimensional Taylor expansion of (\cdot) about i may be expressed concisely as

$$(\cdot)_j = (\cdot)_i + \mathbf{X}_{ji} \cdot \mathbf{D}(\cdot)|_i, \quad (3)$$

in which the subscript i in the final term on the RHS indicates \mathbf{D} is evaluated at \mathbf{r}_i . We define a general discrete operator, operating on node i :

$$L_i^d(\cdot) = \sum_j (\cdot)_{ji} w_{ji}^d \quad (4)$$

where d identifies the specific partial derivative(s) being approximated, and w_{ji}^d are the set of weights that, at this stage we require to be functions of \mathbf{r}_{ji} (i.e. $w_{ji}^d = w^d(\mathbf{r}_{ji})$) and differentiable. Equation (4) approximates $\mathbf{C}^d \cdot \mathbf{D}(\cdot)|_i$ where \mathbf{C}^d is a vector pointing to the appropriate derivatives for a given d . Usually we are interested in estimating the gradients and Laplacian of a function, for which cases we set $d = x$, $d = y$, and $d = L$, giving

$$\mathbf{C}^d = \begin{cases} \begin{bmatrix} 1, 0, 0, 0, 0, 0, 0, 0, 0 \dots \end{bmatrix}^T & \text{if } d = x \\ \begin{bmatrix} 0, 1, 0, 0, 0, 0, 0, 0, 0 \dots \end{bmatrix}^T & \text{if } d = y \\ \begin{bmatrix} 0, 0, 1, 0, 1, 0, 0, 0, 0 \dots \end{bmatrix}^T & \text{if } d = L. \end{cases} \quad (5)$$

The choice to use $(\cdot)_{ji}$ in (4) (as opposed to $(\cdot)_j$) allows inter-node weight functions with a singularity at w_{ii}^d . We analyse the error in L_i^d by substituting the Taylor expansion (3) into (4), obtaining

$$L_i^d(\cdot) = \sum_j \mathbf{X}_{ji} \cdot \mathbf{D}(\cdot)|_i w_{ji}^d, \quad (6)$$

which when expanded is

$$\begin{aligned} L_i^d(\cdot) = & \frac{\partial(\cdot)}{\partial x} \Big|_i \sum_j x_{ji} w_{ji}^d + \frac{\partial(\cdot)}{\partial y} \Big|_i \sum_j y_{ji} w_{ji}^d + \frac{\partial^2(\cdot)}{\partial x^2} \Big|_i \sum_j \frac{x_{ji}^2}{2} w_{ji}^d \\ & + \frac{\partial^2(\cdot)}{\partial x \partial y} \Big|_i \sum_j x_{ji} y_{ji} w_{ji}^d + \frac{\partial^2(\cdot)}{\partial y^2} \Big|_i \sum_j \frac{y_{ji}^2}{2} w_{ji}^d + \frac{\partial^3(\cdot)}{\partial x^3} \Big|_i \sum_j \frac{x_{ji}^3}{6} w_{ji}^d + \frac{\partial^3(\cdot)}{\partial x^2 \partial y} \Big|_i \sum_j \frac{x_{ji}^2 y_{ji}}{2} w_{ji}^d \\ & + \frac{\partial^3(\cdot)}{\partial x \partial y^2} \Big|_i \sum_j \frac{x_{ji} y_{ji}^2}{2} w_{ji}^d + \frac{\partial^3(\cdot)}{\partial y^3} \Big|_i \sum_j \frac{y_{ji}^3}{6} w_{ji}^d + \frac{\partial^4(\cdot)}{\partial x^4} \Big|_i \sum_j \frac{x_{ji}^4}{24} w_{ji}^d + \dots \quad (7) \end{aligned}$$

We refer to the sums on the RHS as the moments of w^d . The m -th moments are denoted $B^{d,m}$, where m is the total order of the monomials within the sum. There are two first moments, three second moments, four third moments, and so on. The vector of moments is defined

$$\mathbf{B}_i^d = \sum_j \mathbf{X}_{ji} w_{ji}^d. \quad (8)$$

To ensure that L^d is non-divergent, we require the terms in (7) of order k to scale with h^{k-l} , where l is the order of the derivative being approximated. For first derivatives, this can be achieved by ensuring that w^d scales with $1/h$. If this condition is satisfied, $B^{d,m} \propto h^{m-1}$, and L^d is zero-order in h for first derivatives. The equivalent condition for second derivatives requires that w^d scales with $1/h^2$, and that the first moments of w^d are zero. Finite difference schemes satisfy these conditions, as do first derivative approximations in SPH, which are obtained by setting (e.g. for $\partial(\cdot)/\partial x$)

$$w_{ji}^x = \frac{\partial W_{ij}}{\partial x} \delta V, \quad (9)$$

where W is the SPH smoothing kernel, and δV is a nominal node volume. The use of a normalized SPH kernel (i.e. $\int_V W dV = 1$, where V is the support volume of the kernel) ensures that the approximation is zero order. A zero-order (and non-zero error) operator is not particularly useful. By manipulation of the specific form of w^d , the magnitudes of moments corresponding to the desired derivative may be brought closer to unity, and the magnitude of other moments may be reduced towards zero, such that the method shows higher (than zero) order convergence over a range of resolutions. In the next section we present a method for doing this and achieving arbitrarily high order difference operators.

III. THE LOCAL ANISOTROPIC BASIS FUNCTION METHOD

A. The elimination of error terms

We now present a method of specifying w^d such that the error in L^d is of arbitrarily high order. We set w^d equal to the weighted sum of a series of Anisotropic Basis Functions (ABFs) $W_{ji} = W(\mathbf{r}_{ji}/h)$, writing

$$w_{ji}^d = \mathbf{W}_{ji} \cdot \boldsymbol{\Psi}_i^d = W_{ji}^1 \Psi_{i,1}^d + W_{ji}^2 \Psi_{i,2}^d + W_{ji}^3 \Psi_{i,3}^d + W_{ji}^4 \Psi_{i,4}^d + \dots, \quad (10)$$

where the vector of basis functions is $\mathbf{W}_{ji} = [W_{ji}^1, W_{ji}^2, W_{ji}^3 \dots]^T$ and the vector of weights is $\Psi_i^d = [\Psi_{i,1}^d, \Psi_{i,2}^d, \Psi_{i,3}^d \dots]^T$. The basis functions are anisotropic, in that they depend on \mathbf{r}_{ji} (as opposed to the dependence on $|\mathbf{r}_{ji}|$ of RBFs). Substituting (10) into (8) gives

$$\mathbf{B}_i^d = \sum_j \mathbf{X}_{ji} \mathbf{W}_{ji} \cdot \Psi_i^d, \quad (11)$$

which may be easily reformulated as a linear system

$$\mathbf{M}_i \Psi_i^d = \mathbf{B}_i^d, \quad (12)$$

in which

$$\mathbf{M}_i = \sum_j \mathbf{X}_{ji} \otimes \mathbf{W}_{ji}. \quad (13)$$

We obtain the weights Ψ_i^d by specifying desired values of each moment, replacing \mathbf{B}_i in (12) with \mathbf{C}^d and solving the resulting system

$$\mathbf{M}_i \Psi_i^d = \mathbf{C}^d. \quad (14)$$

Having solved (14) we use (10) to obtain the weights w_{ji}^d .

The above analysis imposes no constraint on the lengths of \mathbf{X} , \mathbf{W} , \mathbf{C} , and the size of \mathbf{M} , although for consistency between (13) and (14), we require \mathbf{X} , \mathbf{W} and \mathbf{C} to be of equal length, and \mathbf{M} square. In practice they are finite, and we use only the first p elements from \mathbf{X} and \mathbf{W} to construct and solve (14). We note here the useful relation between the number of elements p , and the order k of the p -th element of \mathbf{X} :

$$p = \frac{k^2 + 3k}{2} = 2, 5, 9, 14, 20, 27 \dots \text{ for } k = 1, 2, 3, 4, 5 \dots, \quad (15)$$

which is valid for two-dimensional problems. If the method were applied to three-dimensional problems p would be the sum of all triangular numbers from 2 to $k + 1$:

$$p = \sum_{m=2}^{m=k+1} m(m+1)/2 = 3, 9, 19, 34, 55 \dots \text{ for } k = 1, 2, 3, 4, 5 \dots \quad (16)$$

When (14) is solved with \mathbf{M}_i having size $p \times p$, exact polynomial reconstruction of order k is ensured. That is to say, the leading order (in h) error terms in L_i^d scale with h^{k-l+1} , where l is the order of the derivative approximated by L_i^d . For first derivatives, $l = 1$ and the leading order error scales with h^k . For second derivatives, $l = 2$ and the leading order error scales with h^{k-1} . The matrices \mathbf{M}_i and the resulting w^d are dependent on the anisotropy of the node distribution. However, in setting w^d via (10) and the solution of (14), the accuracy of L_i^d is maintained, *provided the node distribution adequately samples the ABFs*. We discuss this criteria further in the following sections. Throughout this work we use k - the order of polynomial reconstruction - to identify the order of the scheme used. The matrix \mathbf{M}_i is the same for all operators (different d), and so we calculate each \mathbf{M}_i once, then solve (14) for each d to obtain the required operators. Once (14) has been solved, we calculate the weights w_{ji}^d and store them. When the method is used for solving PDEs in an Eulerian framework, the bulk of the cost is at startup, the cost of applying the operators L_i^d is low - the run-time efficiency is comparable with finite difference methods - and the method is highly scalable.

B. Constructing appropriate ABFs

To obtain an appropriate set of ABFs, we start with the requirement that each ABF in \mathbf{W}_{ji} approximates a different partial derivative, so that if $w_{ji}^d = \mathbf{W}_{ji} \cdot \mathbf{C}^d$

$$L_i^d(\cdot) = \sum_j (\cdot)_{ji} w_{ji}^d = \sum_j (\cdot)_{ji} \mathbf{W}_{ji} \cdot \mathbf{C}^d \approx \mathbf{D}(\cdot)|_i \cdot \mathbf{C}^d, \quad (17)$$

for any d . This approximation need not be very accurate. We define the inner product as

$$\langle f(\mathbf{r}_{ji}), g(\mathbf{r}_{ji}) \rangle = \frac{1}{V_i} \int_{V_i} f(\mathbf{r}_{ji}) g(\mathbf{r}_{ji}) dV_j, \quad (18)$$

where f and g are any two functions of \mathbf{r} . Note that the moments \mathbf{B}_i^d in (8) are constructed from the discrete form of $\langle \mathbf{X}_{ji}, w_{ji}^d \rangle$. We denote the m^{th} element of \mathbf{X}_{ji} as X_{ji}^m , and the n^{th} ABF in \mathbf{W}_{ji} as W_{ji}^n .

If the ABFs are chosen such that $\langle W^m, X^m \rangle$ is independent of h for all $m \leq p$, where p is the number of ABFs used, the approximation requirement (17) will be satisfied. Given the elements of \mathbf{X}_{ji} scale with h , h^2 , h^3 , h^4 and so on for $p \in [1, 2]$, $p \in [3, 5]$, $p \in [6, 9]$, and $p \in [10, 14]$ respectively, the ABFs must scale with h^{-1} , h^{-2} , h^{-3} , h^{-4} and so on. This ensures that the diagonal of \mathbf{M} is independent of resolution. This requirement may be satisfied by constructing the ABFs from the partial derivatives of some appropriately scaled fundamental RBF $W^0 = W^0(|\mathbf{r}_{ji}|)$, according to

$$\mathbf{W}_{ji} = \mathbf{D}(W^0)|_{\mathbf{r}_{ji}} \quad (19)$$

Taking W^0 as a very simple RBF - a cone - defined by

$$W^0(q) = \frac{3}{4\pi} \left(1 - \frac{q}{2}\right), \quad (20)$$

where $q = |\mathbf{r}_{ji}|/h = r_{ji}/h$, the radial derivative of W^0 , $dW^0/dr_j = -3/8\pi h$, and higher radial derivatives are zero ($\partial^k W^0/\partial r_j^k = 0, \forall k \geq 2$). We obtain the vector of ABFs by applying the operator \mathbf{D} to W^0 :

$$\begin{aligned} \mathbf{W}_{ji} = \mathbf{D}(W^0)|_{\mathbf{r}_{ji}} = & \frac{-3}{8\pi h} \left[\frac{x}{r}, \frac{y}{r}, \frac{y^2}{r^3}, \frac{-xy}{r^3}, \frac{x^2}{r^3}, \frac{-3xy^2}{r^5}, \frac{2x^2y - y^3}{r^5}, \frac{2xy^2 - x^3}{r^5}, \right. \\ & \left. \frac{-3x^2y}{r^5}, \frac{12x^2y^2 - 3y^4}{r^7}, \frac{9xy^3 - 6x^3y}{r^7}, \frac{2x^4 - 11x^2y^2 + 2y^4}{r^7}, \frac{9x^3y - 6xy^3}{r^7}, \frac{12x^2y^2 - 3x^4}{r^7} \dots \right]^T, \quad (21) \end{aligned}$$

in which the subscripts ji have been dropped from x_{ji} , y_{ji} , and r_{ji} for ease of exposition. Figures 2, 3, and 4 show the first, second, and third and fourth order ABFs respectively. We observe that because $dW^0/dr_j|_{q=0} \neq 0$ (i.e. the fundamental RBF has a pointed peak), the ABFs generated from it have singularities at $\mathbf{r}_{ji} = 0$. The first order ABFs (Figure 2) have a removable singularity. The second order ABFs have a simple pole, the third order ABFs have a second order pole, and the order k ABFs have a pole of order $k - 1$.

This approach may be used with an arbitrary fundamental RBF. In this work we investigate four choices of W^0 : the cone, as described above, a quadratic RBF, the Wendland C6 kernel [42], and the Gaussian, details of which are provided in Appendix A. We note here that when we set W^0 to an SPH kernel with $p = 5$, the matrix on the LHS of (14) is the transpose of the matrix obtained and solved in the method of Sibilla [28].

The statement in the previous section, that the node distribution must adequately sample the ABFs used is analogous to stating that the sums from which the elements of \mathbf{M}_i are constructed must adequately approximate continuous integrals:

$$\sum_j X_{ji}^m W_{ji}^n \approx \langle X_{ji}^m, W_{ji}^n \rangle \quad \forall m, n \in [1, p]. \quad (22)$$

In practice this requirement imposes a lower limit on $h/\delta r$. For each value of k , there is a critical value of $h/\delta r$ (and hence \mathcal{N}) below which one or more of the eigenvalues of \mathbf{M}_i has magnitude of machine precision zero, and below which the condition number of \mathbf{M}_i increases significantly (typically by about 10 orders of magnitude). The critical values for $k = \{2, 4, 6, 8\}$ are $h/\delta r_{crit} \approx \{0.75, 1.15, 1.6, 2.1\}$ and in two dimensions $\mathcal{N}_{crit} \approx \{8, 21, 37, 57\}$, and are (on average) independent of disorder of the node distribution. We note that the optimal number of nodes required for polynomial reconstruction of the same order is $\mathcal{N}_{poly} = \{6, 15, 28, 45\}$ in two dimensions. It is clear from Figures 2, 3, and 4, that as the order of ABF increases (higher k), more nodes are required, due to the increasing complexity of the ABFs (higher order poles, higher wavenumbers). When the ABFs in (21) are expressed in local polar coordinates (r_{ji} and θ_{ji} having origin at \mathbf{r}_i), the order k ABFs contain terms of $\cos k\theta$ and $\sin k\theta$. For the stencil to adequately sample the ABFs, the Nyquist sampling criteria must be satisfied. That is, if the domain of support is divided into $2k$ equal segments (each with angle π/k), there is a node in every segment. This is illustrated in Figure 1 for $k = 4$. For $h/\delta r = 1$ (left, red stencil) there is a segment containing no nodes: the 4th order ABFs are undersampled by this stencil. With $h/\delta r = 1.25$ (right, black stencil), all 8 segments contain at least one node, and the stencil satisfies the Nyquist sampling criteria for the 4th order ABFs. This criteria provides a more accurate definition of the word “adequately” use for (22), and the values of $h/\delta r_{crit}$ and \mathcal{N}_{crit} correspond to the minimum values for which the Nyquist criteria is satisfied. The effects of choosing $h/\delta r < h/\delta r_{crit}$ can be seen in the numerical results in Section IV, in particular in the horizontal contours in Figure 9.

