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: January 10, 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 eliminate all errors below 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 (4th order with \( N \approx 25 \) nodes, 8th order with \( N \approx 60 \) nodes). At domain boundaries (with incomplete support) LABFM automatically provides one-sided differences of the same order as the internal scheme, up to 4th order. We use the method to solve elliptic, parabolic and mixed hyperbolic-parabolic partial differential equations (PDEs), showing up to 8th 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 vanish. 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, \( N \) 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 moving least squares (GMLS) method, in which a polynomial reconstruction is used to obtain high order approximate

* jack.king@manchester.ac.uk
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], 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), provided \( N \) is held constant (for typical smoothing kernels), but in practice this drops to typically \( \sim 1 \) due to particle disorder. 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 \( N \), and this dependence is influenced by the choice of smoothing kernel. For the widely used Wendland C2 kernel (e.g. \( \mu(\frac{1}{2}) \)), the method approaches the theoretical convergence rate (in \( h \)) of 2 with \( N^{-\frac{5}{2}} \). A number of corrections to SPH have been devised, including by Bonet and Lok [17], whose relatively cheap correction (which requires inversion of a square matrix of rank \( \mu \), where \( \mu \) 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) [18, 19], and corrective SPH (CSPH) [4] 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. [20] and Chen and Beraun [21] 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 [22] and Liu et al. [23] 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 \( 3^{rd} \) order, although the consistency of the approximation deteriorated in the vicinity of boundaries. Liu and Liu [24] 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. [25] developed a modified FPM, with the kernel modified by multiplication with a polynomial to eliminate certain moments, and in [26] extended the idea by replacing the kernel and its derivatives with a series of monomials, demonstrating second order convergence in two dimensions. Building on [24], Sibilla [27] 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 [27] shows a convergence rate which doesn’t deteriorate at very fine resolutions: the order collapses only when machine precision errors start to dominate. Recent work by Lind and Stansby [28] 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 [29]. 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 \( (N \approx 450 \text{ nodes in two dimensions, compared with } N \leq 50 \text{ for typical SPH}) \). The computational cost per particle per time-step is at best \( O(N^2) \) FLOPs (FLoating Point Operations), and so whilst the latter approach can yield convergence to machine precision, it increases the cost over compactly supported kernels.

A featured shared by GFDM, GMLS and the above mentioned SPH-derived methods is that interpolation involves summation over all \( N \) 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 [30], and were first applied to the solution of PDEs in [31, 32], 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 [33], although the cost of the global approach is significant, at \( O(N^2) \) FLOPs per time-step [34]. 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. [35], Cecil et al. [36], Wright [37], where each node only interacts with the nearest \( N \) nodes (and generally \( N \ll N \)). 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 \( N \ll N \), the cost is \( O(N) \) 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 \( \sqrt{N} \). 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 \([38, 39]\). A good review of both global RBF and RBF-FD methods is given by Fornberg and Flyer [33].

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 convergence up to the desired order, on arbitrary node distributions. In this work we demonstrate up to 8th order convergence on disordered node distributions. 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, we can obtain 4th order with \( N \approx 25 \), and 6th order with \( N \approx 48 \), and 8th-order with \( N \approx 75 \) on severely disordered distributions. 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 \( O(N) \) FLOPs per time-step of RBF-FD, and for fourth order and higher, allowing smaller \( N \). In certain conditions LABFM is equivalent (in terms of errors) to the approach of [24, 27], 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.

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 \( \Omega \), with \( r_i = (x_i, y_i) \) 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 \( \delta r \), where typically \( N \delta r^2 = V_\Omega \), with \( V_\Omega \) the volume of the domain. The value of some field \( \phi \) at node \( i \) is \( \phi_i \). When calculating derivatives, we use stencils with finite size with characteristic width \( h \) (in the present work, all stencils have a radius of 2\( h \)). For any node \( i \), we refer to all nodes \( j \) within the stencil of \( i \) as neighbours of \( i \). \( N_i \) is the number of neighbours of \( i \). Sums over \( j \) (e.g. \( \sum_j \)) are over all \( j \in N_i \). In this paper, we consider the case where \( h/\delta r \) is uniform and constant, and hence also \( N_i \) 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
\( (\cdot)_{ij} = (\cdot)_j - (\cdot)_i \), and where the subscript \( j \) appears after a function, the difference is applied to the function arguments. We define a vector of monomials

\[
\mathbf{X} = \left[ x, y, \frac{x^2}{2}, xy, \frac{y^2}{2}, \frac{x^3}{6}, \frac{x^2y}{2}, \frac{xy^2}{2}, \frac{y^3}{6}, \frac{x^4}{24}, \ldots \right]^T, \tag{1}
\]

