Scattered Node Compact Finite Difference-Type Formulas Generated from Radial Basis Functions

Grady B. Wright [a,*,1], Bengt Fornberg [b,2]

[a] *Department of Mathematics, University of Utah, Salt Lake City, UT 84112-0090, USA*

[b] *Department of Applied Mathematics, University of Colorado, Boulder, CO 80309-0526, USA*

Abstract

In standard equispaced finite difference (FD) formulas, symmetries can make the order of accuracy relatively high compared to the number of nodes in the FD stencil. With *scattered* nodes, such symmetries are no longer available. The generalization of compact FD formulas that we propose for scattered nodes and radial basis functions (RBFs) achieves the goal of still keeping the number of stencil nodes small without a similar reduction in accuracy. We analyze the accuracy of these new compact RBF-FD formulas by applying them to some model problems, and study the effects of the shape parameter that arises in, for example, the multiquadric radial function.

Key words: Radial basis functions, partial differential equations, finite difference method, compact, *Mehrstellenverfahren*, mesh-free
1991 MSC: 41A21, 41A30, 41A63, 65D25, 65N06

1 Introduction

Radial basis functions (RBFs) are a primary tool for interpolating multidimensional scattered data. In the past decade or so they have also received

[*] Corresponding author.
Email addresses: wright@math.utah.edu (Grady B. Wright), fornberg@colorado.edu (Bengt Fornberg).

[1] The work was supported by NSF VIGRE grant DMS-0091675.
[2] The work was supported by NSF grants DMS-9810751 (VIGRE) and DMS-0309803.

increased attention as a "mesh-free" method for numerically solving partial differential equations (PDEs) on irregular domains by a global collocation approach (see, for example, [1–4]). While these methods can be spectrally accurate (when proper attention is paid to boundaries), they generally result in having to solve a large, ill-conditioned, dense linear system. Some attempts have been made to resolve this problem [3,5,6] (and the references therein). However, stability issues have limited the use of RBFs for time dependent problems and adapting the methods for non-linear equations has proven to be difficult. To combat *all* of these problems, a "local" method has recently been proposed that gives up spectral accuracy for a sparse, better-conditioned linear system and more flexibility for handling non-linearities. This new "mesh-free" method is essentially a generalization of the classical finite difference (FD) method to scattered node layouts and appears to have been investigated independently by Shu et al. [7,8], Tolstykh et al. [9], Cecil et al. [10], and the present authors [11].

In its most general form, the FD method consists of approximating some derivative of a function u at a given point based on a linear combination of the value of u at some surrounding nodes. In the classical 1-D case, u is given at n equispaced nodes, $\{x_i\}_{i=1}^n$, and the k^{th} derivative of u, at say $x = x_j$ $(1 \leq j \leq n)$, is approximated by the FD formula

$$\left.\frac{d^k u}{dx^k}\right|_{x=x_j} = u^{(k)}(x_j) \approx \sum_{i=1}^{n} c_{j,i}^k u(x_i) \ , \tag{1}$$

where $c_{j,i}^k$ are called the FD weights and are usually computed using polynomial interpolation [12,13]. These 1-D formulas can be combined to create FD formulas for partial derivatives in two and higher dimensions. This strategy, however, requires that the nodes of the stencils are situated on some kind of structured grid (or collection of structured grids), which severely limits the geometric flexibility of the FD method.

One obvious approach for bypassing this problem is to instead allow the nodes of the FD stencil to be placed freely, so that a good discretization of the physical domain can be obtained. However, this approach also raises the question of how the weights of the scattered node FD formulas should be computed. Abgrall [14] and Schönauer and Adolph [15], for example, propose extending the classical polynomial interpolation technique. This idea, however, has not become widely used, partly because it leads to several ambiguities about how to generate the FD formulas. For example, what mixed terms should be included in the multivariate polynomial interpolant to the nodes, and how should a set of nodes that leads to a singular polynomial interpolation problem be handled (polynomial interpolation in > 1-D is not well-posed [16]). All of these ambiguities can be resolved if RBF interpolation is instead used to generate the weights in the FD formula [7,9–11].