Figure 5 illustrates an example of the matrix \mathbf{M}_i for for $k = 8$ with the quadratic ABFs, $h/\delta r = 2.5$, and a range of values of distribution noise $\varepsilon/\delta r$. We see for the uniform Cartesian distribution, \mathbf{M} has a clear structure. The entries

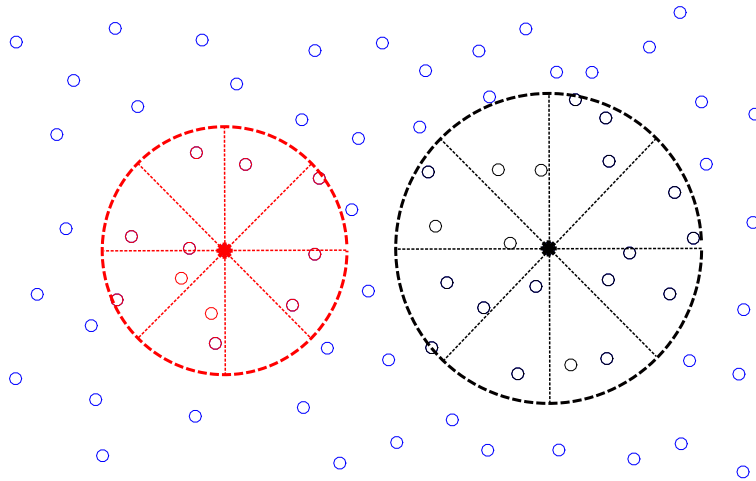


FIG. 1. Two example computational stencils. The red stencil (left) corresponds to $h/\delta r = 1$, whilst the black stencil (right) corresponds to $h/\delta r = 1.25$. The thin dashed lines divide the stencil into $2k$ equal segments for $k = 4$.

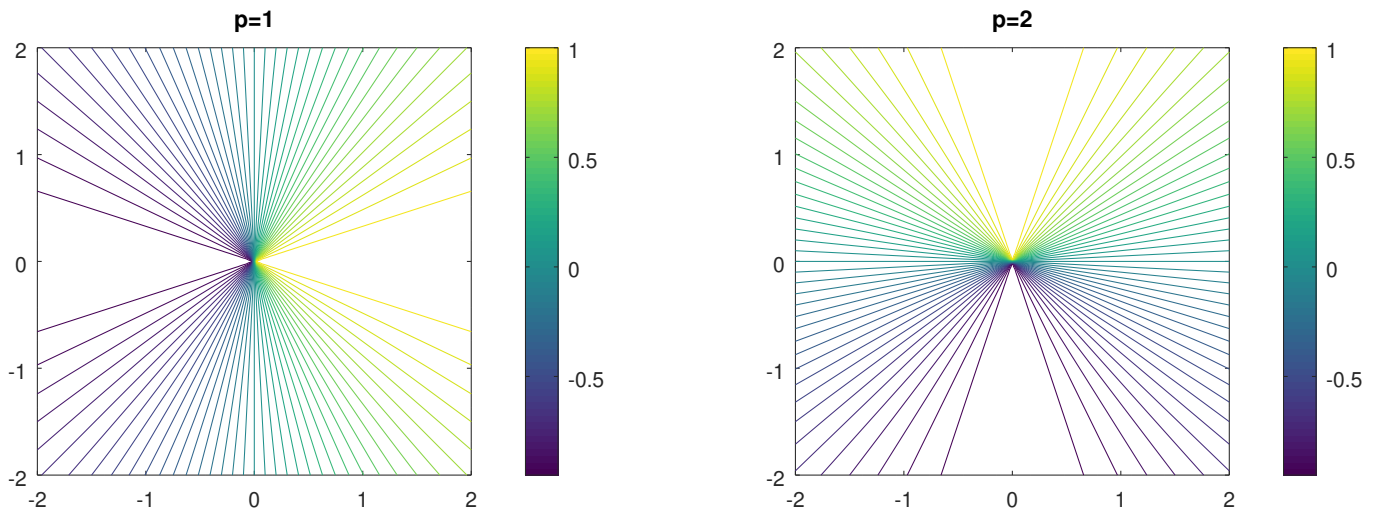


FIG. 2. The first order ABFs, derived from a conic RBF.

coloured in dark blue are zero *to within machine precision*, due to the orthogonality of the odd order ABFs with the even order monomials (and vice versa). The magnitude of elements in top right are small, and bottom left are large, which is consistent with the scaling of \mathbf{X}_{ji} and \mathbf{W}_{ji} with h . For smaller k , the matrices \mathbf{M}_i are just sub-matrices of those illustrated here. We see as the noise is increased, the magnitude of the entries in the bottom left corner increases (the brighter yellow patch), which is a consequence of the undersampling of the highest order ABFs. We investigate the effect of stencil size (\mathcal{N}_i and $h/\delta r$) further in Section IV B below.

IV. CONVERGENCE AND STABILITY

We now analyse the convergence properties of LABFM, and present a brief stability analysis.

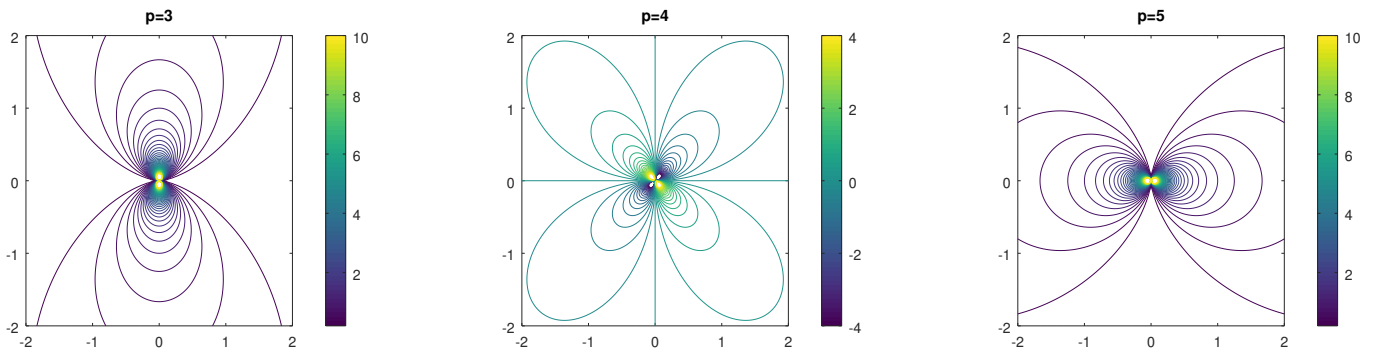


FIG. 3. The second order ABFs, derived from a conic RBF.

A. Convergence

We consider a square domain defined by $(x, y) \in [0, H] \times [0, H]$. We define the test function

$$\phi(\hat{x}, \hat{y}) = 1.0 + (\hat{x}\hat{y})^4 + \sum_{n=1}^6 (\hat{x}^n + \hat{y}^n), \quad (23)$$

where $\hat{x} = x - 0.1453H$ and $\hat{y} = y - 0.16401H$, and set $H = 1$. This offset is psuedo-random, and included to ensure asymmetry in the function, to prevent the masking of errors which could cancel for a symmetric function. Nodes are distributed uniformly (Cartesian grid) with spacing δr , and randomly perturbed. The maximum perturbation distance is ε , and hence we take $\varepsilon/\delta r$ as a measure of the irregularity or noise of the distribution. In terms of fill distance and separation distance as used in the field of scattered data approximation, and defined in [43], $\varepsilon/\delta r = 0$ corresponds to a separation distance of $\delta r/2$ and a fill distance of $\delta r/\sqrt{2}$. For $\varepsilon/\delta r = 0.5$, the separation distance is approximately $0.29\delta r$, and the fill distance approximately $0.86\delta r$. Initially we set $h = 2\delta r$ which, with a support radius of $2h$, yields $\mathcal{N}_i \approx 50$ in two dimensions. An example node distribution and computational stencil is depicted in Figure 6. The stencil indicated in black has full support, and the stencil indicated in red has incomplete support. At boundaries, we create additional “ghost” nodes (not shown in Figure 6) with the same distribution properties in a strip of width $2h$ around the domain, ensuring that all computational nodes have fully supported stencils. We test our method by calculating approximations of the gradients and Laplacian of ϕ for a range of resolutions. As a measure of the error we take the normalised L_2 -norm:

$$L_2\text{-norm}(\cdot)^d = \frac{\left\{ \sum_{i=1}^N [L_i^d(\cdot) - \mathbf{C}^d \cdot \mathbf{D}(\cdot)|_i]^2 \right\}^{\frac{1}{2}}}{\left\{ \sum_{i=1}^N [\mathbf{C}^d \cdot \mathbf{D}(\cdot)|_i]^2 \right\}^{\frac{1}{2}}} \quad (24)$$

Figure 7 shows the L_2 -norm of the error in gradient and Laplacian approximations for node distributions with $\varepsilon/\delta r = 0.5$ (the distribution depicted in Figure 6), for a range of resolutions, and increasing numbers of ABFs contributing to the discrete operator. For gradients, we observe orders of convergence increasing from second to sixth, as we increase k from 2 to 6 ($p = 5$ and $p = 27$ respectively). The order of convergence for gradients is k . For the Laplacian, the order of convergence is $k - 1$, ranging from first to fifth order. For $k = 2$ there is a range of resolutions for which second order convergence is observed in the Laplacian approximation. However, this is because the second order error terms are larger than the first order error terms at these coarse resolutions, and this effect is strongly dependent on the choice of ϕ . For $k > 6$, we require $h/\delta r > 2$. Setting $h/\delta r = 2.5$, with $k = 7$ we find convergence rates of 7 and 6 for gradients and Laplacian’s respectively. With $k = 8$ the errors in gradients are typically 10^{-14} , and do not converge. This is because the 8th order derivatives of ϕ (the highest non-zero derivatives) are constant. For Laplacian’s with $k = 8$ the errors scale with h^{-1} , but with magnitude 10^{-13} for the coarsest resolution. These errors are machine precision errors, accumulated during the construction of \mathbf{M} , and carried through the solution of (14). For other test functions (e.g. a sinusoidally varying ϕ) with $k = 8$ the convergence rates of 8 and 7 for gradients and Laplacians hold. For clarity, we only show results were obtained using the conic ABFs in Figure 7. Quadratic ABFs would yield the same convergence rates, and error magnitudes approximately 32% lower.

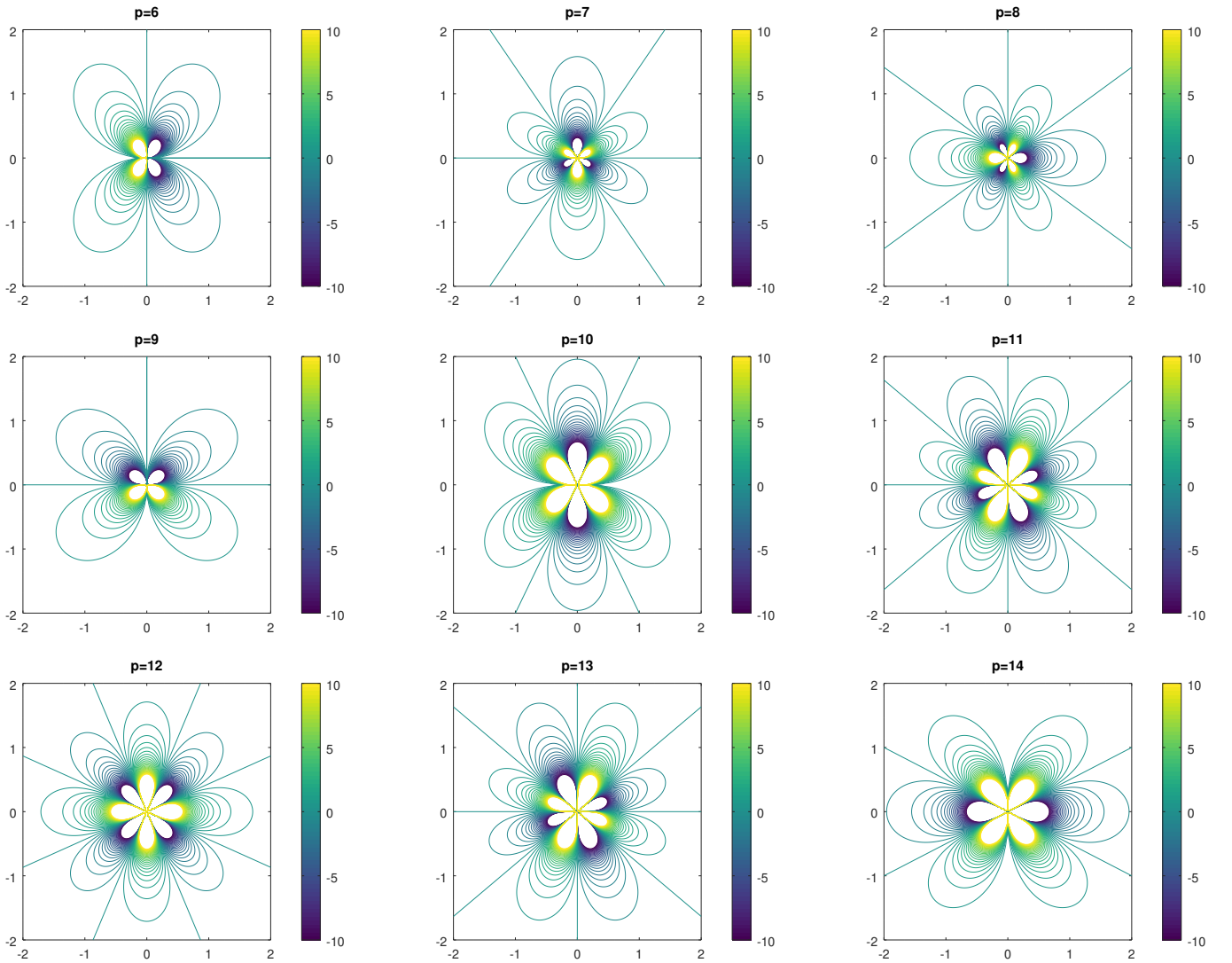


FIG. 4. The third and fourth order ABFs, derived from a conic RBF.