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

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

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

\[
(\cdot)_j = (\cdot)_i + \mathbf{X}_{ji} \cdot \mathbf{D}(\cdot)|_i, \tag{3}
\]

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 \( i \):

\[
L^d_i(\cdot) = \sum_j (\cdot)_j w^d_{ji}, \tag{4}
\]

where \( d \) identifies the specific partial derivative(s) being approximated, and \( w^d_{ji} \) are the set of weights that, at this stage we require to be functions of \( \mathbf{r}_j \) (i.e. \( w^d_{ji} = w^d(\mathbf{r}_j) \)) 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

\[
\mathbf{C}^d = \begin{cases} [1, 0, 0, 0, 0, 0, 0, \ldots]^T & \text{if } d = x \\ [0, 1, 0, 0, 0, 0, 0, \ldots]^T & \text{if } d = y \\ [0, 0, 1, 0, 0, 0, 0, \ldots]^T & \text{if } d = L. \end{cases} \tag{5}
\]

The choice to use \((\cdot)_{ji}\) in (3) (as opposed to \((\cdot)_{j}\)) allows inter-node weight functions with a singularity at \( w^d_{ji} \). We analyse the error in \( L^d_i(\cdot) \) by substituting the Taylor expansion (3) into (4), obtaining

\[
L^d_i(\cdot) = \sum_j \mathbf{X}_{ji} \cdot \mathbf{D}(\cdot)|_i w^d_{ji}, \tag{6}
\]

which when expanded is

\[
L^d_i(\cdot) = \frac{\partial (\cdot)}{\partial x}|_i \sum_j x_{ji} w^d_{ji} + \frac{\partial (\cdot)}{\partial y}|_i \sum_j y_{ji} w^d_{ji} + \frac{\partial^2 (\cdot)}{\partial x^2}|_i \sum_j \frac{x^2_{ji}}{2} w^d_{ji} + \frac{\partial^2 (\cdot)}{\partial y^2}|_i \sum_j \frac{y^2_{ji}}{2} w^d_{ji} + \frac{\partial^2 (\cdot)}{\partial x \partial y}|_i \sum_j \frac{x y_{ji}}{2} w^d_{ji} + \frac{\partial^3 (\cdot)}{\partial x^3}|_i \sum_j \frac{x^3_{ji}}{6} w^d_{ji} + \frac{\partial^3 (\cdot)}{\partial x^2 \partial y}|_i \sum_j \frac{x^2 y_{ji}}{2} w^d_{ji} + \frac{\partial^3 (\cdot)}{\partial x \partial y^2}|_i \sum_j \frac{x y^2_{ji}}{2} w^d_{ji} + \frac{\partial^3 (\cdot)}{\partial y^3}|_i \sum_j \frac{y^3_{ji}}{6} w^d_{ji} + \frac{\partial^4 (\cdot)}{\partial x^4}|_i \sum_j \frac{x^4_{ji}}{24} w^d_{ji} + \ldots. \tag{7}
\]

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

\[
B^d_i = \sum_j \mathbf{X}_{ji} w^d_{ji}. \tag{8}
\]

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 $\partial (\cdot)/\partial x$)

$$w_{ji}^x = \frac{\partial W_{ji}}{\partial x} \delta V,$$

where $W$ is the SPH smoothing kernel, and $\delta V$ 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 ANISTROPIC BASIS FUNCTION METHOD

A. The elimination of error terms

We now present a method of specifying $w_d$ such that $L_d$ is of arbitrarily high order. We set $w_d$ equal to the weighted sum of a series of Anisotropic Basis Functions (ABFs) $W_{ji} = W(r_{ji}/h)$, writing

$$w_{ji}^d = W_{ji} \cdot \Psi_i^d = W_{ji}^1 \Psi_i^{d,1} + W_{ji}^2 \Psi_i^{d,2} + W_{ji}^3 \Psi_i^{d,3} + W_{ji}^4 \Psi_i^{d,4} + \ldots,$$