An RBF interpolant is formed by taking a linear combination of translates of a single univariate basis function $\phi(r)$ that is radially symmetric about its center. Thus, no decisions need to be made about what mixed terms to include in the interpolant. Moreover, for the appropriate choice of $\phi(r)$, the RBF interpolation method is well-posed in all dimensions. The following are some other properties that make RBF interpolants an attractive choice:

- Since the only geometric property used by an RBF interpolant is the pairwise distances between points, higher dimensions do not increase the coding complexity associated with computing the interpolants.
- RBF interpolants can be very accurate at approximating derivatives [17,18].
- Certain types of radial functions $\phi(r)$ feature a "shape" parameter that allows them to vary from being nearly flat to sharply peaked. Recently it has been shown [19–22] that, under some mild restrictions on $\phi(r)$, the RBF interpolants converge to polynomial interpolants in the limit of $\phi(r)$ becoming entirely flat. Thus, all classical FD formulas can be recovered by the limiting RBF interpolant (when the nodes are arranged accordingly).

We refer to this method of using RBF interpolants to generate the weights of a FD formula as the RBF-FD method.

One of the issues with using scattered node FD formulas (whether the weights are generated by polynomials or RBFs) is that symmetries cannot be exploited to increase the accuracy of the formulas. Thus, the number of nodes in the stencils tend to be relatively large compared to the resulting accuracy. To circumvent this problem, we propose a generalization of a method introduced by Collatz [23] under the name *Mehrstellenverfahren*, and later developed into compact FD formulas by Lele [24]. The basic idea behind this method is to keep the stencil size fixed and to also include in the FD formula a linear combination of derivatives of u at surrounding nodes. For example, the accuracy of the 1-D formula (1) can be increased with a formula of the form

$$\left.\frac{d^k u}{dx^k}\right|_{x=x_j} = u^{(k)}(x_j) \approx \sum_{\substack{i=1\\ i\neq j}}^{n} \tilde{c}^{k}_{j,i} u^{(k)}(x_i) + \sum_{i=1}^{n} c^{k}_{j,i} u(x_i) \ . \tag{2}$$

In the case of 1-D and equispaced nodes, the weights $\tilde{c}^{k}_{j,i}$ and $c^{k}_{j,i}$ can be conveniently derived using Padé approximants [13]. For scattered nodes, and for RBFs, this Padé approach is no longer available. Instead, we propose a method based on Hermite RBF interpolation. We refer to our proposed compact FD method as the RBF-HFD method.

The remainder of the paper is structured as follows: In Section 2, we review some properties of the standard and Hermite RBF interpolation methods and introduce some closed form expressions for the cardinal RBF interpolants, which prove valuable for deriving the RBF-FD and HFD formulas in Section

3. In Section 4, we study the effect of the shape parameter on the RBF-FD and HFD formulas. In particular, we are interested in how the formulas behave in the limit of flat radial functions. In Section 5, we illustrate the effectiveness of the RBF-HFD formulas for solving some elliptic PDEs and discuss various implementation details. We conclude with some remarks in Section 6.

2 RBF interpolation

Since the RBF-FD formulas are obtained from RBF interpolants, we review in this section some properties of RBF interpolation. We discuss standard RBF interpolation in which only function values are specified. This forms the background for Hermite RBF interpolation, in which function and derivative values are specified.

2.1 Standard RBF interpolation

Given a set of distinct nodes $\underline{x}_i \in \mathbb{R}^d$, $i = 1, \ldots, n$, and corresponding (scalar) function values $u(\underline{x}_i)$, $i = 1, \ldots, n$, the standard RBF interpolation problem is to find an interpolant of the form

$$s(\underline{x}) = \sum_{i=1}^{n} \lambda_i \ \phi(\|\underline{x} - \underline{x}_i\|) \ + \sum_{j=1}^{m} \beta_j p_j(\underline{x}), \tag{3}$$

where $\phi(r)$ is some radial function, $\| \cdot \|$ is the standard Euclidean norm, and $\{p_k(\underline{x})\}_{k=1}^{m}$ is a basis for the space of all d-variate polynomials that have degree $\leq Q$. The expansion coefficients λ_i and β_j are determined by enforcing the conditions

$$s(\underline{x}_i) = u(\underline{x}_i), \ \ i = 1, \ldots, n, \ \text{and} \tag{4}$$

$$\sum_{i=1}^{n} \lambda_i p_j(\underline{x}_i) = 0, \ \ j = 1, \ldots, m. \tag{5}$$

The degree of the augmented polynomial in (3) depends on $\phi(r)$ and its inclusion may be necessary to guarantee a well-posed interpolation problem [25]. Table 1 lists a few of the many available choices for $\phi(r)$.