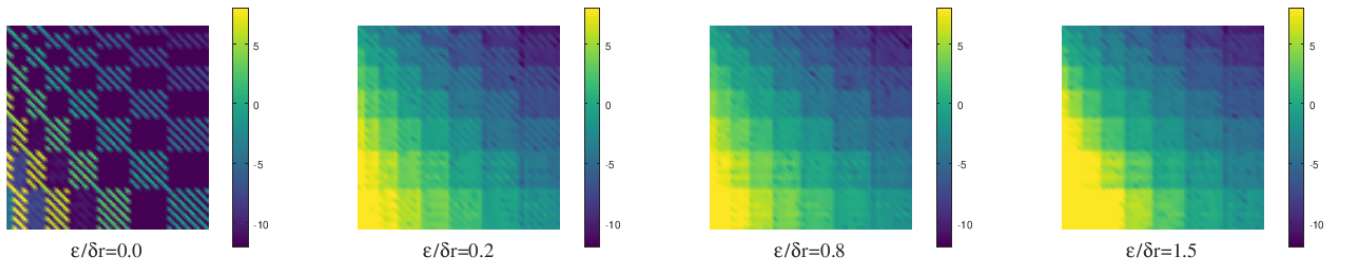


FIG. 5. Illustrations of the matrix \mathbf{M}_i obtained with $k = 8$ ($p = 44$), and the quadratic ABFs, with $h/\delta r = 2/5$, for increasingly noisy node distributions. Colour indicates the base 10 logarithm of the absolute value of each element $\log_{10}(|\mathbf{M}_i|)$.

Figure 8 shows the variation of the L_2 -norm with resolution h/H and the degree of noise in the node distribution $\varepsilon/\delta r$. For all k , we see a maintenance of convergence order (contours in Figure 8 are parallel) for levels of noise below $\varepsilon/\delta r = 0.5$. For $k = 2$ the convergence is maintained up to $\varepsilon/\delta r = 1$ and beyond, whilst as k increases, and convergence breaks down at lower $\varepsilon/\delta r$. The resilience of the method to noisy node distributions is greater for first derivatives (top row in Figure 8) than for second derivatives (bottom row). For first derivatives, the magnitude of the error is also independent of $\varepsilon/\delta r$ as long as convergence is maintained (vertical contours), whilst for second

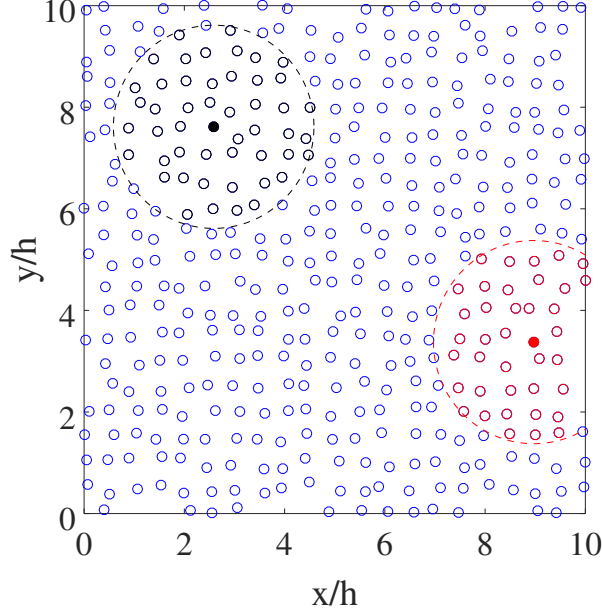


FIG. 6. An example node distribution with $\varepsilon/\delta r = 0.5$, and $h/\delta r = 2$. The dashed circles with radius $2h$ indicate the support domains of two nodes (indicated by filled circles). The computational stencil depicted in black has full support, and the computational stencil depicted in red has incomplete support.

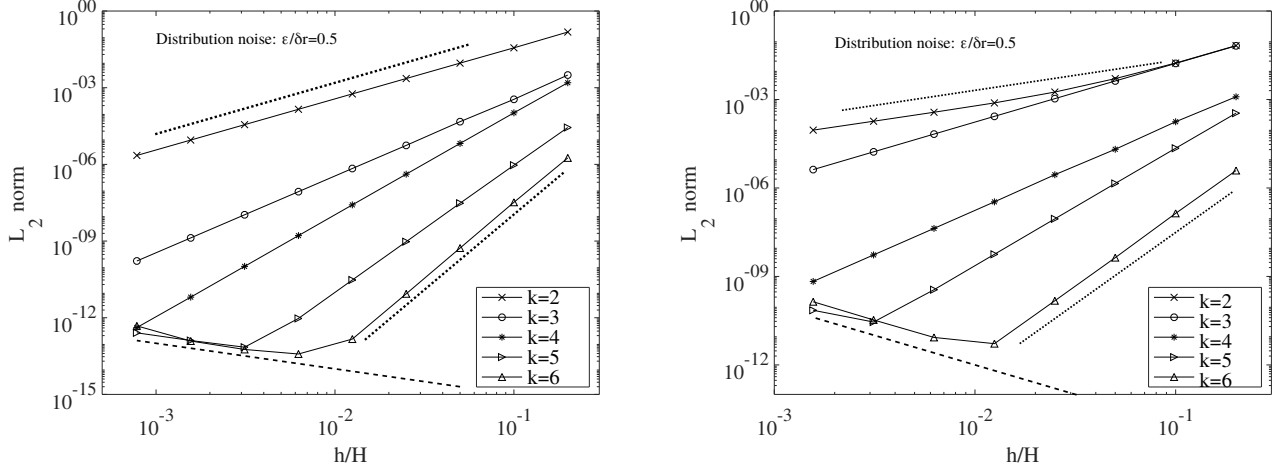


FIG. 7. Error in gradient (left) and Laplacian (right) approximations of (23). The dashed lines represent machine precision errors of $10^{-16}/h$ and $10^{-16}/h^2$ for gradients and Laplacians respectively. The dotted lines show convergence rates of 2 and 6 for gradients, and 1 and 5 for Laplacians.

derivatives, the magnitude of the error increases approximately linearly with increasing $\varepsilon/\delta r$ (contours are diagonal, and approximately straight). This result is intuitive, as all results in Figure 8 were obtained with the same $h/\delta r$, and so the sampling of the ABFs by the node distribution is *relatively* better for $k = 2$ than $k = 6$.

Figure 9 shows the variation of the L_2 -norm with resolution h/H and stencil size $h/\delta r \propto \sqrt{\mathcal{N}}$, as k is increased from 2 to 6. We use $h/\delta r$ as a measure of stencil size because stencils are calculated based on a support radius ($2h$ here) as opposed to the desired stencil number size \mathcal{N} as in RBF methods. For reference, a value of $h/\delta r = 2$ corresponds to $\mathcal{N} \approx 50$ in two dimensions. For all values of k , we see convergence of the L_2 -norm with increasing resolution at large $h/\delta r$ (parallel contours in upper section of plots). However, there is a clear critical value of $h/\delta r$ below which the method is divergent. Below this value, the node distribution within the computational stencil does

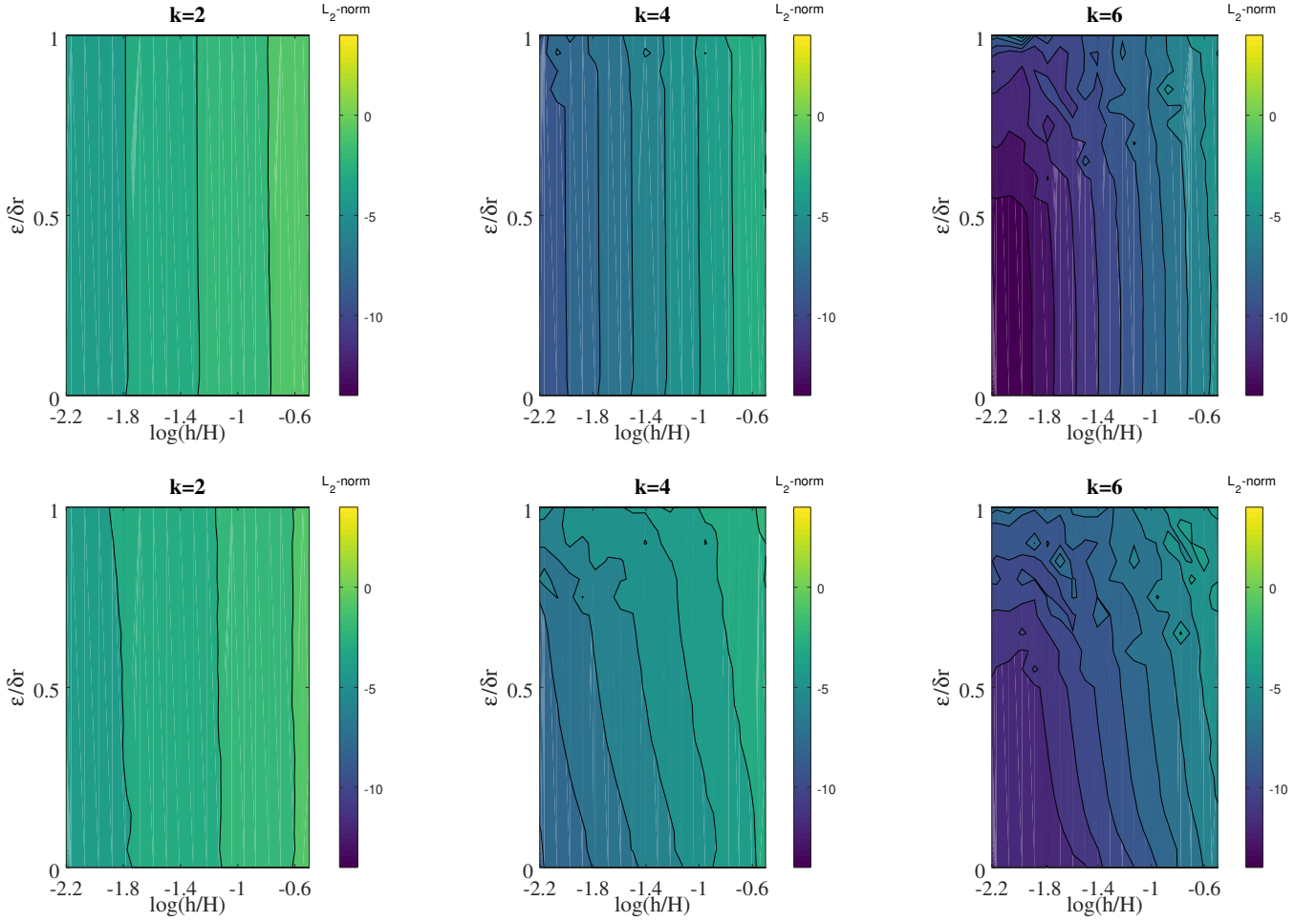


FIG. 8. L_2 -norm when approximating the gradient (upper row) and Laplacian (lower row) of (23), for a range of resolutions and distribution noise, for $k = 2$, $k = 4$, and $k = 6$. The scale on the abscissa is $\log_{10}(h/H)$, and colour indicates the base 10 logarithm of the L_2 -norm. The conic ABFs are used, with $h/\delta r = 2$.

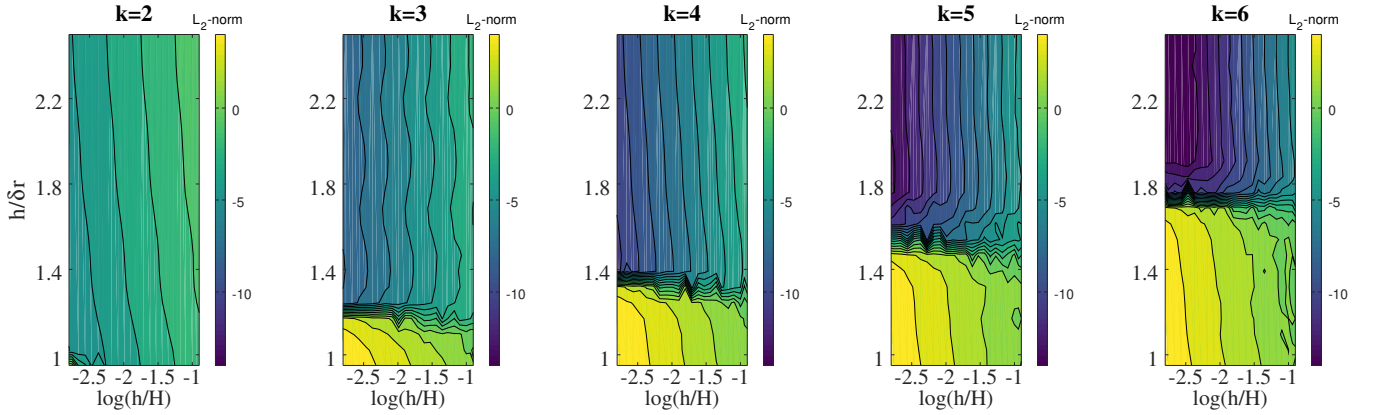


FIG. 9. Contour plots of the L_2 -norm when approximating the gradient of (23), for a range of resolutions and stencil sizes ($h/\delta r \propto \sqrt{N}$), for increasing values of k . The scale on the abscissa is $\log_{10}(h/H)$, and colour indicates the base 10 logarithm of the L_2 -norm. The node distribution is noisy Cartesian, with $\epsilon/\delta r = 0.5$.

not adequately sample the ABFs, leading to ill conditioned matrices \mathbf{M}_i in (14). As discussed in the previous section, this critical value is larger for larger k , because for higher k we include higher order ABFs which have a more complex structure, requiring a greater number of nodes to adequately sample them. In the majority of the remainder of this paper, we use $h/\delta r = 2$, which leads to convergence for $k \leq 6$. However, we could, for low order schemes (e.g. $k = 2$) use a significantly smaller stencils, with $h/\delta r = 1$, reducing computational costs. For $k = 8$, we observe 8th order convergence with $h/\delta r = 2.5$ ($\mathcal{N} \approx 78$ in two dimensions). For a domain with incomplete support at the boundaries, we find the method yields accurate one sided derivatives and Laplacians, provided $h/\delta r$ (and hence, the computational stencil) is large enough at the domain boundaries. For nodes with incomplete support, the critical $h/\delta r$ is larger than in the fully supported case in Figure 9. With this in mind, a potential approach when applying the method to practical problems is to have a region of higher resolution (smaller δr , maintaining $h/\delta r$) in the vicinity of boundaries, to preserve the global convergence properties.

B. Stability analysis

To analyse the stability of the discrete operators, we construct a global derivative matrix \mathbf{A}^d (following the procedure described in detail in Section VB). The eigenvalues of \mathbf{A}^d provide information about the stability of the method. The convective derivatives (e.g. $d = x, y$ to approximate $\partial(\cdot)/\partial x$ or $\partial(\cdot)/\partial y$) are purely dispersive - that is they contain Fourier modes which correspond to a translation, with no growth or decay. Hence the eigenvalues λ of the discrete convective operators \mathbf{A}^x and \mathbf{A}^y would (ideally) lie on imaginary axis ($\text{Re}(\lambda) = 0 \forall \lambda$) [44]. For stability in purely convective problems, no modes should grow, and so $\text{Re}(\lambda) \leq 0 \forall \lambda$. Figure 10 shows the eigenvalues of \mathbf{A}^x (with $N = 121$ computational nodes, $h/\delta r = 2$, and ghost nodes completing stencil support at boundaries) for Cartesian node distributions with and without noise, for ABFs generated from the Conic, Quadratic, Wendland, and Gaussian RBFs detailed in Appendix A. For ideal node distributions, the conic ABFs give \mathbf{A}^x with substantial non-zero real parts to the eigenvalues, whilst for the quadratic, Wendland and Gaussian ABFs, the eigenvalues lie very close to the imaginary line, although the real parts of the eigenvalues using Gaussian ABFs are several orders of magnitude greater than the quadratic and Wendland ABFs (note the scalings in the upper row of Figure 10). The scatter in the eigenvalues appears random (machine precision errors) with quadratic and Wendland ABFs, whilst there is some structure for the conic and Gaussian ABFs. Without proof, we postulate based on our numerical explorations that a condition for the real parts of the eigenvalues of \mathbf{A}^x to be zero is that $dW^0/dr = 0$ on the boundary of the support domain. This condition is exactly satisfied by the quadratic and Wendland ABFs, but not by the conic ABFs, and only approximately by the truncated Gaussian ABFs.