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

$$B_i^d = \sum_j X_{ji} W_{ji} \cdot \Psi_i^d,$$

which may be easily reformulated as a linear system

$$M_i \Psi_i^d = B_i^d,$$

in which

$$M_i = \sum_j X_{ji} \otimes W_{ji}.$$

We obtain the weights $\Psi_i^d$ by specifying desired values of each moment, replacing $B_i$ in (12) with $C_d$ and solving the resulting system

$$M_i \Psi_i^d = C_d.$$

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

The above analysis imposes no constraint on the lengths of $X$, $W$, $C$, and the size of $M$, although for consistency between (13) and (14), we require $X$, $W$ and $C$ to be of equal length, and $M$ square. In practice they are finite, and we use only the first $p$ elements from $X$ and $W$ 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 $X$:

$$p = \frac{k^2 + 3k}{2} = 2, 5, 9, 14, 20, 27 \ldots \text{ for } k = 1, 2, 3, 4, 5 \ldots,$$

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 $k + 1$:

$$p = \sum_{m=2}^{m=k+1} m (m + 1)/2 = 3, 9, 19, 34, 55 \ldots \text{ for } k = 1, 2, 3, 4, 5 \ldots.$$
When (14) is solved with \( M_i \) having size \( p \times p \), 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 \( M_i \) 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 highest order of moments corrected for in solving (14) - to identify the order of the scheme used.

The matrix \( M_i \) is the same for all operators (different \( d \)), and so we calculate each \( M_i \) once, then solve (14) for each \( d \) to obtain the required operators. Once (14) has been solved, we calculate the weights \( w^d \) 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 \( W_{ji} \) approximates a different partial derivative, so that if \( w^d_{ji} = W_{ji} \cdot C^d \)

\[
L_i^d(\cdot) = \sum_j (\cdot)_{ji} w^d_{ji} = \sum_j (\cdot)_{ji} W_{ji} \cdot C^d \approx D(\cdot)_{ij} \cdot C^d,
\]

(17)

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

\[
\langle f(r_{ji}), g(r_{ji}) \rangle = \frac{1}{V_i} \int_{V_i} f(r_{ji}) g(r_{ji}) \, dV_j,
\]

(18)

where \( f \) and \( g \) are any two functions of \( r \). Note that the moments \( B^d_{ij} \) in (8) are constructed from the discrete form of \( \langle X_{ji}, w^d_{ji} \rangle \). We denote the \( n^{th} \) element of \( X_{ji} \) as \( X^m_{ji} \) and the \( n^{th} \) ABF in \( W_{ji} \) as \( W^n_{ji} \).

If the ABFs are chosen such that \( (W^m, X^m) \) 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 \( X_{ji} \) scale with \( h^2, h^3, h^4 \) 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 \( M \) is independent of resolution. This requirement may be satisfied by constructing the ABFs from the partial derivatives of some appropriately scaled fundamental RBF \( W^0 = W^0(\|r_{ji}\|) \), according to

\[
W_{ji} = D(W^0)|_{r_{ji}},
\]

(19)

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

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

(20)

where \( q = \|r_{ji}\|/h = r_{ji}/h \), 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 \( D \) to \( W^0 \):

\[
W_{ji} = D(W^0)|_{r_{ji}} = \left[ \begin{array}{c} 1 - \frac{q}{2} \frac{y}{r^3} + \frac{y^2}{r^6}, \frac{r^3}{x^2} - \frac{y^2}{r^6}, \frac{r^3}{x^3} - \frac{2x^2y - y^3}{r^5}, \frac{2xy^2 - x^3}{r^5}, \\
\frac{-3x^2y}{r^6}, \frac{12x^2y^2 - 3y^4}{r^7}, \frac{9xy^3 - 6x^3y}{r^7}, \frac{2x^4 - 11x^2y^2 + 2y^4}{r^7}, \frac{9x^3y - 6xy^3}{r^7}, \frac{12x^2y^2 - 3x^4}{r^7} \end{array} \right]^T,
\]

(21)

in which the subscripts \( ji \) have been dropped from \( x_{ji}, y_{ji} \), and \( r_{ji} \) for ease of exposition. Figures 2 and 3 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 \( r_{ji} = 0 \). The first order ABFs (Figure 1) 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 [10], 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 [27].