In this study, we are primarily interested in the infinitely smooth radial functions since, by suitable choices of the shape parameter ε, they can provide more accurate interpolants than the piecewise smooth case [26,27]. We postpone the discussion on the effect of ε until Section 4. For now, we assume it is fixed at some non-zero value.

Type of basis function	$\phi(r)$
Piecewise smooth RBFs	
Generalized Duchon spline	$r^{2k}\log r,\ \ k \in \mathbb{N},$ or
	$r^{2\nu},\ \ \nu > 0$ and $\nu \notin \mathbb{N}$
Wendland	$(1-r)_+^k p(r),\ \ p$ a polynomial, $k \in \mathbb{N}$
Infinitely smooth RBFs	
Gaussian (GA)	$e^{-(\varepsilon r)^2}$
Generalized multiquadric	$(1+(\varepsilon r)^2)^{\nu/2},\ \ \nu \neq 0$ and $\nu \notin 2\mathbb{N}$
• Multiquadric (MQ)	$(1+(\varepsilon r)^2)^{1/2}$
• Inverse multiquadric (IMQ)	$(1+(\varepsilon r)^2)^{-1/2}$
• Inverse quadratic (IQ)	$(1+(\varepsilon r)^2)^{-1}$

Table 1
Some commonly used radial basis functions. Note: in all cases, $\varepsilon > 0$.

Although (3) is well-posed without any polynomial augmentation for the GA, MQ, IMQ, and IQ RBFs in Table 1, we set $Q = 0$ (i.e. $m = 1$) in order to impose the condition that the RBF-FD formulas are exact for constants. Thus, the standard RBF interpolant that we consider has the form

$$s(\underline{x}) = \sum_{i=1}^{n} \lambda_i \ \phi(\|\underline{x} - \underline{x}_i\|) \ + \beta \ , \tag{6}$$

Imposing the conditions (4) and (5) leads to the symmetric, block linear system of equations

$$\underbrace{\begin{bmatrix} \Phi & \mathbf{e} \\ \mathbf{e}^T & 0 \end{bmatrix}}_{A} \begin{bmatrix} \boldsymbol{\lambda} \\ \beta \end{bmatrix} = \begin{bmatrix} \mathbf{u} \\ 0 \end{bmatrix} \ , \tag{7}$$

where $\Phi_{i,j} = \phi(\|\underline{x}_i - \underline{x}_j\|)$, $i, j = 1, \ldots, n$, and $\mathbf{e}_i = 1$, $i = 1, \ldots, n$.

When deriving RBF-FD formulas in Section 3, we make use of the Lagrange form the RBF interpolant (6):

$$s(\underline{x}) = \sum_{i=1}^{n} \psi_i(\underline{x}) u(\underline{x}_i) \ , \tag{8}$$

where $\psi_i(\underline{x})$ are of the form (6) and satisfy the cardinal conditions

$$\psi_i(\underline{x}_k) = \begin{cases} 1 & \text{if } k = i \\ 0 & \text{if } k \neq i \end{cases}, \quad k = 1, \ldots, n \ . \tag{9}$$

There turns out to be a nice closed-form expression for $\psi_i(\underline{x})$ that first appeared in [20]. Denote by $A_k(\underline{x})$, $k = 1, \ldots, n$, the matrix A in (7) with the

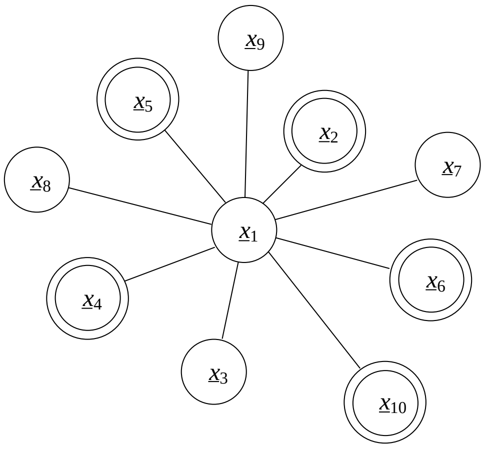


Fig. 1. Example of a scattered node Hermite interpolation problem.

k^{th} row replaced by the vector

$$B(\underline{x}) = \Big[\, \phi(\|\underline{x} - \underline{x}_1\|) \;\; \phi(\|\underline{x} - \underline{x}_2\|) \;\; \cdots \;\; \phi(\|\underline{x} - \underline{x}_n\|) \Big| \;\; 1 \Big] \, . \tag{10}$$

Note that with this notation $A = A_k(\underline{x}_k)$.