For noisy node distributions, the real parts of the eigenvalues of \mathbf{A}^x become significant for all ABFs tested. We see that the conic and quadratic ABFs lead to eigenvalues lying closer to the imaginary axis than the Wendland and Gaussian ABFs, with the quadratic ABFs performing best, and Gaussian ABFs worst. For all distributions, and all choices of ABF, the eigenvalues of \mathbf{A}^x fall closer to the imaginary axis for smaller k (the blue dots in Figure 10 lie closer to $\text{Im}(\lambda) = 0$ than the red and black dots.), suggesting that low order convective derivatives are likely to be more stable.

For the Laplacian, the eigenvalues of \mathbf{A}^L should be as close as possible to the negative real axis, corresponding to decay of all modes. Figure 11 shows the eigenvalues of \mathbf{A}^L for three levels of distribution noise, and three values of k , using the quadratic ABFs. For all k , the eigenvalues lie close to the real line, and the imaginary parts of the eigenvalues increase with increasing $\varepsilon/\delta r$. As k is increased, the magnitude of the eigenvalues increases, and the spread of eigenvalues move further along the negative real line. These results show that the Laplacian operator obtained with this method is stable, but that stability is conditional on the distribution noise. No modes grow in time (i.e. there are no eigenvalues with $\text{Re}(\lambda) > 0$, and those modes which translate ($\text{Im}(\lambda) \neq 0$) have significantly negative real parts to their eigenvalues, and so will be highly damped. The eigenvalue distribution when conic ABFs are used is similar to that shown in Figure 11. For the Wendland and Gaussian ABFs, we find that $\text{Re}(\lambda) > 0$ for at lower values of $\varepsilon/\delta r$. For each k and $h/\delta r$, there is a certain noise level at which the Laplacian operator will become unstable ($\text{Re}(\lambda) > 0$ for some λ). These values are listed in Table I. We see that for small k , the Laplacian is stable for much greater noise levels than large k . For $k = 2$, the Laplacian is stable for all $\varepsilon/\delta r$ tested. For large k , the Laplacian is stable for a range of $\varepsilon/\delta r$, provided a large enough $h/\delta r$ is chosen. Values of $\varepsilon/\delta r$ of 0.5 or greater represent relatively severe distribution noise, and correspond to levels which can be easily obtained in complex geometries through iterative methods such as that described in Section VB2. For Gaussian ABFs with $k = 2$, there is an upper limit to $h/\delta r$ above which the Laplacian is unstable for all noise levels. In general the Gaussian ABFs have a smaller range of stable $h/\delta r$ and $\varepsilon/\delta r$ than the Wendland and quadratic ABFs, with the quadratic ABFs being stable for the greatest range of $\varepsilon/\delta r$. The conic ABFs (not shown in Table I) exhibit similar maximum stable values of $\varepsilon/\delta r$ to the quadratic ABFs.

For purely convective problems, all choices of ABF will be unstable on noisy node distributions, given the non-zero

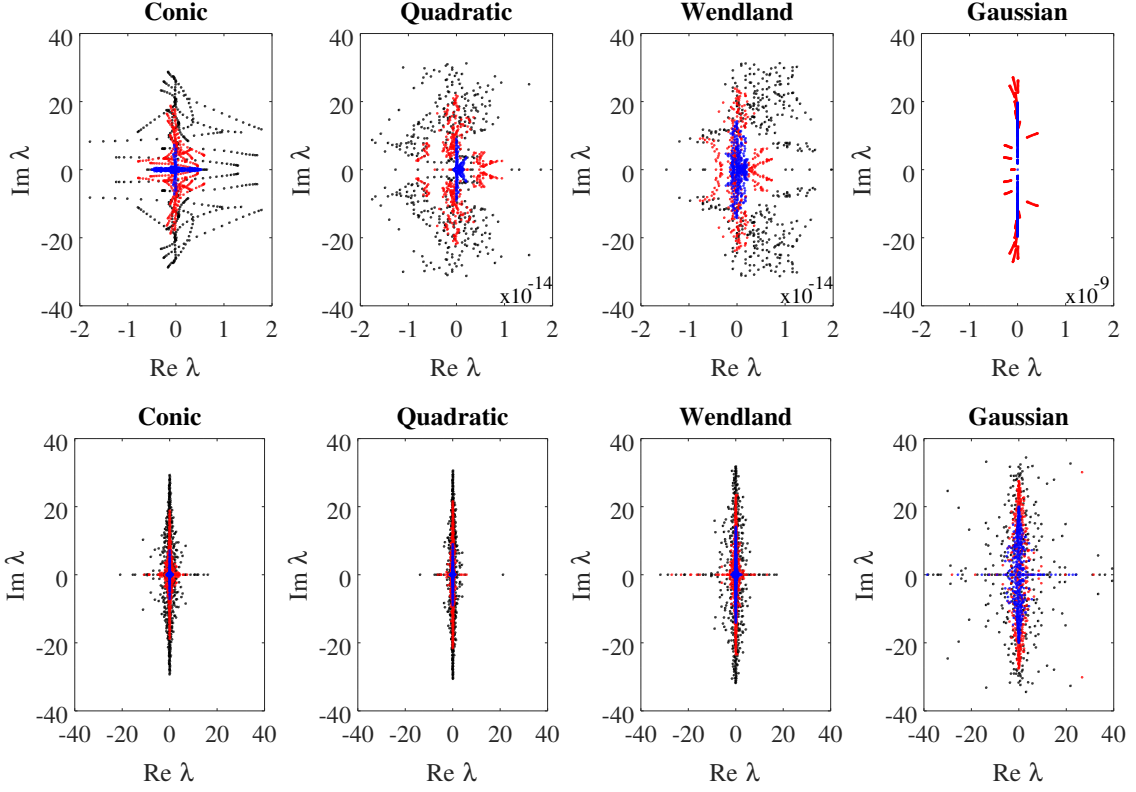


FIG. 10. The eigenvalues of the discretisation matrix \mathbf{A}^x on a 441 node Cartesian distribution, with boundaries providing complete support, for four sets of ABFs, generated from conic, quadratic, Wendland and Gaussian RBFs, and for $k = 2$ (blue), $k = 4$ (red), and $k = 6$ (black). The upper row is for ideal Cartesian node distributions, whilst the lower row is for noisy Cartesian distributions, with $\varepsilon/\delta r = 0.5$. Note the scalings of the abscissa in the upper row. In the right-most upper panel, the spread of the eigenvalues for $k = 6$ is not shown, being approximately 3 orders of magnitude greater than that for $k = 4$.

TABLE I. The maximum value of $\varepsilon/\delta r$ for stable \mathbf{A}^L (i.e. $Re(\lambda) < 0 \forall \lambda$) with the quadratic generated ABFs, on a 441 node noisy Cartesian distribution. Entries of > 3 indicate stability up the maximum value tested of $\varepsilon/\delta r = 3$. Entries of “none” indicate there is no stable noise level.

$h/\delta r$	Quadratic				Gaussian				Wendland			
	$k = 2$	$k = 4$	$k = 6$	$k = 8$	$k = 2$	$k = 4$	$k = 6$	$k = 8$	$k = 2$	$k = 4$	$k = 6$	$k = 8$
1.2	1.8	0.5	none	none	0.5	0.25	none	none	0.85	0.15	none	none
1.6	> 3	0.9	0.25	none	0.55	0.35	none	none	1.1	0.55	none	none
2.0	> 3	1.0	0.75	none	0.2	0.5	0.1	none	1.5	0.7	0.35	none
2.4	> 3	1.1	0.75	0.4	none	0.55	0.2	none	2.9	0.75	0.45	0.1
2.8	> 3	1.1	0.8	0.4	none	0.6	0.35	none	> 3	0.9	0.7	0.4

real parts to the eigenvalues of the convective derivative operators. A technique used to overcome this issue in RBF-FD is to introduce some hyperviscosity into the governing equations [41, 44], which provides the required dissipation to stabilise the solution. Hyperviscosity (i.e. $-\nabla^4$ or ∇^6), is preferred to standard second order viscosity, as it acts more on shorter wavelength modes, which leaving long wavelength modes (relative to the node distribution) which are of physical interest, relatively untouched. The addition of hyperviscosity in the present method is straightforward. A discrete hyperviscosity operator L^{hv} may be constructed in the same manner as the Laplacian and convective operators, by setting \mathbf{C}^{hv} used in the RHS of (14) appropriately. The coefficient of hyperviscosity α scales with h^m , where m is the order of the derivatives in the hyperviscosity operator, and is chosen large enough to ensure stability, but as small as possible. For example, to generate an L^{hv} which approximates the biharmonic operator $-\alpha\nabla^4$ we set

$$\mathbf{C}^{hv} = \left[0, 0, \dots, -\alpha, 0, -2\alpha, 0, -\alpha, 0 \dots \right]^T, \quad (25)$$

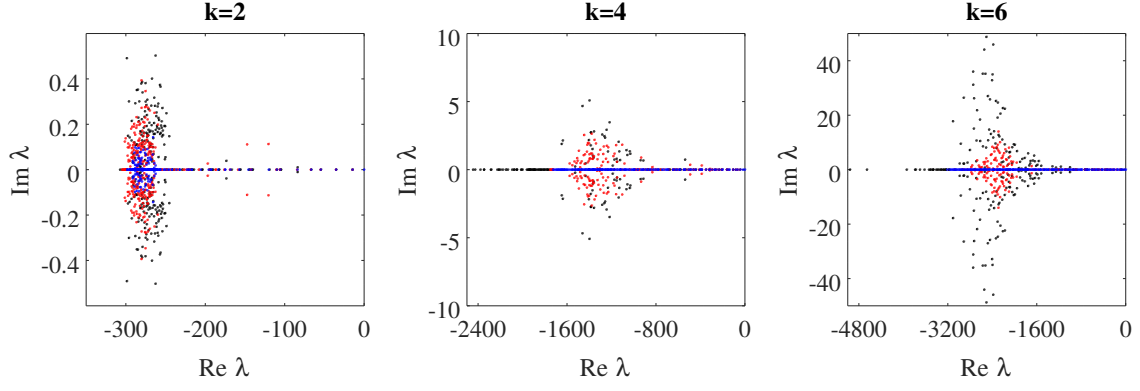


FIG. 11. The eigenvalues of the discretisation matrix \mathbf{A}^L on a 441 node noisy Cartesian distribution, with noise levels $\varepsilon/\delta r = 0$ (blue), $\varepsilon/\delta r = 0.2$ (red), and $\varepsilon/\delta r = 0.5$ (black), for three values of k . In all cases, results are obtained with the Quadratic ABFs, and with $h/\delta r = 2$.

where the non-zero elements are the elements 10, 12, and 14. To include the triharmonic operator $\alpha \nabla^6$, we set elements 21, 23, 25, and 27 equal to α , 3α , 3α , and α respectively. Note that in LABFM we can only create hyperviscosity operators up to ∇^k . We test our hyperviscosity formulation in Section V C.

V. APPLICATION TO PARTIAL DIFFERENTIAL EQUATIONS

In this section we use LABFM to solve a range of PDEs. We have deliberately chosen prototype PDEs in unbounded or simple geometries in order to demonstrate the potential of the method for application to broad range of practical problems. Throughout this section we use ABFs generated from the quadratic RBF, unless otherwise specified. As in the previous section, we use the relative L_2 -norm as a measure of error

$$L_2\text{-norm}(\cdot) = \frac{\left\{ \sum_{i=1}^N \left[(\cdot)_i^{LABFM} - (\cdot)_i^{exact} \right]^2 \right\}^{\frac{1}{2}}}{\left\{ \sum_{i=1}^N \left[(\cdot)_i^{exact} \right]^2 \right\}^{\frac{1}{2}}}, \quad (26)$$

where superscripts *LABFM* and *exact* indicate the numerical and analytic solutions respectively.

A. Parabolic: Heat equation

We first test the method on the prototypical parabolic equation - the (homogeneous) unsteady heat equation - given by

$$\frac{\partial u}{\partial t} = \kappa \nabla^2 u, \quad (27)$$

where κ is the coefficient of diffusivity. We solve (27) on a periodic square domain $(x, y) \in [0, H] \times [0, H]$. With periodic boundary conditions, all computational nodes have complete stencils. The initial conditions (at time $t = 0$) and solution are given by

$$u(x, y, t) = \sin\left(\frac{2\pi x}{H}\right) \sin\left(\frac{2\pi y}{H}\right) \exp\left[\frac{-8\kappa\pi^2 t}{H^2}\right]. \quad (28)$$

We discretise the domain with a noisy Cartesian distribution, with the noise having magnitude $\varepsilon/\delta r = 0.5$, where δr is the noise-free node spacing. We set $h/\delta r = 2$, and use the classical fourth order Runge-Kutta (RK4) scheme for time integration, and a time step of $\delta t = 0.05h^2/\kappa$, setting $\kappa = 1$.

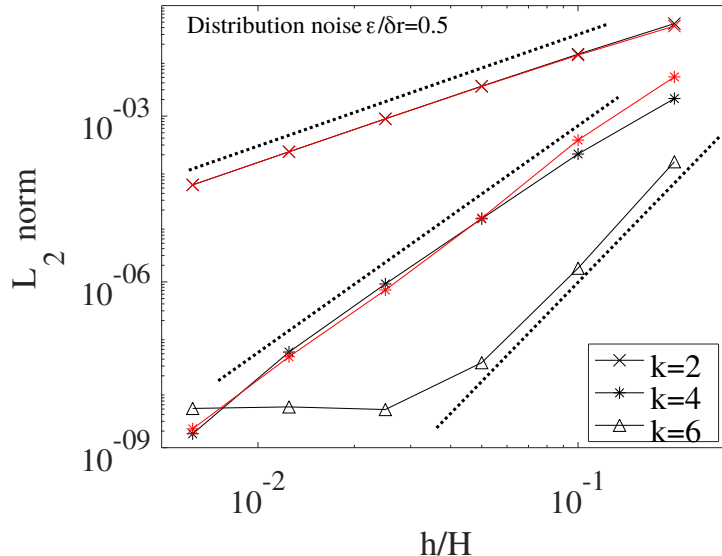


FIG. 12. Variation of the L_2 -norm of the solution to the heat equation at $t^* = 1$ with resolution, for different values of k (see legend). Black lines correspond to the case with periodic boundaries, and red lines indicate Dirichlet conditions with incomplete stencils near boundaries. The dotted lines correspond to convergence rates of 2, 4 and 6.