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 $M_i$ are constructed must adequately approximate continuous integrals:

$$\sum_j X^m_{ji} W^n_{ji} \approx \langle X^m_{ji}, W^n_{ji} \rangle \quad \forall m, n \in [1, p].$$ \hspace{2cm} (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 $N$) below which one or more of the eigenvalues of $M_i$ has magnitude of machine precision zero, and below which the condition number of $M_i$ 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 $N_{crit} \approx \{8, 21, 37, 57\}$, and are (on average) independent of disorder of the node distribution. It is clear from Figures 1, 2, and 3, 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 $\theta_{ji}$ having origin at $r_i$), 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 $\pi/k$), there is a node in every segment. This criteria provides a more accurate definition of the word “adequately” use for (22), and the values of $h/\delta r_{crit}$ and $N_{crit}$ correspond to the minimum values for which the Nyquist criteria is satisfied.

Figure 4 illustrates an example of the matrix $M_i$ 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, $M$ has a clear structure. The entries 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 $X_{ji}$ and $W_{ji}$ with $h$. For smaller $k$, the matrices $M_i$ 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 ($N_i$ and $h/\delta r$) further in Section IVB below.

IV. CONVERGENCE AND STABILITY

We now analyse the convergence properties of LABFM, and present a brief stability analysis.
We consider a square domain defined by \((x, y) \in [0, H] \times [0, H]\). We define the test function

\[
\phi(\hat{x}, \hat{y}) = 1.0 + (\hat{x}\hat{y})^4 + \sum_{n=1}^{6} (\hat{x}^n + \hat{y}^n),
\]

(23)

where \(\hat{x} = x - 0.1453H\) and \(\hat{y} = y - 0.16401H\), and set \(H = 1\). This offset is pseudo-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 \(\delta r\), and randomly perturbed. The maximum perturbation distance is \(\varepsilon\), and hence we take \(\varepsilon/\delta r\) as a measure of the distribution’s irregularity. Initially we set \(h = 2\delta r\) which, with a support radius of \(2h\), yields \(N_i \approx 50\). An example node distribution and computational stencil is depicted in Figure 5.

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 5) 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 \(\phi\) for a range of resolutions. As a measure of the error we take the normalised \(L_2\)-norm:

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

(24)

Figure 6 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 5), 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 \(\phi\). 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 \(\phi\) (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 \(M\), and carried through the solution of (14).

For other test functions (e.g. a sinusoidally varying \(\phi\)) with \(k = 8\) the convergence rates of 8 and 7 for gradients and Laplacian hold. For clarity, we only show results were obtained using the conic ABFs in Figure 6. Quadratic ABFs would yield the same convergence rates, and error magnitudes approximately 32% lower.

Figure 7 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 7 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 7) 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 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 7 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 \( k = 6 \).
FIG. 5. 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. 6. 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.