Theorem 1 ([20]) *The cardinal RBF interpolant of the form (6) that satisfies (9) is given by*

$$\psi_i(\underline{x}) = \frac{\det\left(A_i(\underline{x})\right)}{\det\left(A\right)} \, . \tag{11}$$

While (11) is not recommended for computational work, it is useful for analytically studying the effects of the shape parameter ε in Section 4.

2.2 Hermite RBF interpolation

Let $\mathcal{L}$ be some linear differential operator (e.g., the Laplacian, $\mathcal{L} = \Delta$) and let σ be a vector containing some combination of $m \leq n$ distinct numbers from the set $\{1, \dots, n\}$. Then the Hermite interpolation problem we consider is to find a function $s(\underline{x})$ that interpolates $u(\underline{x})$ at the distinct nodes $\underline{x}_i$, $i = 1, \dots, n$, and interpolates $\mathcal{L}u(\underline{x})$ at $\underline{x}_{\sigma_j}$, $j = 1, \dots, m$. To clarify the notation, suppose u is given at the nodes $\underline{x}_1, \underline{x}_2, \dots, \underline{x}_{10}$ in Figure 1 and $\mathcal{L}u$ is given at the nodes with double circles (i.e. $\underline{x}_2, \underline{x}_4, \underline{x}_5, \underline{x}_6, \underline{x}_{10}$). Then using the notation of the problem $n = 10$, $m = 5$, and $\sigma = \{2, 4, 5, 6, 10\}$.

Solutions to this Hermite problem with RBFs have been around since Hardy's introduction of the MQ method [28] (see also [29]). Other expositions on this and related Hermite-type RBF problems can be found, for example, in [30–32]. The Hermite RBF method we will use is similar to the Hermite-Birkhoff method proposed by Wu [32]. The idea is to introduce m more unknowns to (6) in order to satisfy the m new interpolation conditions. Formally, the interpolant takes the form

$$s(\underline{x}) = \sum_{i=1}^{n} \lambda_i \phi(\|\underline{x} - \underline{x}_i\|) \ + \sum_{j=1}^{m} \alpha_j \mathcal{L}_2 \phi(\|\underline{x} - \underline{x}_{\sigma_j}\|) \ + \beta \ , \tag{12}$$

where

$$\mathcal{L}_2 \phi(\|\underline{x} - \underline{x}_j\|) := \mathcal{L}\phi(\|\underline{x} - \underline{y}\|)\Big|_{\underline{y}=\underline{x}_j} \ ,$$

i.e. $\mathcal{L}$ acts on ϕ as a function of the second variable $\underline{y}$. For the remainder of this article we will also use the notation

$$\mathcal{L}\phi(\|\underline{x}_j - \underline{y}\|) = \mathcal{L}_1\phi(\|\underline{x}_j - \underline{y}\|) := \mathcal{L}\phi(\|\underline{x} - \underline{y}\|)\Big|_{\underline{x}=\underline{x}_j} \ ,$$

i.e. if $\mathcal{L}$ or $\mathcal{L}_1$ act on ϕ, then the operator applies to the first variable. For example, if $d = 1$ and $\mathcal{L} = \frac{d}{dx}$ then $\mathcal{L}_2\phi(\|x - x_j\|) = -\mathcal{L}_1\phi(\|x - x_j\|) = -\mathcal{L}\phi(\|x - x_j\|)$.

Imposing the interpolation conditions (4), (5), and

$$\mathcal{L}s(\underline{x}_{\sigma_j}) = \mathcal{L}u(\underline{x}_{\sigma_j}), \ j = 1, \ldots, m, \tag{13}$$

leads to the following block linear system of equations

$$\underbrace{\begin{bmatrix} \Phi & \Phi^{\mathcal{L}_2} & \mathbf{e} \\ \Phi^{\mathcal{L}_1} & \Phi^{\mathcal{L}_1\mathcal{L}_2} & \mathbf{0} \\ \mathbf{e}^T & \mathbf{0}^T & 0 \end{bmatrix}}_{A^{\mathrm{H}}} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \\ \beta \end{bmatrix} = \begin{bmatrix} \mathbf{u} \\ \mathcal{L}\mathbf{u} \\ 0 \end{bmatrix} \ , \tag{14}$$