The L_2 -norm error of the numerical solution at non-dimensional time $t^* = 8\pi^2\kappa t/H^2 = 1$ is shown by the black lines and symbols in Figure 12. We see convergence rates of approximately k , as k is varied between 2 and 6. Next, we replace the periodic boundary conditions with homogeneous Dirichlet conditions ($u = 0$ on the boundary), so that the stencils of nodes near and on the boundaries are incomplete. Dirichlet conditions (either homogeneous or inhomogeneous) are imposed by prescribing u for all nodes on the boundary, and solving (27) only for internal nodes. The results with Dirichlet conditions are shown in red in Figure 12. For $k \leq 4$ the scheme is stable, and again we see convergence rates close to k . For $k > 4$, the incomplete stencils near the boundaries (in particular the corners in the present case) lead to non-negative real parts to the eigenvalues of \mathbf{A}^L , and hence the scheme becomes unstable. We next replace the homogeneous Dirichlet boundary conditions with the inhomogeneous $u(x, 0, t) = \sin(\pi x)$ for $x \in [0, H]$, and $u(x, y, t) = 0$ on all other boundaries. Subject to these boundary conditions, the steady state solution of (27) is given by

$$u_{ss}(x, y) = \sinh(\pi[1 - y]) \sin(\pi x) / \sinh \pi. \quad (29)$$

Table II shows the L_2 -norm and convergence rate of the steady state numerical solution (taken as the solution at non-dimensional time $t^* = 100$). We find the method converges with order between k and $k + 1$ (for $k \leq 4$). The ability to handle incomplete stencils accurately is a particularly attractive feature of LABFM. In effect, for $k \leq 4$ the method automatically generates one sided derivative approximations near boundaries, of order k . Indeed, for $k > 4$, the gradient and Laplacian operators can still converge with order k , provided $\varepsilon/\delta r$ is small enough near boundaries, and $h/\delta r$ is large enough. Typically the range of acceptable $\varepsilon/\delta r$ and $h/\delta r$ decreases near boundaries with incomplete support. Obvious approaches to improving stability and accuracy near boundaries include increasing the resolution near boundaries, or decreasing k to 4 near boundaries - which will be explored in a future study on adaptivity.

B. Elliptic: Poisson's equation

Our next PDE is Poisson's equation. Solving this elliptic PDE is in essence the inverse problem to the solution of the heat equation in the previous section. The Poisson's equation is given by

$$\nabla^2 \phi = f(x, y) \quad (30)$$

TABLE II. The steady state L_2 -norm error in the numerical solution of the heat equation subject to inhomogeneous boundary conditions (with analytical solution given by (29)). Figures in brackets show the convergence rate. Results were obtained using quadratic generated ABFs with $h/\delta r = 2$ and $\varepsilon/\delta r = 0.5$.

h/H	$k = 2$	$k = 3$	$k = 4$
0.2	3.9×10^{-3} (-)	4.6×10^{-4} (-)	1.4×10^{-4} (-)
0.1	7.1×10^{-4} (2.46)	5.4×10^{-5} (3.08)	5.6×10^{-6} (4.68)
0.05	1.1×10^{-4} (2.73)	9.3×10^{-6} (2.54)	2.2×10^{-7} (4.67)
0.025	1.5×10^{-5} (2.83)	2.0×10^{-6} (2.20)	8.3×10^{-9} (4.72)
0.0125	2.2×10^{-6} (2.78)	4.7×10^{-7} (2.10)	2.9×10^{-10} (4.83)

in the domain Ω , with boundary conditions

$$\phi = g_1(x, y) \quad \text{on } \Gamma_D \quad (31a)$$

$$\frac{\partial \phi}{\partial n} = g_2(x, y) \quad \text{on } \Gamma_N, \quad (31b)$$

where the boundary $\Gamma = \Gamma_D + \Gamma_N$. To solve (30) we construct a (linear) global discrete Laplacian operator $\mathbf{A}^{d=L} = \mathbf{A}^L$ (an $N \times N$ matrix) from the local operators L_i^L (where the superscript L indicates that the operator approximates the Laplacian). The i -th row of \mathbf{A}^L is a rearrangement of the local operator L_i^L , which is achieved by setting the elements of \mathbf{A}^L as

$$A_{i,j}^L = w_{ji}^L \quad \forall j \neq i \quad (32a)$$

$$A_{i,i}^L = - \sum_j w_{ji}^L. \quad (32b)$$

The discretised form of (30) is then

$$\mathbf{A}^L \Phi = \mathbf{F}, \quad (33)$$

with solution vector

$$\Phi = [\phi_1, \phi_2, \dots, \phi_N]^T \quad (34)$$

and source vector

$$\mathbf{F} = [f_1, f_2, \dots, f_N]^T. \quad (35)$$

The linear system (33) may be solved using an iterative solver. In the present work, we use the stabilised bi-conjugate gradient algorithm, with a Jacobi preconditioner. Given the implicit method of solution, boundary conditions (31a) and (31b) must be incorporated into (33), and this is done by manipulation of the rows, columns and elements of \mathbf{A}^L and \mathbf{F} . On sections of the boundary Γ_D , we satisfy (31a) (on an example boundary node b) by setting

$$A_{b,j}^L = 0 \quad \forall j \neq b \quad (36a)$$

$$A_{b,b}^L = 1 \quad (36b)$$

$$F_b = g_{1,b}, \quad (36c)$$

which ensures that the solution to (33) yields $\phi_b = g_{1,b}$. On Γ_N , where Neumann boundary conditions are specified, we must satisfy both (30) and (31b) for each boundary node b . To do this, we use an approach which is consistent with the order of the spatial discretisation as follows. For every boundary node b on Γ_N , we introduce an additional ‘ghost’ node \hat{b} , such that $\mathbf{r}_{\hat{b}} = \mathbf{r}_b - \mathbf{n}_b \delta r$, where $\mathbf{n}_b = (n_{b,x}, n_{b,y})$ is the unit normal pointing into the domain at parent node b , and δr is the initial node spacing. We increase the size of the linear system (33) to include the ghost nodes. The rows of \mathbf{A}^L corresponding to ghost nodes \hat{b} are used to enforce (31b), by ensuring the discrete boundary normal derivative operator at node b satisfies (31b)

$$L_b^n = \sum_j (\cdot)_{jb} (n_{b,x} w^{x,bj} + n_{b,y} w^{y,bj}) = g_{2,b}. \quad (37)$$

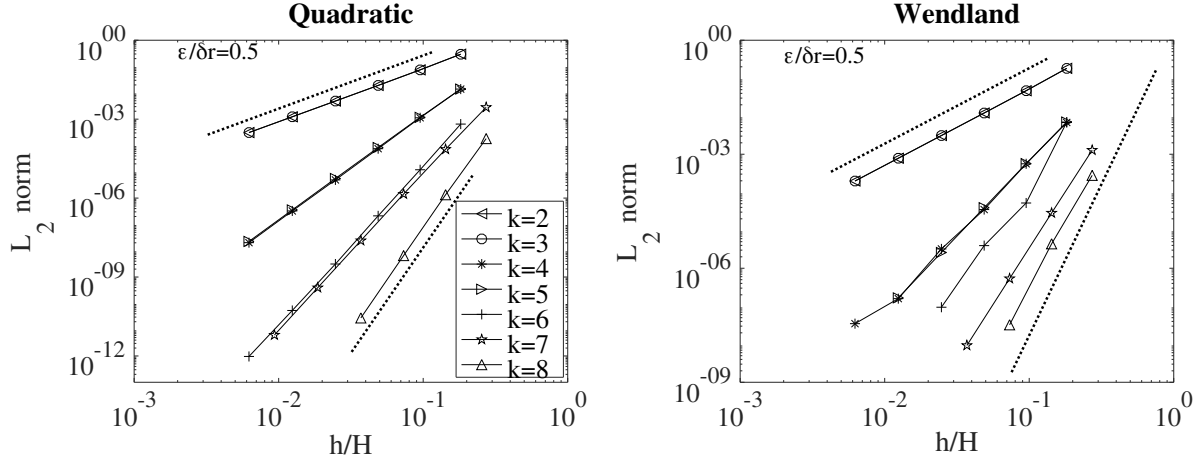


FIG. 13. Variation of the L_2 -norm with resolution for the periodic Poisson's equation case 1 (periodic), for $k \in [2, 8]$. The left panel shows errors using ABFs obtained from the Quadratic RBF, and the right panel shows ABFs obtained from the Wendland RBF. In all cases, $\varepsilon/\delta r = 2$. For $k \leq 6$ we use $h/\delta r = 2$, and for $k = 7, 8$ we use $h/\delta r = 3$. The dotted lines show convergence rates of 2 and 8.

We do this by setting, for every \hat{b} and its parent b

$$A_{\hat{b},j}^L = n_{b,x}w^{x,bj} + n_{b,y}w^{y,bj} \quad \forall j \neq b \quad (38a)$$

$$A_{\hat{b},b}^L = - \sum_j (n_{b,x}w^{x,bj} + n_{b,y}w^{y,bj}) \quad (38b)$$

$$F_{\hat{b}} = g_{2,b}, \quad (38c)$$

where the sum is over all j which are neighbours of b . When the system (33) is solved with this modification, the elements of the solution vector corresponding to the ghost nodes ($\Phi_{\hat{b}} \forall \hat{b}$) will take values which ensure that the Neumann boundary condition (31b) is satisfied on Γ_N . We note that this approach to Neumann boundary conditions appears (coincidentally) similar to that described by Mishra et al. [45].

1. Periodic test case

Our first test case for Poisson's equation uses a periodic square domain $(x, y) \in [0, H] \times [0, H]$, and sets f in (30) such that $\phi = \sin(2\pi x/H) \sin(2\pi y/H)$. Figure 13 shows the L_2 -norm of the solution for a range of resolutions and values of k . Using the Quadratic ABFs (left panel) we find convergence orders of 2 for $k = 2, 3, 4$ for $k = 4, 5, 6$ for $k = 6, 7$ and 8 for $k = 8$. For the $k = 8$ the linear solver fails to converge at the finest resolutions. With the Wendland ABFs (right panel) the convergence rates follow the same pattern as with the Quadratic ABFs, although but at higher k , the solution becomes dependent on the node distribution disorder. The linear solver fails to converge at the finest resolution for $k \geq 5$. This improved stability of the Quadratic over the Wendland ABFs is consistent with the observation in IV B that the eigenvalues of \mathbf{A}^L obtained with Wendland ABFs contain positive real parts for lower levels of distribution noise than when Quadratic ABFs are used.

2. Annular domain test case

Next we use an annular domain with inner diameter $H/4$ and outer diameter H , centred on the origin. The domain is illustrated in the left panel of Figure 14. We solve (30) with the source term

$$f = \left[12\pi \cos 4\pi r - \left(16\pi^2 r - \frac{1}{r} \right) \sin 4\pi r \right] \cos 3\theta - \frac{9}{r} \cos 3\theta \sin 4\pi r, \quad (39)$$

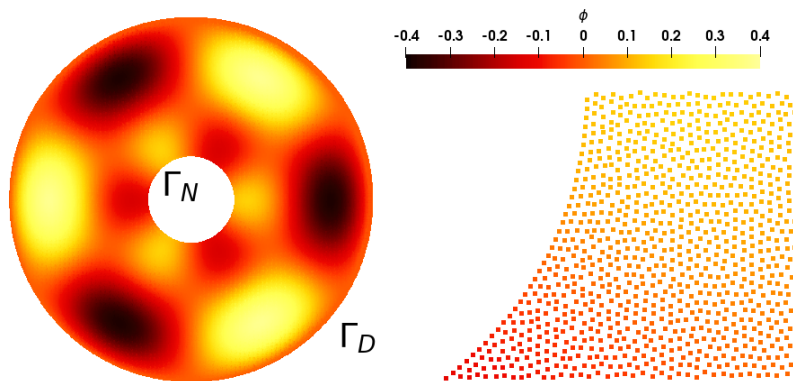


FIG. 14. The domain and solution for Poisson problem 2 (left), with a close up of a typical node distribution (right).

with r and θ being polar coordinates based at the origin, subject to the boundary conditions $g_1 = 0$ on Γ_D (the external boundary) and $g_2 = \cos 3\theta$ on Γ_N (the internal boundary). The solution is given by

$$\phi = r \sin 4\pi r \cos 3\theta, \quad (40)$$

and shown in the left panel of Figure 14. The node distribution is generated by discretising the boundaries (placing nodes at intervals δr along all boundaries, then filling the domain with a noisy Cartesian distribution, with $\varepsilon/\delta r = 0.5$. Temporary ghost nodes are generated outside the domain to complete support, whilst an iterative procedure is used to shift nodes according to:

$$\mathbf{r}_i^{n+1} = \mathbf{r}_i^n + \frac{\delta r^2}{h} \sum_{j \in |\mathbf{r}_{ji}| < h} \left(\frac{|\mathbf{r}_{ji}|}{h} - 1 \right) \frac{\mathbf{r}_{ji}}{|\mathbf{r}_{ji}|}. \quad (41)$$

We apply typically 10 iterations of (41) to obtain the final node distribution, an example of which is shown in Figure 14 (right panel). The procedure given by (41) moves nodes from regions of high node density to low node density, providing a more uniform distribution. The temporary ghost nodes prevent nodes from being shifted out of the domain during the application of (41). After the shifting procedure, the temporary ghost nodes are discarded, and a ghost node is added for every node on Γ_N as described above. The procedure (41) is relatively unrefined, however the resilience of LABFM to node distribution noise allows us to use this cheap procedure with only a few iterations, and still obtain high order convergence rates.

Table III shows the variation of the L_2 -norm of the solution with resolution, for $k = \{2, 3, 4\}$. For $k = 2, 3$, we observe second order convergence, whilst for $k = 4$ we observe fourth order convergence. As with the parabolic problem in the previous section, for $k > 4$, the ABFs are inadequately sampled on nodes near the boundaries, where computational stencils have incomplete support. For $k \leq 4$, LABFM automatically generates one sided differences as required. A similar result was found by Bayona et al. [46], who observed that when solving elliptic problems with RBF-FD, the method could handle incomplete support provided large enough computational stencils were used near boundaries. For $k > 4$, the linear solver used to solve (30) did not converge. Preliminary investigations in which we set $k = 4$ near boundaries, and $k > 4$ on internal nodes were promising, yielding stable solutions, but for the present problem, only a minor increase in accuracy. This *adaptive accuracy* approach will be explored further in a future study.