Figure 8 shows the variation of the $L_2$-norm with resolution $h/H$ and stencil size $h/\delta r \propto \sqrt{N}$, 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 $N$ as in RBF methods. For reference, a value of $h/\delta r = 2$ corresponds to $N \approx 50$. 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 not adequately sample the ABFs, leading to ill conditioned matrices $M_i$ 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. $k = 2$)
FIG. 7. \(L_2\)-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. 8. 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 \(\varepsilon/\delta r = 0.5\).
use a significantly smaller stencils, with $h/\delta r = 1$, reducing computational costs. For $k = 8$, we observe $8^{th}$ order convergence with $h/\delta r = 2.5 \ (N \approx 78)$. 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 8. With this in mind, a potential approach when applying the method to practical problems is to have a region of higher resolution (smaller $\delta r$, 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 $A^d$ (following the procedure described in detail in Section 3B). The eigenvalues of $A^d$ provide information about the stability of the method. For convective derivatives (e.g. $d = x, y$ to approximate $\partial (\cdot)/\partial x$ or $\partial (\cdot)/\partial y$), no modes should grow or decay, and hence the eigenvalues $\lambda$ of $A^x$ and $A^y$ must (ideally) lie on imaginary axis ($\text{Re}(\lambda) = 0 \ \forall \lambda$). Figure 9 shows the eigenvalues of $A^x$ (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 $A^x$ 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 9). 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 $A^x$ to be zero is that $dW^0/d\epsilon = 0$ 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 $A^x$ 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 $A^x$ fall closer to the imaginary line for smaller $k$ (the blue dots in Figure 9 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 $A^L$ should be as close as possible to the negative real axis, corresponding to decay of all modes. Figure 10 shows the eigenvalues of $A^L$ 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 $\epsilon/\delta r$. 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 10. For the Wendland and Gaussian ABFs, we find that $\text{Re}(\lambda) > 0$ for at lower values of $\epsilon/\delta r$. 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 $k$). 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 $\epsilon/\delta r$ tested. For large $k$, the Laplacian is stable for a range of $\epsilon/\delta r$, provided a large enough $h/\delta r$ is chosen. Values of $\epsilon/\delta r$ 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 3B. 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 $\epsilon/\delta r$ than the Wendland and quadratic ABFs, with the quadratic ABFs being stable for the greatest range of $\epsilon/\delta r$. The conic ABFs (not shown in Table I) exhibit similar maximum stable values of $\epsilon/\delta r$ to the quadratic ABFs.

For purely convective problems, all choices of ABF will be unstable on noisy node distributions, given the non-zero 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 (9, 11), which provides the required dissipation to stabilise the solution. Hyperviscosity (i.e. $-\nabla^4$ or $-\nabla^6$), 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 $C^{hv}$ used in the RHS of (14) appropriately. The coefficient of hyperviscosity $\alpha$ 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

$$C^{hv} = \begin{bmatrix} 0, & 0, & \ldots & -\alpha, & 0, & -2\alpha, & 0, & -\alpha, & 0 & \ldots \end{bmatrix}^T,$$

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 $\alpha$, 3$\alpha$, 3$\alpha$, and $\alpha$ respectively. Note that in LABFM we can only create hyperviscosity operators up to $\nabla^8$. 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 a 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

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

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

$$\frac{\partial u}{\partial t} = \kappa \nabla^2 u,$$

where $\kappa$ 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

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

We discretise the domain with a noisy Cartesian distribution, with the noise having magnitude $\varepsilon/\delta r = 0.5$, where $\delta r$ 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 $\kappa = 1$.

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 11. 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 11. 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 $A^L$, 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

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

(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.
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

$$\nabla^2 \phi = f(x, y)$$  \hspace{1cm} (30)

in the domain $\Omega$, with boundary conditions

$$\phi = g_1(x, y) \quad \text{on} \quad \Gamma_D$$  \hspace{1cm} (31a)

$$\frac{\partial \phi}{\partial n} = g_2(x, y) \quad \text{on} \quad \Gamma_N,$$  \hspace{1cm} (31b)

where the boundary $\Gamma = \Gamma_D + \Gamma_N$. To solve (30) we construct a (linear) global discrete Laplacian operator $A_{d=L} = A^L$ (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 $A^L$ is a rearrangement of the local operator $L_i^L$, which is achieved by setting the elements of $A^L$ as

$$A_{i,j}^L = w_{ji}^L \quad \forall j \neq i$$  \hspace{1cm} (32a)

$$A_{i,i}^L = -\sum_j w_{ji}^L.$$  \hspace{1cm} (32b)

The discretised form of (30) is then

$$A^L \Phi = F,$$  \hspace{1cm} (33)

with solution vector

$$\Phi = [\phi_1, \phi_2, \ldots, \phi_N]^T$$  \hspace{1cm} (34)

and source vector

$$F = [f_1, f_2, \ldots, f_N]^T.$$  \hspace{1cm} (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 $A^L$ and $F$. On sections of the boundary $\Gamma_D$, we satisfy (31a) (on an example boundary node $b$) by setting

$$A_{b,j}^L = 0 \quad \forall j \neq b$$  \hspace{1cm} (36a)