where Φ and $\mathbf{e}$ are given by (7) and

$$\begin{aligned} \Phi^{\mathcal{L}_2}_{i,j} &= \mathcal{L}_2\phi(\|\underline{x}_i - \underline{x}_{\sigma_j}\|) \ , & i = 1, \ldots, n, \ j = 1, \ldots, m, \\ \Phi^{\mathcal{L}_1}_{i,j} &= \mathcal{L}_1\phi(\|\underline{x}_{\sigma_i} - \underline{x}_j\|) \ , & i = 1, \ldots, m, \ j = 1, \ldots, n, \\ \Phi^{\mathcal{L}_1\mathcal{L}_2}_{i,j} &= \mathcal{L}_1[\mathcal{L}_2\phi(\|\underline{x}_{\sigma_i} - \underline{x}_{\sigma_j}\|)] \ , & i = 1, \ldots, m, \ j = 1, \ldots, m, \end{aligned}$$

Note that by using $\mathcal{L}_1$ and $\mathcal{L}_2$ in defining (14), we have the relationship $\Phi^{\mathcal{L}_1} = (\Phi^{\mathcal{L}_2})^T$, making A^{H} symmetric. For the appropriate choice of ϕ, A^{H} is also guaranteed to be non-singular [31].

We can also express (12) in the following Lagrange form:

$$s(\underline{x}) = \sum_{i=1}^{n} \psi_i(\underline{x}) u(\underline{x}_i) + \sum_{j=1}^{m} \widetilde{\psi}_{\sigma_j}(\underline{x}) \mathcal{L}u(\underline{x}_{\sigma_j}) \ , \tag{15}$$

where $\psi_i(\underline{x})$ and $\widetilde{\psi}_{\sigma_j}(\underline{x})$ are of the form (12) and satisfy the cardinal conditions

$$\psi_i(\underline{x}_k) = \begin{cases} 1 & \text{if } k = i \\ 0 & \text{if } k \neq i \end{cases}, \quad k = 1, \ldots, n \ , \tag{16}$$

$$\mathcal{L}\psi_i(\underline{x}_{\sigma_k}) = 0 \ , \quad k = 1, \ldots, m \ , \tag{17}$$

and

$$\widetilde{\psi}_{\sigma_j}(\underline{x}_k) = 0 \ , \quad k = 1, \ldots, n \ , \tag{18}$$

$$\mathcal{L}\widetilde{\psi}_{\sigma_j}(\underline{x}_{\sigma_k}) = \begin{cases} 1 & \text{if } k = j \\ 0 & \text{if } k \neq j \end{cases}, \quad k = 1, \ldots, m \ . \tag{19}$$

Like the standard RBF interpolation case, there turns out to be a nice closed-form expression for $\psi_i(\underline{x})$ and $\widetilde{\psi}_{\sigma_j}(\underline{x})$. Denote $A^{\mathrm{H}}_k(\underline{x})$, $k = 1, \ldots, n+m$, as the matrix A^{H} in (14) with the k^{th} row replaced by the vector

$$B^{\mathrm{H}}(\underline{x}) = \Big[\, B(\underline{x}) \big| \mathcal{L}_2\phi(\|\underline{x} - \underline{x}_{\sigma_1}\|) \ \cdots \ \mathcal{L}_2\phi(\|\underline{x} - \underline{x}_{\sigma_m}\|) \big| \ 1 \Big] \ , \tag{20}$$

where $B(\underline{x})$ is given by (10), but without the last entry.

Theorem 2 *Let $\psi_i(\underline{x})$ and $\widetilde{\psi}_{\sigma_j}(\underline{x})$ be of the form (12) and satisfy the conditions (16),(17) and (18),(19). Then, provided the matrix A^H in (14) is non-singular,*

$$\psi_i(\underline{x}) = \frac{\det\left(A^H_i(\underline{x})\right)}{\det\left(A^H\right)} \ , \tag{21}$$

and

$$\widetilde{\psi}_{\sigma_j}(\underline{x}) = \frac{\det\left(A^H_{n+j}(\underline{x})\right)}{\det\left(A^H\right)} \ , \tag{22}$$

PROOF. The result follows by using the same inspection argument given in [20] for proving Theorem 1.

We postpone the discussion of the consequences of this formula as it relates to the shape parameter ε until Section 4.