C. Hyperbolic-Parabolic: Viscous Burgers' equation

Next we test LABFM on the viscous Burgers' equation, which is of mixed hyperbolic-parabolic type, and a prototypical shock-admitting conservation equation. The viscous Burgers' equation is written

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = \frac{1}{Re} \nabla^2 \mathbf{u} \quad (42)$$

where Re is the Reynolds number, and $\mathbf{u} = (u, v)$ is a velocity vector. We integrate (42) using the RK4 scheme, with a time step of

$$\delta t = \min(0.2h/\max|u|, 0.05h^2 Re). \quad (43)$$

TABLE III. The L_2 -norm error in the numerical solution of Poisson's equation case 2 (annular), for different k . Figures in brackets show the convergence rate. Results were obtained using quadratic generated ABFs with $h/\delta r = 2$.

h/H	$k = 2$	$k = 3$	$k = 4$
2/13	0.655 (-)	0.539 (-)	1.05 (-)
2/25	0.136 (2.40)	0.146 (2.00)	0.101 (3.58)
2/49	3.5×10^{-2} (2.01)	3.7×10^{-2} (2.03)	2.9×10^{-3} (5.28)
2/97	9.1×10^{-3} (1.98)	9.5×10^{-3} (2.01)	1.4×10^{-4} (4.44)
2/193	2.3×10^{-3} (1.98)	2.4×10^{-3} (2.00)	6.5×10^{-6} (4.47)
2/385	5.9×10^{-4} (1.99)	6.0×10^{-4} (2.00)	3.6×10^{-7} (4.20)

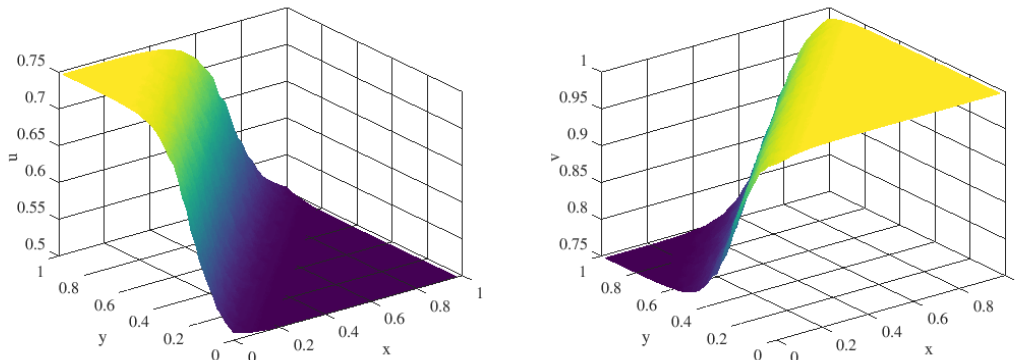


FIG. 15. Numerical solution to the Burgers' equation case 1 (travelling wave) at non-dimensional time $t = 1$ for $Re = 100$. Left and right panels show u and v respectively.

We solve (42) for two cases.

1. Travelling wave test case

The first test comprises a travelling wave oriented diagonally along the xy -plane, in which steepening due to non-linear advection is balanced by viscous diffusion. The steepness of the travelling wave increases with increasing Reynolds number, and in the inviscid limit the wave becomes a discontinuity. We use a computational domain $(x, y) \in [0, 1] \times [0, 1]$, discretised with a noisy Cartesian node distribution with $\varepsilon/\delta r = 0.2$. At the boundaries, we continue the node distribution outwards to provide full support, and the analytic solution is prescribed on nodes outside the boundary. The initial conditions and solution are

$$u(x, y, t) = \frac{3}{4} - \frac{1}{4(1 + e^{Re(-t-4x+4y)/32})}, \quad (44a)$$

$$v(x, y, t) = \frac{3}{4} + \frac{1}{4(1 + e^{Re(-t-4x+4y)/32})}. \quad (44b)$$

The numerical solution at $t = 1$ is shown in Figure 15. Figure 16 shows the convergence of errors for this problem at $t = 1$, for increasing k , for three Reynolds numbers, $Re = 10$ (blue lines), $Re = 100$ (black lines), and $Re = 500$ (red lines). For the highest Reynolds number the shock is too steep to be resolved by the coarsest two resolutions (not shown), and the simulations are unstable. The magnitude of the errors increase with increasing Re , due to the increasingly steep gradients present in the solution. For all cases, the solution converges with order approximately k , and convergence is within $[k - 1/2, k + 1/2]$. For $Re = 10$ and $k = 6$, the coarsest resolution shown corresponds to only 10×10 over the domain, and yet the relative error is less than 10^{-8} .

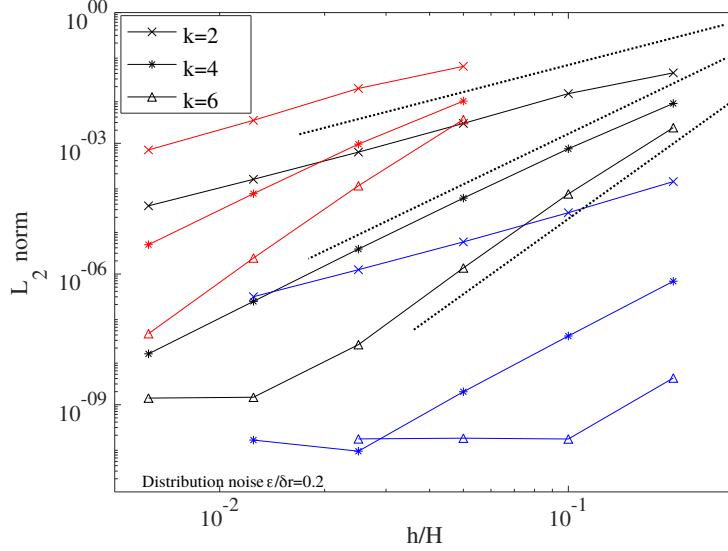


FIG. 16. L_2 -norm of the error in the solution of Burgers' equation case 1 (travelling wave) at non-dimensional time $t = 1$, for a range of resolutions, for various k (see symbols in legend), and different Reynolds numbers: $Re = 10$ (blue lines), $Re = 100$ (black lines) and $Re = 500$ (red lines). In all cases, $h/\delta r = 2$ and $\varepsilon/\delta r = 0.2$. The dotted lines show convergence rates of 2, 4 and 6.

2. Periodic test case

Next we solve (42) on a periodic square domain $(x, y) \in [0, H] \times [0, H]$, with sinusoidal initial conditions: $u(x, y, 0) = \sin(2\pi x/H)$ and $v(x, y, t) = 0$. The advection terms contribute to the development of a shock (as the sinusoidal solution tends to a sawtooth wave), which the viscous terms dissipate. The solution was first derived by Cole [47], and has since been used as a benchmark for numerical methods by numerous authors, for example [28]. The solution is given by

$$u(x, t) = \frac{4\pi}{HRe} \frac{\sum_{n=1}^{\infty} n A_n \sin\left(\frac{2n\pi x}{H}\right) e^{\left(\frac{-n^2 \pi^2 t}{H^2 Re}\right)}}{A_0 + \sum_{n=1}^{\infty} A_n \cos\left(\frac{2n\pi x}{H}\right) e^{\left(\frac{-n^2 \pi^2 t}{H^2 Re}\right)}} \quad (45)$$

where

$$A_0 = \exp\left[\frac{-ReH}{4\pi}\right] I_0\left(\frac{ReH}{4\pi}\right); \quad A_n = 2 \exp\left[\frac{-ReH}{4\pi}\right] I_n\left(\frac{ReH}{4\pi}\right), \quad (46)$$

and I_n are modified Bessel functions of the first kind. Note that $v(x, y, t) = 0$ for all t . We use $Re = 100$, and find that only the first 30 terms are necessary, as $A_n < 10^{-16}$ for $n \geq 30$. Figure 17 shows the analytic (red lines) and numerical solutions (black dots) at a range of times, from $t = 0$ to $t = 1$, with $\varepsilon/\delta r = 0.5$, $k = 4$, $h/\delta r = 2$ and $h/H = 2/81$. Figure 18 shows the variation of the L_2 -norm with time, for a range of resolutions and k , with $\varepsilon/\delta r = 0.2$. As with the previous case, we found convergence rates between k and $k - 1$ as expected. These convergence rates held when the disorder in the node distribution was increased to $\varepsilon/\delta r = 0.5$. The quadratic ABFs (red symbols) provide slightly better accuracy than the conic ABFs (black). For all simulations which remain stable, the errors grow during the advection dominated initial stages (as the sinusoidal profile steepens), and then decrease during the viscosity dominated stages thereafter. For $k = 2$, the coarsest resolution is unstable for both conic and quadratic ABFs, and the second coarsest resolution is unstable for the conic ABFs (left panel of Figure 18). Generally we observed the quadratic ABFs to be slightly more accurate, and slightly more stable (for this problem) than the conic ABFs.

Figure 19 (left panel) shows the solution at $t = 0.4$ with $k = 4$ as the resolution is varied. For the coarse resolution of $h/H = 2/41$ (blue dots) there is significant noise in the solution, as the shock is under-resolved, although the

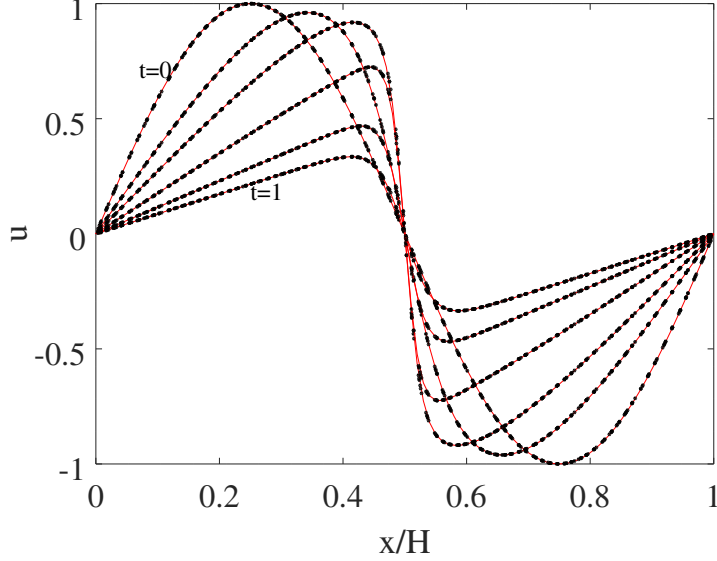


FIG. 17. The analytic (red line) and numerical solutions (black dots) of the periodic Burgers' problem with $Re = 100$, $h/\delta r = 2$, and $\varepsilon/\delta r = 0.5$, at times $t = 0$, $t = 0.1$, $t = 0.2$, $t = 0.4$, $t = 0.7$ and $t = 1$, obtained with a resolution of $h/H = 2/81$.

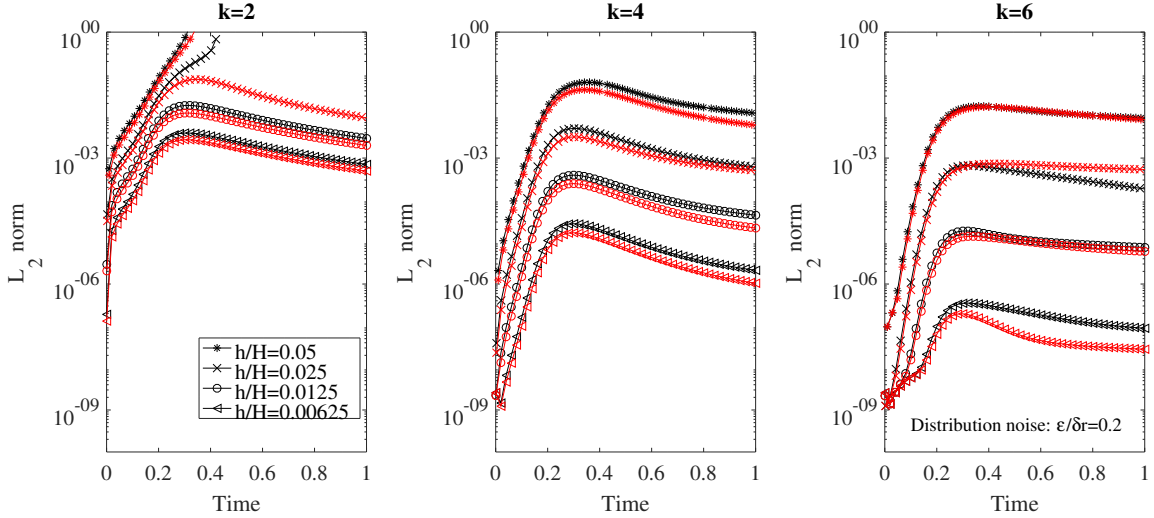


FIG. 18. Variation of the L_2 -norm of the solution to the periodic Burgers' problem with time, for a range of resolutions (identified in legend), for the conic ABFs (black) and quadratic ABFs (red). The node distribution is noisy Cartesian, with $\varepsilon/\delta r = 0.2$.

simulation remains stable. The right panel of Figure 19 shows a single resolution $h/H = 2/81$, as k is increased. For $k = 2$ (red dots) the solution contains overshoots around the shock. This is because the eigenvalues of the Laplacian operator with $k = 2$ have smaller negative real part than those for higher k , and hence are less able to dissipate the high order modes present in the solution. For $k = 4$ and $k = 6$ we see a close match with the analytic solution (thin red line).

We increase the Reynolds number to $Re = 250$, and again run the simulation with $\varepsilon/\delta r = 0.5$, $h/\delta r = 2$, $k = 6$ and $h/H = 2/81$. The simulation becomes unstable prior to $t = 0.4$. By adding hyperviscosity to the scheme as described in Section IV B, we are able to stabilise the simulations, whilst retaining relatively good accuracy. Figure 20 shows the solution at several times using a biharmonic hyperviscosity of $-100h^4\nabla^4$ (black dots). The inset shows detail of the solution at $t = 0.5$ using the biharmonic hyperviscosity, and a triharmonic hyperviscosity of $8h^6\nabla^6$ (blue dots). The coefficients of hyperviscosity were chosen by numerical experiment as approximately the minimum values which

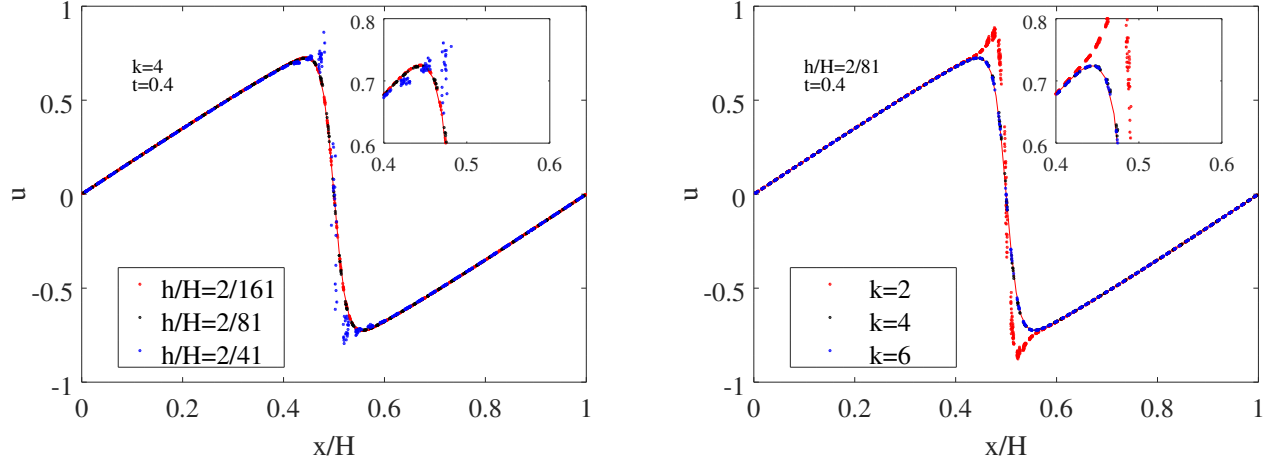


FIG. 19. The analytic (red line) and numerical solutions (coloured dots) of the periodic Burgers' problem for a range of parameters, with $Re = 100$, $h/\delta r = 2$, and $\varepsilon/\delta r = 0.5$. Left panel: the effects of varying resolution on the solution at $t = 0.4$, with $k = 4$. Right panel: The effects of varying k on the solution at $t = 0.4$, with $h/H = 2/81$.