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

$$F_b = g_1,b,$$  \hspace{1cm} (36c)

which ensures that the solution to (33) yields $\phi_b = g_1,b$. On $\Gamma_N$, 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 $\Gamma_D$, we introduce an additional “ghost” node $\hat{b}$, such that $r_{\hat{b}} = r_b - n_b \delta r$, where $n_b = (n_{b,x}, n_{b,y})$ is the unit normal pointing into the domain at parent node $b$ and $\delta r$ is the initial node spacing. We increase the size of the linear system (33) to include the ghost nodes. The rows of $A^L$ corresponding to ghost nodes $\hat{b}$ are used to enforce (31b), by ensuring the discrete boundary normal derivative operator at node $b$ satisfies (31b)

$$L_b^n = \sum_j (\cdot)_j (n_{b,x} w_{x,bj} + n_{b,y} w_{y,bj}) = g_{2,b}.$$  \hspace{1cm} (37)

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

$$A_{\hat{b},j}^L = n_{b,x} w_{x,bj} + n_{b,y} w_{y,bj} \quad \forall j \neq b$$  \hspace{1cm} (38a)

$$A_{\hat{b},b}^L = -\sum_j (n_{b,x} w_{x,bj} + n_{b,y} w_{y,bj})$$  \hspace{1cm} (38b)

$$F_{\hat{b}} = g_{2,b},$$  \hspace{1cm} (38c)
FIG. 12. 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 $h/\delta r = 3$.

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 $\Gamma_N$. We note that this approach to Neumann boundary conditions appears (coincidentally) similar to that described by Mishra et al. [42].

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 12 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 $A^L$ 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 13. We solve (30) with the source term

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

(39)

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

$$\phi = r \sin 4\pi r \cos 3\theta,$$

(40)

and shown in the left panel of Figure 13. The node distribution is generated by discretising the boundaries (placing nodes at intervals $\delta r$ 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.
FIG. 13. The domain and solution for Poisson problem 2 (left), with a close up of a typical node distribution (right).

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   | $3.5 \times 10^{-2}$ (2.01) | $3.7 \times 10^{-2}$ (2.03) | $2.9 \times 10^{-3}$ (5.28) |
| 2/97   | $9.1 \times 10^{-3}$ (1.98) | $9.5 \times 10^{-3}$ (2.01) | $1.4 \times 10^{-4}$ (4.44) |
| 2/193  | $2.3 \times 10^{-3}$ (1.98) | $2.4 \times 10^{-3}$ (2.00) | $6.5 \times 10^{-6}$ (4.47) |
| 2/385  | $5.9 \times 10^{-4}$ (1.99) | $6.0 \times 10^{-4}$ (2.00) | $3.6 \times 10^{-7}$ (4.20) |

To shift nodes according to:

$$r_i^{n+1} = r_i^n + \frac{\delta y^2}{h} \sum_{j \in |r_{ji}| < h} \left( \frac{|r_{ji}|}{h} - 1 \right) \frac{r_{ji}}{|r_{ji}|}.$$  \hspace{1cm} (41)

We apply typically 10 iterations of (41) to obtain the final node distribution, an example of which is shown in Figure 13 (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 $\Gamma_N$ 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. [43], 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

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

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

$$\delta t = \min \left( \frac{0.2h}{\max|u|}, 0.05h^2Re \right).$$  \hspace{1cm} (43)

We solve $\text{(42)}$ 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

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

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

The numerical solution at $t = 1$ is shown in Figure 14. Figure 15 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 $10 \times 10$ over the domain, and yet the relative error is less than $10^{-8}$.

2. Periodic test case

Next we solve $\text{(42)}$ on a periodic square domain $(x, y) \in [0, H] \times [0, H]$, with sinusoidal initial conditions: $u(x, y, 0) = \sin \left(\frac{2\pi x}{H}\right)$ 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 [44], and has since been used as a benchmark for numerical methods by numerous authors, for example [27]. The solution is given by

\[
    u(x,t) = 4\pi H Re \left[ \sum_{n=1}^{\infty} nA_n \sin \left( \frac{2n\pi x}{H} \right) e^{\left( -\frac{n^2\pi^2}{4H^2} t \right)} / A_0 + \sum_{n=1}^{\infty} A_n \cos \left( \frac{2n\pi x}{H} \right) e^{\left( -\frac{n^2\pi^2}{4H^2} t \right)} \right] \tag{45}
\]

where

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

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 16 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 17 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 17). Generally we observed the quadratic ABFs to be slightly more accurate, and slightly more stable (for this problem) than the conic ABFs.