3 RBF-FD Formulation

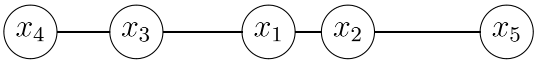
In this section we describe how the RBF-FD and HFD formulas are generated. With out loss of generality, we consider a stencil consisting of n (scattered) nodes $\underline{x}_1, \ldots, \underline{x}_n$ and are interested in approximating $\mathcal{L}u(\underline{x}_1)$, for some linear differential operator $\mathcal{L}$.

3.1 RBF-FD method

The goal is to find weights c_i such that,

$$\mathcal{L}u(\underline{x}_1) \approx \sum_{i=1}^{n} c_i u(\underline{x}_i) \ . \tag{23}$$

For example, given the 1-D scattered node stencil



,

we look for an approximation of the form

$$\mathcal{L}u(x_1) \approx \sum_{i=1}^{5} c_i u(x_i) \ ,$$

or in *computational molecule* form

$$\mathcal{L}u(x_1) \approx \boxed{c_4}\!-\!\!-\!\boxed{c_3}\!-\!\!-\!\!-\!\boxed{c_1}\!-\!\boxed{c_2}\!-\!\!-\!\!-\!\boxed{c_5}\ u \ . \tag{24}$$

Using the Lagrange form of the standard RBF interpolant (8) to approximate $u(\underline{x})$ and then applying $\mathcal{L}$ gives

$$\mathcal{L}u(\underline{x}_1) \approx \mathcal{L}s(\underline{x}_1) = \sum_{i=1}^{n} \mathcal{L}\psi_i(\underline{x}_1) u(\underline{x}_i) \ .$$

Thus, the weights in RBF-FD formula (23) are formally given by

$$c_i = \mathcal{L}\psi_i(\underline{x}_1) \ .$$

In practice, the weights are computed by solving the linear system

$$A[\mathbf{c}|\mu]^T = (\mathcal{L}B(\underline{x}_1))^T, \tag{25}$$

where A is the matrix in (7), $B(\underline{x})$ is the row vector (10), and μ is a (scalar) dummy value related to the constant β in (6). Note that the constraint (5) enforces the condition

$$\sum_{i=1}^{n} c_i = 0 \ ,$$

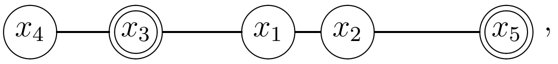
i.e. the RBF-FD stencil is exact for all constants.

3.2 RBF-HFD method

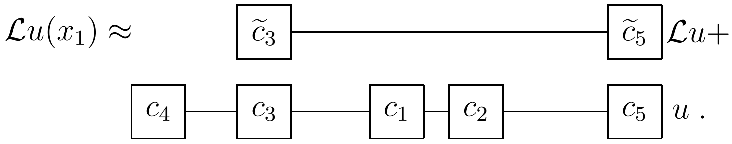
The goal now is to increase the accuracy of the approximation (23) without increasing the stencil size by using nodes where u and $\mathcal{L}u$ are given exactly. Let σ be a vector containing some combination of $m < n$ distinct numbers from the set $\{2, \ldots, n\}$, then we seek to find weights c_i and $\widetilde{c}_{\sigma_j}$ such that

$$\mathcal{L}u(\underline{x}_1) \approx \sum_{j=1}^{m} \widetilde{c}_{\sigma_j} \mathcal{L}u(\underline{x}_{\sigma_j}) + \sum_{i=1}^{n} c_i u(\underline{x}_i) \ . \tag{26}$$

For example, suppose we wish to increase the accuracy of the 1-D example (24) from the previous section by including values of $\mathcal{L}u(x_3)$ and $\mathcal{L}u(x_5)$, i.e. using the stencil



where a single circle indicates that the value of u is used at that node, and a double circle indicates that the values of u and $\mathcal{L}u$ are used. Then we let $m = 2$, $\sigma = \{3, 5\}$, and look for an approximation of the form



The Hermite interpolation example in Figure 1 can alternatively be viewed as an example of a 2-D compact scattered node stencil with $n = 10$, $m = 5$ and $\sigma = \{2, 4, 5, 6, 10\}$.