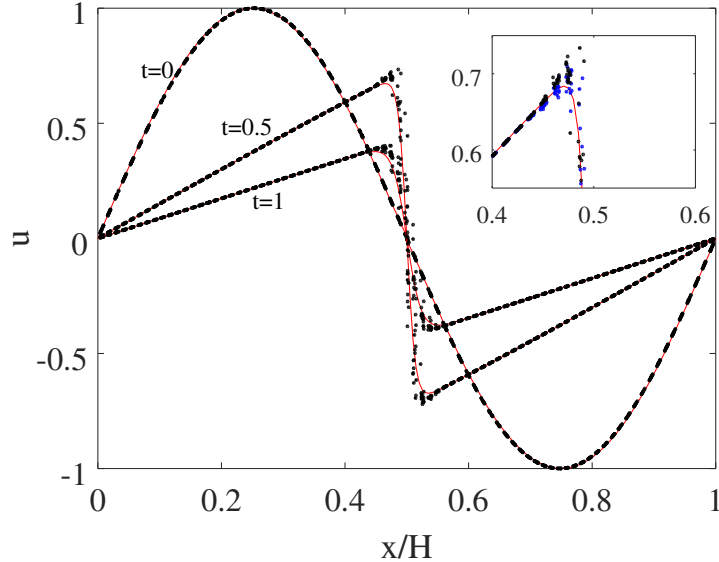


FIG. 20. The analytic (red line) and numerical solutions (coloured dots) of the periodic Burgers' problem with $Re = 250$, $h/\delta r = 2$, and $\varepsilon/\delta r = 0.5$, and $k = 6$. The solution is shown at times $t = 0$, $t = 0.5$, and $t = 1$, obtained with a resolution of $h/H = 2/81$. Black dots indicate biharmonic hyperviscosity, and blue dots (inset) indicate triharmonic hyperviscosity.

stabilised the solution. Both forms of hyperviscosity are capable of stabilising the solution, whilst retaining relatively good accuracy. In the present case we see only minor differences between biharmonic and triharmonic hyperviscosity.

We now modify our initial conditions to the two-dimensional sinusoidal functions

$$u = \sin\left(\frac{2\pi x}{H}\right) \sin\left(\frac{2\pi y}{H}\right); \quad v = -\cos\left(\frac{2\pi x}{H}\right) \cos\left(\frac{2\pi y}{H}\right) \quad (47)$$

and run the simulation for a range of Reynolds numbers, up to $Re = 400$. We do not have an analytic solution for the two-dimensional initial conditions, although intuitively it can be seen that certain slices through the domain must have an analytic solution which matches (or is a translation of) the solution for the one-dimensional initial conditions. Figure 21 shows the solution at $t = 0.4$ for several values of Re , with a resolution of $h/H = 2/41$ corresponding to 6561 nodes in the domain. In cases, we set $k = 6$, $h/\delta r = 2$, and $\varepsilon/\delta r = 0.5$. In ordinary use, we found a limit of $Re \approx 150$

at this resolution (the limit increased with increasing resolution), due to the lack of upwinding in the LABFM. With biharmonic hyperviscosity the simulations are stable up to $Re = 400$. For this Reynolds number (rightmost panel of Figure 21), the width of the shock is approximately h , and despite this under-resolution, hyperviscosity provides stability without significant smoothing of the solution.

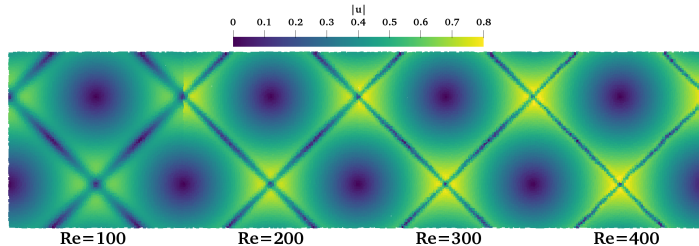


FIG. 21. The velocity magnitude field of the periodic Burgers' problem with two-dimensional initial conditions at $t = 0.4$ for a range of Re . The solution is obtained with $h/H = 2/41$, $\varepsilon/\delta r = 0.5$, $h/\delta r = 2$ and $k = 6$.

D. Navier-Stokes equations

Finally, we test the ability of our method to solve the incompressible Navier-Stokes equations. This test incorporates aspects of the previous PDEs, with the momentum equation being a non-linear hyperbolic-parabolic equation, and the incompressibility constraint being manifest as an elliptic PDE. Expressed in terms of the vorticity, the Navier-Stokes equations may be written

$$\frac{\partial \omega}{\partial t} + \mathbf{u} \cdot \nabla \omega = \frac{1}{Re} \nabla^2 \omega \quad (48)$$

where ω is the vorticity and \mathbf{u} is the velocity. Incompressibility is enforced by obtaining the velocity from a stream-function ψ , such that $\mathbf{u} = \nabla \times \psi$. The stream-function is obtained from the vorticity by solution of a Poisson's equation:

$$\nabla^2 \psi = -\omega. \quad (49)$$

We integrate (48) using RK4, and after each substep, solve (49) to obtain ψ , then calculate gradients of ψ to obtain \mathbf{u} for use in the next substep. We set the time step using (43). In the present study, we consider the Navier-Stokes equations in unbounded domains. The vorticity-streamfunction formulation is chosen for its ease of implementation in high-order explicit time-stepping schemes. The case we study here is that of Taylor-Green vortices. The initial conditions and solution are given by

$$\omega(x, y, t) = \frac{4\pi}{H} \sin(2\pi x/H) \sin(2\pi y/H) \exp\left(\frac{-8\pi^2 t}{ReH^2}\right). \quad (50)$$

The velocity field is orthogonal to the gradient of the vorticity field, and so the advection term in (48) is zero for the analytic solution, and the vorticity field decays exponentially. Figure 22 shows the solution at times $t = 0$ and $t = 1$. For $Re = 100$, the maximum vorticity decays from 4π (approximately 12.6) to 5.7. Figure 23 shows the variation of the L_2 -norm with resolution at time $t = 1$ with $Re = 100$, for different values of k . The coarsest resolution shown in Figure 23 is twice as coarse as the results shown in Figure 22. For $k = \{2, 4, 6\}$ we use $h/\delta r = 2$ and $\varepsilon/\delta r = 0.5$, whilst for $k = 8$ we use $h/\delta r = 2.5$ and $\varepsilon/\delta r = 0.2$ (for stability). The errors converge with approximately order k , as expected. The clear error limit of approximately 10^{-8} is due to the accumulation of errors in the time stepping scheme. The dominant errors contributing to this limit are those originating in the approximate solution of (49), where the tolerance of the linear solver is set to 10^{-10} . We find the rates of convergence remain when the Reynolds number is reduced to $Re = 10$ or increased to $Re = 1000$.

Whilst in the present study we have considered the Navier-Stokes equations only in unbounded domains, these results clearly demonstrate the potential of LABFM as a viable mesh-free method for computational fluid dynamics simulations. In complex geometries the vorticity-stream function formulation chosen here requires careful (and computationally expensive) boundary treatment [48], and for three-dimensional flows becomes more expensive still, as a Poisson equation must be solved for each component of the stream function vector field. Due to the collocated

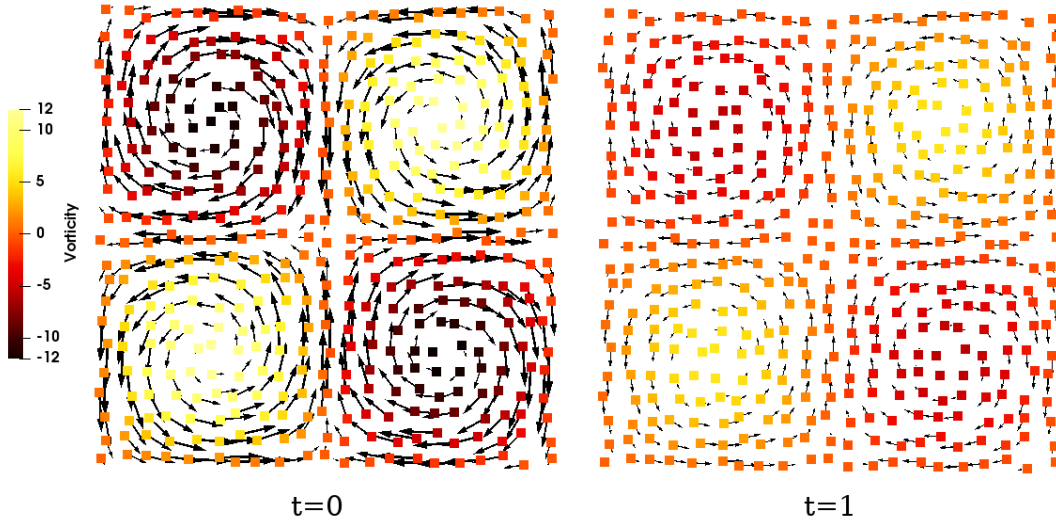


FIG. 22. The numerical solution to Taylor-Green vortices at times $t = 0$ (left) and $t = 1$ (right) for $Re = 100$, on a node distribution with $h/H = 2/11$ and $\varepsilon/\delta r = 0.5$. Colour indicates vorticity, and arrows indicate velocity.

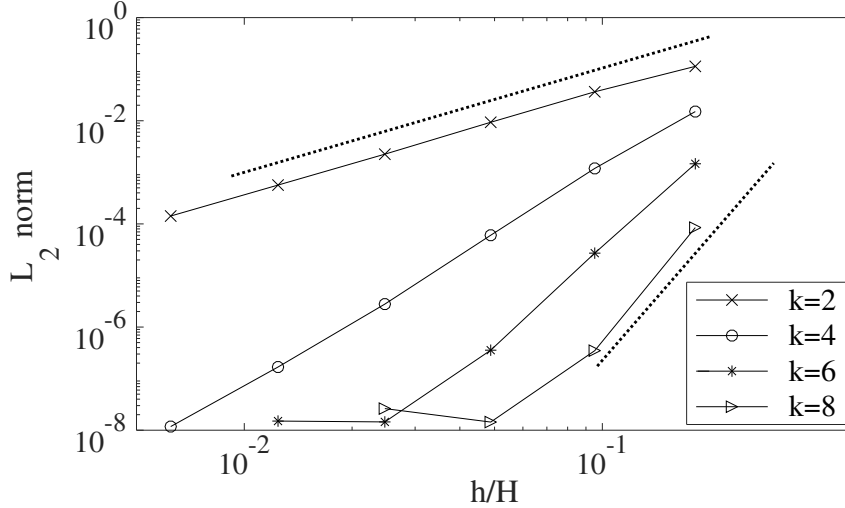


FIG. 23. Variation of the L_2 -norm with resolution for the Taylor-Green vortices problem at time $t = 1$, for different values of k . For $k = \{2, 4, 6\}$ we use $h/\delta r = 2$ and $\varepsilon/\delta r = 0.5$. For $k = 8$ we use $h/\delta r = 2.5$ and $\varepsilon/\delta r = 0.2$. Simulations were run with $Re = 100$, and using quadratic ABFs. The dotted lines show convergence rates of 2 and 8.

nature of this method, a primitive variable formulation is unstable, as the LBB condition [49] is not satisfied. The development of a robust primitive variable formulation for LABFM is an ongoing area of research: a further study is planned involving the development of the LABFM framework for the solution of the Navier-Stokes equations in bounded complex geometries, with adaptivity in resolution and accuracy. The pairwise symmetry which leads to the conservation properties of SPH is necessarily broken ($w_{ji}^d \neq w_{ij}^d$) in LABFM, as the node distribution is accounted for to yield consistency. Hence, the method does not exhibit exact conservation: global conservation errors are determined by the order of the spatial discretisations k , the time integration scheme, and any boundary conditions. Based on finite volume concepts, a number of authors have developed consistent mesh-free methods which also exhibit exact conservation [50, 51, 52]), though these have been limited to first or second order convergence rates. Such methods have significant potential for the accurate, efficient and robust simulation of flows in complex geometries, and an implementation of LABFM such that both exact conservation and high order consistency are retained is the subject of ongoing research.

VI. CONCLUSIONS

In this work we have presented a novel framework for the calculation of high order finite differences on disordered node distributions. We have shown the ability of our method, referred to as the Local Anisotropic Basis Function method (LABFM), to generate up to eighth order discrete operators with compact computational stencils, which we have used to solve a range of prototype PDEs. Our method has several attractive characteristics:

1. The accuracy of the approximations holds even for node distributions with significant local anisotropy, provided the ABFs contributing to the discrete operator are adequately sampled over the computational stencil. In practice, in two dimensions we can obtain 4th order with $\mathcal{N} \approx 25$, and 6th order with $\mathcal{N} \approx 48$, and 8th-order with $\mathcal{N} \approx 60$ on severely disordered distributions. This characteristic will allow the method to remain extremely accurate even in complex geometries where non-uniformity in the node distribution is unavoidable. In this regard the method has similarities with RBF-FD, although a benefit of LABFM is our ability to exactly prescribe the order or accuracy level. Furthermore, in the limiting low order case, our method collapses to SPH. Hence there is significant scope for a naturally adaptive (in resolution, accuracy and frame of reference) ALE scheme.
2. In LABFM we solve a linear system to obtain weights for a discrete operator, rather than to directly obtain derivative approximations as in FPM and CSPM. Therefore for Eulerian schemes the method is extremely fast, matching the $\mathcal{O}(N)$ FLOPs per time-step of RBF-FD, and for fourth order and higher, allowing smaller \mathcal{N} . In certain conditions LABFM is equivalent (in terms of errors) to certain forms of CSPM, but the generalisation to an arbitrary set of ABFs yields increased stability and resilience to node disorder as the order of the scheme is increased.
3. As with RBF-FD, one-sided discrete operators are automatically generated at truncated domain boundaries, with the same order as the internal scheme, provided the node distribution adequately samples the set of ABFs used. Hyperviscosity, which effectively filters out unphysical short wavelength disturbances, can also be easily included in the discrete operators in LABFM, providing a simple means to increase stability for hyperbolic problems.
4. The notation describing the construction of LABFM is concise and easy to follow, and the implementation of the method into existing mesh-free codes should be straightforward.

These characteristics give the method significant potential to be a powerful tool for the solution of PDEs in complex geometries. By virtue of its local nature, we expect LABFM to be highly scalable and computationally efficient, properties which will be explored in a further study. Whilst we have demonstrated up to eighth order schemes in the present study, this need not be the limit, provided computational stencils are increased as required. A further study, applying LABFM to solve the Navier Stokes equations in complex geometries is ongoing, and future work will involve the extension of the framework to include adaptivity in resolution and accuracy.

ACKNOWLEDGEMENTS

We are grateful for financial support from the Leverhulme Trust, via Research Project Grants RPG-2017-144 and RPG-2019-206, and EPSRC via grant EP/R005729/1. We thank Georgios Fourtakas for several useful discussions regarding node distributions. We would like to thank two anonymous reviewers for very helpful comments, both regarding the current manuscript and our future work.