Figure 18 (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 simulation remains stable. The right panel of Figure 18 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).
FIG. 16. 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. 17. 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 $\varepsilon/\delta r = 0.2$.

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 II B we are able to stabilise the simulations, whilst retaining relatively good accuracy. Figure 19 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 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
\[ u = \sin \left( \frac{2\pi x}{H} \right) \sin \left( \frac{2\pi y}{H} \right); \quad v = -\cos \left( \frac{2\pi x}{H} \right) \cos \left( \frac{2\pi y}{H} \right) \] (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 20 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...
FIG. 20. 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 $k = 6$.

of Figure 20, the width of the shock is approximately $h$, and despite this under-resolution, hyperviscosity provides stability without significant smoothing of the solution.

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

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

(48)

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

$$\nabla^2 \psi = -\omega.$$ 

(49)

We integrate (48) using RK4, and after each substep, solve (49) to obtain $\psi$, then calculate gradients of $\psi$ 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

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

(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 21 shows the solution at times $t = 0$ and $t = 1$. For $Re = 100$, the maximum vorticity decays from $4\pi$ (approximately $12.6$) to $5.7$. Figure 22 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 22 is twice as coarse as the results shown in Figure 21. 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 $Re = 1000$. 

![FIG. 20](image-url)
FIG. 21. 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. 22. 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.

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. 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.

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. A key strength of the method is that convergence rates and stability
are unaffected by local node distribution anisotropy up to quite severe levels of disorder. This characteristic will allow
the method to remain extremely accurate even in complex geometries where non-uniformity in the node distribution is
unavoidable. Although the focus of this work is on PDEs containing some diffusion, the ease with which hyperviscosity
may be implemented suggests LABFM has promise for solution of advection dominated problems. The method has
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 LABFM generates high order finite differences, by construction it collapses to SPH in the low
order limit, and thus has potential to be incorporated into an adaptive Eulerian-Lagrangian scheme. 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 to the Leverhulme Trust for funding this work via Research Project Grant RPG-2017-144. We
thank Georgios Fourtakas for several useful discussions regarding node distributions.

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

\[
2\pi \int_0^2 q W^0 (q) dq = 1. \tag{A1}
\]

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

\[
W_{ji} = \frac{dW^0}{dr} \left[ \frac{x}{r^3}, \frac{y}{r^3}, \frac{y^2}{r^3}, \frac{2x^2 - y^2}{r^5}, \frac{x^2 y - y^3}{r^5}, \frac{3xy^2}{r^5}, \frac{2x^2 y - y^3}{r^5}, \frac{x^2 y - y^3}{r^5}, \frac{3xy^2}{r^5}, \frac{2x^2 y - y^3}{r^5}, \frac{x^2 y - y^3}{r^5}, \frac{3xy^2}{r^5}, \ldots \right]^T + \\
\frac{d^2W^0}{dr^2} \left[ 0, 0, \frac{x^2}{r^2}, \frac{xy}{r^2}, \frac{y^2}{r^2}, \frac{3x^2 y}{r^4}, \frac{2x^2 y - y^3}{r^4}, \frac{x^3 - x^2 y^2}{r^4}, \frac{3x^2 y}{r^4}, \frac{2x^2 y - y^3}{r^4}, \frac{x^3 - x^2 y^2}{r^4}, \frac{3x^2 y}{r^4}, \ldots \right]^T + \\
\frac{d^3W^0}{dr^3} \left[ 0, 0, 0, 0, \frac{x^3}{r^3}, \frac{x^2 y}{r^3}, \frac{xy^2}{r^3}, \frac{y^3}{r^3}, \ldots \right]^T + \frac{d^4W^0}{dr^4} \left[ \ldots \right]^T + \ldots \tag{A2}
\]

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
\( W_{ji} \)), 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

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

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

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

which is the Wendland C6 kernel, (see Dehnen and Aly [16]) 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

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

which is the classical Gaussian kernel, similar to that shown in [11, 28]. Note that for $W^G$, satisfaction of (A1) is not exact, but is approximate to within $2 \times 10^{-16}$, and $W^G(2) \approx 6.6 \times 10^{-16}$.