Using the Lagrange form of the Hermite RBF interpolant (15) to approximate $u(\underline{x})$ and then applying $\mathcal{L}$ gives

$$\mathcal{L}u(\underline{x}_1) \approx \mathcal{L}s(\underline{x}_1) = \sum_{i=1}^{n} \mathcal{L}\psi_i(\underline{x}_1) u(\underline{x}_i) + \sum_{j=1}^{m} \mathcal{L}\widetilde{\psi}_{\sigma_j}(\underline{x}_1) \mathcal{L}u(\underline{x}_{\sigma_j}) \ .$$

Thus, the weights in the RBF-HFD formula (26) are formally given by

$$c_i = \mathcal{L}\psi_i(\underline{x}_1) \quad \text{and} \quad \widetilde{c}_{\sigma_j} = \mathcal{L}\widetilde{\psi}_{\sigma_j}(\underline{x}_1) \ .$$

In practice, the weights are computed by solving the linear system

$$A^{\mathrm{H}}[\mathbf{c}|\widetilde{\mathbf{c}}|\mu]^T = (\mathcal{L}B^{\mathrm{H}}(\underline{x}_1))^T, \tag{27}$$

where A^{H} is the matrix in (14), $B^{\mathrm{H}}(\underline{x})$ is the row vector (20), and μ is a dummy (scalar) value related to the constant β in (12). Like the standard RBF-FD formula, the constraint (5) enforces that (26) is exact for all constants.

Note that the above formulation for computing compact FD formulas differs from the standard polynomial-based formulation. For polynomials, the point at which we are constructing the approximation about does not matter; the only thing that does matter is that the formulas are exact for as high a degree polynomial as possible [13].

4 Observations on the effect of the shape parameter ε

In this section, we make some observations with regard to ε on the RBF-FD and HFD formulas. We are primarily interested in studying the resulting formulas as we let $\varepsilon \to 0$.

4.1 Some theoretical results on the $\varepsilon \to 0$ limit

For nodes in 1-D, it was first shown in [19] that, under some mild restrictions on the infinitely smooth radial functions $\phi(r)$ (e.g. MQ, see Table 1), the standard RBF interpolant converges to the Lagrange interpolating polynomial as $\varepsilon \to 0$. This result means that all "classical"polynomial-based FD stencils are reproduced by the standard RBF-FD stencils (with the appropriate choice of $\phi(r)$) in the limit of $\varepsilon \to 0$. A few examples of this result are given in the next section.

For nodes in two and higher dimensions, the situation is complicated by the fact that multivariate polynomial interpolation is not well-posed [16]. While the standard RBF interpolant typically converges to a low degree multivariate polynomial as $\varepsilon \to 0$, there are circumstances where the interpolant may instead diverge; see [20–22,33,34] for more details. The authors are unaware of any similar study of limiting ($\varepsilon \to 0$) Hermite RBF interpolants. However, we expect similar results to those of the standard RBF interpolant. For example, the following theorem indicates that polynomial type results may be expected in the $\varepsilon \to 0$ limit.

Theorem 3 *Consider the Hermite RBF interpolant (12). If $\lim_{\varepsilon\to 0} s(\underline{x})$ exists, it will be a (multivariate) finite degree polynomial in $\underline{x}$.*

PROOF. Consider the Lagrange form of $s(\underline{x})$ (15). Note that we can expand $\phi(\|\underline{x}-\underline{x}_i\|)$ and $\mathcal{L}_2\phi(\|\underline{x}-\underline{x}_i\|)$ in powers of ε^2 so that the coefficients in front of these powers will be some polynomial in $\underline{x}$. The same will, therefore, hold for the determinant in the numerator of (21) and (22). Thus, the ratios in (21) and (22) will be of the respective forms

$$\psi_i(\underline{x}) = \frac{\varepsilon^{2p_i}\{\text{poly in } \underline{x}\} + \varepsilon^{2p_i+2}\{\text{poly in } \underline{x}\} + \cdots}{\varepsilon^{2q_i}\{\text{constant}\} + \varepsilon^{2q_i+2}\{\text{constant}\} + \cdots}$$

and

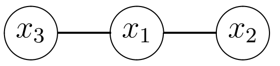
$$\widetilde{\psi}_{\sigma_j}(\underline{x}) = \frac{\varepsilon^{2\widetilde{p}_j}\{\text{poly in } \underline{x}\} + \varepsilon^{2\widetilde{p}_j+2}\{\text{poly in } \underline{x}\} + \cdots}{\varepsilon^{2\widetilde{q}_j}\{\text{constant}\} + \varepsilon^{2\widetilde{q}_j+2}\{\text{constant}\} + \cdots} ,$$

where p_i, q_i, $\widetilde{p}_j$, and $\widetilde{q}_j$ are positive integers. Since $\psi_i(\underline{x}_i) = 1$ and $\mathcal{L}\widetilde{\psi}_{\sigma_j}(\underline{x}_{\sigma_j}) = 1$, it is impossible to have $p_i > q_i$ and $\widetilde{p}_j > \widetilde{q}_j$. If $p_i < q_i$ or $\widetilde{p}_j < \widetilde{q}_j$, the limit fails to exist. Thus, when $p_i = q_i$ and $\widetilde{p}_j = \widetilde{q}_j$, $s(\underline{x})$ is some polynomial in $\underline{x}$.