Appendix A: Details of ABFs used

Here we provide details of the ABFs investigated in this study, which are based on the following fundamental RBFs: conic, quadratic, Wendland C6, and Gaussian. Complete details of the ABFs based on the conic RBF are provided in the main body of the paper, in Section III B. All fundamental RBFs except the Gaussian have been normalised to yield $W^0(2) = 0$, and

$$2\pi \int_0^2 qW^0(q) dq = 1. \quad (\text{A1})$$

The series of ABFs generated from W^0 are given by

$$\begin{aligned} \mathbf{W}_{ji} = & \frac{dW^0}{dr} \left[\frac{x}{r}, \frac{y}{r}, \frac{y^2}{r^3}, \frac{-xy}{r^3}, \frac{x^2}{r^3}, \frac{-3xy^2}{r^5}, \frac{2x^2y - y^3}{r^5}, \frac{2xy^2 - x^3}{r^5}, \right. \\ & \left. \frac{-3x^2y}{r^5}, \frac{12x^2y^2 - 3y^4}{r^7}, \frac{9xy^3 - 6x^3y}{r^7}, \frac{2x^4 - 11x^2y^2 + 2y^4}{r^7}, \frac{9x^3y - 6xy^3}{r^7}, \frac{12x^2y^2 - 3x^4}{r^7}, \dots \right]^T + \\ & \frac{d^2W^0}{dr^2} \left[0, 0, \frac{x^2}{r^2}, \frac{xy}{r^2}, \frac{y^2}{r^2}, \frac{3xy^2}{r^4}, \frac{y^3 - 2x^2y}{r^4}, \frac{x^3 - 2xy^2}{r^4}, \frac{3x^2y}{r^4}, \dots \right]^T + \\ & \frac{d^3W^0}{dr^3} \left[0, 0, 0, 0, 0, \frac{x^3}{r^3}, \frac{x^2y}{r^3}, \frac{xy^2}{r^3}, \frac{y^3}{r^3}, \dots \right]^T + \frac{d^4W^0}{dr^4} [\dots]^T + \dots \quad (\text{A2}) \end{aligned}$$

In practice we omit the terms in d^2W^0/dr^2 and higher derivatives for all $k \geq 4$ (i.e. for elements 10 and onwards of \mathbf{W}_{ji}), regardless of whether d^2W^0/dr^2 and higher derivatives are zero. We do this because: a) it is computationally simpler, b) it yields a scheme which is more resilient to node distribution disorder.

1. Convex Quadratic

The fundamental RBF is set as $W^0 = W^Q$, where

$$W^Q(q) = \frac{3}{16\pi} (q-2)^2 \quad 0 \leq q \leq 2. \quad (\text{A3})$$

The design of W^Q is such that $dW^Q/dq|_{q=2} = 0$, which appears from our stability analysis to be a necessary condition for stability of the convective operator on a uniform node distribution. W^Q is convex, with positive curvature everywhere and discontinuous gradient (i.e. a sharp peak) at the origin. Note that W^Q is only valid for $q \in [0, 2]$.

2. Wendland C6

The fundamental RBF is set as $W^0 = W^W$, where

$$W^W(q) = \frac{78}{28\pi} \left(1 - \frac{q}{2}\right)^8 (4q^3 + 6.25q^2 + 4q + 1) \quad 0 \leq q \leq 2, \quad (\text{A4})$$

which is the Wendland C6 kernel, (see Dehnen and Aly [17]) commonly used in SPH. Note that W^W is only valid for $q \in [0, 2]$.

3. Gaussian

The fundamental RBF is set as $W^0 = W^G$, where

$$W^G(q) = \frac{9}{\pi} e^{-9q^2}, \quad (\text{A5})$$

which is the classical Gaussian kernel, similar to that shown in [11, 29]. Note that for W^G , satisfaction of (A1) is not exact, but is approximate to within 2×10^{-16} , and $W^G(2) \approx 6.6 \times 10^{-16}$.

-
- [1] S. K. Lele, Compact finite difference schemes with spectral-like resolution, *Journal of Computational Physics* 103 (1992) 16 – 42. doi:doi:10.1016/0021-9991(92)90324-R.
- [2] D. Gottlieb, S. A. Orszag, *Numerical analysis of spectral methods: theory and applications*, volume 26, SIAM, 1977.
- [3] S. Li, W. K. Liu, Meshfree and particle methods and their applications, *Applied Mechanics Reviews* 55 (2002) 1–34. doi:doi:10.1115/1.1431547.

- [4] S. Garg, M. Pant, Meshfree methods: A comprehensive review of applications, *International Journal of Computational Methods* 15 (2018) 1830001. doi:doi:10.1142/S0219876218300015.
- [5] P. S. Jensen, Finite difference techniques for variable grids, *Computers & Structures* 2 (1972) 17 – 29. doi:doi:10.1016/0045-7949(72)90020-X.
- [6] N. Perrone, R. Kao, A general finite difference method for arbitrary meshes, *Computers & Structures* 5 (1975) 45 – 57. doi:doi:10.1016/0045-7949(75)90018-8.
- [7] J. Benito, F. Ureña, L. Gavete, Solving parabolic and hyperbolic equations by the generalized finite difference method, *Journal of Computational and Applied Mathematics* 209 (2007) 208 – 233. doi:doi:10.1016/j.cam.2006.10.090.
- [8] L. Gavete, F. Ureña, J. Benito, A. García, M. Ureña, E. Saete, Solving second order non-linear elliptic partial differential equations using generalized finite difference method, *Journal of Computational and Applied Mathematics* 318 (2017) 378 – 387. doi:doi:10.1016/j.cam.2016.07.025.
- [9] N. Trask, M. Perego, P. Bochev, A High-Order Staggered Meshless Method for Elliptic Problems, *SIAM Journal on Scientific Computing* 39 (2017) A479–A502. doi:doi:10.1137/16M1055992.
- [10] N. Trask, M. Maxey, X. Hu, A compatible high-order meshless method for the stokes equations with applications to suspension flows, *Journal of Computational Physics* 355 (2018) 310 – 326. doi:doi:10.1016/j.jcp.2017.10.039.
- [11] R. A. Gingold, J. J. Monaghan, Smoothed particle hydrodynamics: theory and application to non-spherical stars, *Monthly Notices of the Royal Astronomical Society* 181 (1977) 375–389. doi:doi:10.1093/mnras/181.3.375.
- [12] L. B. Lucy, Numerical approach to the testing of the fission hypothesis, *Astron. J.* 82 (1977) 1013–1024. doi:doi:10.1086/112164.
- [13] J. J. Monaghan, Smoothed Particle Hydrodynamics and its diverse applications, *Annual Review of Fluid Mech.* 44 (2012) 323–346.
- [14] J. Monaghan, Smoothed Particle Hydrodynamics, *Reports on Progress in Physics* 68 (2005) 1703–1759. doi:doi:10.1088/0034-4885/68/8/R01.
- [15] N. J. Quinlan, M. Basa, M. Lastiwka, Truncation error in mesh-free particle methods, *International Journal for Numerical Methods in Engineering* 66 (2006) 2064–2085. doi:doi:10.1002/nme.1617.
- [16] S. J. Lind, R. Xu, P. K. Stansby, B. D. Rogers, Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves, *Journal of Computational Physics* 231 (2012) 1499–1523. doi:doi:10.1016/j.jcp.2011.10.027.
- [17] W. Dehnen, H. Aly, Improving convergence in smoothed particle hydrodynamics simulations without pairing instability, *Monthly Notices of the Royal Astronomical Society* 425 (2012) 1068–1082. doi:doi:10.1111/j.1365-2966.2012.21439.x.
- [18] J. Bonet, T.-S. Lok, Variational and momentum preservation aspects of smooth particle hydrodynamic formulations, *Computer Methods in Applied Mechanics and Engineering* 180 (1999) 97 – 115. doi:doi:10.1016/S0045-7825(99)00051-1.
- [19] W. K. Liu, S. Jun, S. Li, J. Adee, T. Belytschko, Reproducing kernel particle methods for structural dynamics, *International Journal for Numerical Methods in Engineering* 38 (1995) 1655–1679. doi:doi:10.1002/nme.1620381005.
- [20] W. K. Liu, S. Jun, Y. F. Zhang, Reproducing kernel particle methods, *International Journal for Numerical Methods in Fluids* 20 (1995) 1081–1106. doi:doi:10.1002/flid.1650200824.
- [21] J. K. Chen, J. E. Beraun, T. C. Carney, A corrective smoothed particle method for boundary value problems in heat conduction, *International Journal for Numerical Methods in Engineering* 46 (1999) 231–252. doi:doi:10.1002/(SICI)1097-0207(19990920)46:2<231::AID-NME672>3.0.CO;2-K.
- [22] J. Chen, J. Beraun, A generalized smoothed particle hydrodynamics method for nonlinear dynamic problems, *Computer Methods in Applied Mechanics and Engineering* 190 (2000) 225 – 239. doi:doi:10.1016/S0045-7825(99)00422-3.
- [23] G. M. Zhang, R. C. Batra, Modified smoothed particle hydrodynamics method and its application to transient problems, *Computational Mechanics* 34 (2004) 137–146. doi:doi:10.1007/s00466-004-0561-5.
- [24] M. Liu, W. Xie, G. Liu, Modeling incompressible flows using a finite particle method, *Applied Mathematical Modelling* 29 (2005) 1252 – 1270. doi:doi:10.1016/j.apm.2005.05.003.
- [25] M. Liu, G. Liu, Restoring particle consistency in smoothed particle hydrodynamics, *Applied Numerical Mathematics* 56 (2006) 19–36. doi:doi:10.1016/j.apnum.2005.02.012.
- [26] D. Asprone, F. Auricchio, G. Manfredi, A. Prota, A. Reali, G. Sangalli, Particle methods for a 1D elastic model problem: Error analysis and development of a second-order accurate formulation, *Computer Modeling in Engineering & Sciences* 62 (2010) 1–21. doi:doi:10.3970/cmcs.2010.062.001.
- [27] D. Asprone, F. Auricchio, A. Reali, Novel finite particle formulations based on projection methodologies, *International Journal for Numerical Methods in Fluids* 65 (2011) 1376–1388. doi:doi:10.1002/flid.2327.
- [28] S. Sibilla, An algorithm to improve consistency in smoothed particle hydrodynamics, *Computers & Fluids* 118 (2015) 148 – 158. doi:doi:10.1016/j.compfluid.2015.06.012.
- [29] S. Lind, P. Stansby, High-order Eulerian incompressible smoothed particle hydrodynamics with transition to Lagrangian free-surface motion, *Journal of Computational Physics* 326 (2016) 290 – 311. doi:doi:10.1016/j.jcp.2016.08.047.
- [30] G. Fountas, P. Stansby, B. Rogers, S. Lind, An Eulerian-Lagrangian incompressible SPH formulation (ELI-SPH) connected with a sharp interface, *Computer Methods in Applied Mechanics and Engineering* 329 (2018) 532 – 552. doi:doi:10.1016/j.cma.2017.09.029.
- [31] L. N. Trefethen, J. A. C. Weideman, The exponentially convergent trapezoidal rule, *SIAM Review* 56 (2014) 385–458. doi:doi:10.1137/130932132.
- [32] R. L. Hardy, Multiquadric equations of topography and other irregular surfaces, *Journal of Geophysical Research* (1896-1977) 76 (1971) 1905–1915. doi:doi:10.1029/JB076i008p01905.

- [33] E. Kansa, Multiquadrics - A scattered data approximation scheme with applications to computational fluid-dynamics - I surface approximations and partial derivative estimates, *Computers & Mathematics with Applications* 19 (1990) 127 – 145. doi:doi:10.1016/0898-1221(90)90270-T.
- [34] E. Kansa, Multiquadrics - A scattered data approximation scheme with applications to computational fluid-dynamics - II solutions to parabolic, hyperbolic and elliptic partial differential equations, *Computers & Mathematics with Applications* 19 (1990) 147 – 161. doi:doi:10.1016/0898-1221(90)90271-K.
- [35] B. Fornberg, N. Flyer, Solving PDEs with radial basis functions, *Acta Numerica* 24 (2015) 215–258. doi:doi:10.1017/S0962492914000130.
- [36] E. F. Bollig, N. Flyer, G. Erlebacher, Solution to PDEs using radial basis function finite-differences (RBF-FD) on multiple GPUs, *Journal of Computational Physics* 231 (2012) 7133 – 7151. doi:doi:10.1016/j.jcp.2012.06.030.
- [37] C. Shu, H. Ding, K. Yeo, Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier-Stokes equations, *Computer Methods in Applied Mechanics and Engineering* 192 (2003) 941 – 954. doi:doi:10.1016/S0045-7825(02)00618-7.
- [38] T. Cecil, J. Qian, S. Osher, Numerical methods for high dimensional Hamilton-Jacobi equations using radial basis functions, *Journal of Computational Physics* 196 (2004) 327 – 347. doi:doi:10.1016/j.jcp.2003.11.010.
- [39] G. B. Wright, Radial Basis Function Interpolation: Numerical and Analytical Developments, Ph.D. thesis, Boulder, CO, USA, 2003. AAI3087597.
- [40] N. Flyer, B. Fornberg, V. Bayona, G. A. Barnett, On the role of polynomials in RBF-FD approximations: I. interpolation and accuracy, *Journal of Computational Physics* 321 (2016) 21 – 38. doi:doi:10.1016/j.jcp.2016.05.026.
- [41] N. Flyer, G. A. Barnett, L. J. Wicker, Enhancing finite differences with radial basis functions: Experiments on the Navier-Stokes equations, *Journal of Computational Physics* 316 (2016) 39–62. doi:doi:10.1016/j.jcp.2016.02.078.
- [42] H. Wendland, Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, *Advances in Computational Mathematics* 4 (1995) 389–396. doi:doi:10.1007/BF02123482.
- [43] H. Wendland, *Scattered data approximation*, volume 17, Cambridge university press, 2004.
- [44] B. Fornberg, E. Lehto, Stabilization of RBF-generated finite difference methods for convective PDEs, *Journal of Computational Physics* 230 (2011) 2270 – 2285. doi:doi:10.1016/j.jcp.2010.12.014.
- [45] P. K. Mishra, G. E. Fasshauer, M. K. Sen, L. Ling, A stabilized radial basis-finite difference (RBF-FD) method with hybrid kernels, *Computers & Mathematics with Applications* 77 (2019) 2354 – 2368. doi:doi:10.1016/j.camwa.2018.12.027.
- [46] V. Bayona, N. Flyer, B. Fornberg, G. A. Barnett, On the role of polynomials in RBF-FD approximations: II. Numerical solution of elliptic PDEs, *Journal of Computational Physics* 332 (2017) 257 – 273. doi:doi:10.1016/j.jcp.2016.12.008.
- [47] J. D. Cole, On a quasi-linear parabolic equation occurring in aerodynamics, *Quarterly of Applied Mathematics* 9 (1951) 225 – 236. doi:doi:10.1090/qam/42889.
- [48] D. Rempfer, On Boundary Conditions for Incompressible Navier-Stokes Problems, *Applied Mechanics Reviews* 59 (2006) 107–125. doi:doi:10.1115/1.2177683.
- [49] J.-L. Guermond, L. Quartapelle, On stability and convergence of projection methods based on pressure poisson equation, *International Journal for Numerical Methods in Fluids* 26 (1998) 1039–1053. doi:doi:10.1002/(SICI)1097-0363(19980515)26:9<1039::AID-FLD675>3.0.CO;2-U.
- [50] E. Kwan-yu Chiu, Q. Wang, R. Hu, A. Jameson, A conservative mesh-free scheme and generalized framework for conservation laws, *SIAM Journal on Scientific Computing* 34 (2012) A2896–A2916. doi:doi:10.1137/110842740.
- [51] P. F. Hopkins, A new class of accurate, mesh-free hydrodynamic simulation methods, *Monthly Notices of the Royal Astronomical Society* 450 (2015) 53–110. doi:doi:10.1093/mnras/stv195.
- [52] N. Trask, P. Bochev, M. Perego, A conservative, consistent, and scalable meshfree mimetic method, *Journal of Computational Physics* 409 (2020) 109187. doi:doi:10.1016/j.jcp.2019.109187.