Like the polynomial result for the standard RBF interpolant, we can use Theorem 3 to conclude that when the underlying Hermite interpolant exists, the RBF-HFD formulas will be exact for some polynomials. In the next two sections, we give a few examples that demonstrate this result.

4.2 A few examples with closed-form solutions for the $\varepsilon \to 0$ limit

We first present two examples based on an $n = 3$ node, equispaced 1-D stencil with the nodes numbered



,

where $\{x_1, x_2, x_3\} = \{0, h, -h\}$. Since the nodes are in 1-D, we drop the underline on the x-values. In all cases, we use a general radial function with an expansion

$$\phi(r) = a_0 + a_1(\varepsilon r)^2 + a_2(\varepsilon r)^4 + a_3(\varepsilon r)^6 \cdots ,$$

and consider the resulting RBF-FD formulas (23) and RBF-HFD formulas (26) as $\varepsilon \to 0$. We make use of the cardinal interpolants from Theorems 1 and 2.

Example 1 *Approximate $u'(x_1)$, i.e let $\mathcal{L} = \dfrac{d}{dx}$*

<u>RBF-FD formula</u>: The determinant in the denominators of the cardinal interpolants (11) is given by

$$\det(A) = 2bh^6F(a)\varepsilon^6 + O(\varepsilon^8) ,$$

while the determinants in the numerators are given by

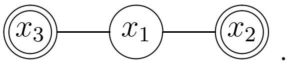
$$\begin{aligned}
\det(A_1(x)) &= -2bh^4(x-h)(x+h)F(a)\varepsilon^6 + O(\varepsilon^8),\\
\det(A_2(x)) &= bh^4x(x+h)F(a)\varepsilon^6 + O(\varepsilon^8),\\
\det(A_3(x)) &= bh^4x(x-h)F(a)\varepsilon^6 + O(\varepsilon^8)\ ,
\end{aligned}$$

where $b = 24$ and

$$F(a) = a_1a_2\ . \tag{28}$$

As expected from [19], provided $a_1a_2 \neq 0$, $\psi_i(x)$, $i = 1, 2, 3$, converge to the standard cardinal interpolating polynomials in the limit as $\varepsilon \to 0$. Thus, the classical, centered, second order FD formula for $u'(x_1)$ is recovered in the $\varepsilon \to 0$ limit.

<u>RBF-HFD formula</u>: Using the notation from Section 2.2 and 3.2, we let $m = 2$ and $\sigma = \{2, 3\}$, i.e.



The determinant in the denominators of the cardinal interpolants (21) and (22) is given by

$$\det(A^{\mathrm{H}}) = 4bh^{16}F(a)\varepsilon^{20} + O(\varepsilon^{22})\ , \tag{29}$$

while the determinants in the numerators are given by

$$\begin{aligned}
\det(A_1^{\mathrm{H}}(x)) &= -4bh^{12}(x^2-h^2)^2F(a)\varepsilon^{20} + O(\varepsilon^{22})\ ,\\
\det(A_2^{\mathrm{H}}(x)) &= bh^{12}(2x-3h)x(x+h)^2F(a)\varepsilon^{20} + O(\varepsilon^{22})\ ,\\
\det(A_3^{\mathrm{H}}(x)) &= bh^{12}(2x+3h)x(x-h)^2F(a)\varepsilon^{20} + O(\varepsilon^{22})\ ,\\
\det(A_4^{\mathrm{H}}(x)) &= bh^{13}(x-h)x(x+h)^2F(a)\varepsilon^{20} + O(\varepsilon^{22})\ ,\\
\det(A_5^{\mathrm{H}}(x)) &= bh^{13}(x-h)^2x(x+h)F(a)\varepsilon^{20} + O(\varepsilon^{22})\ ,
\end{aligned}$$

where $b = 7680$ and

$$F(a) = (2a_2^2 - 5a_1a_3)(15a_3^2 - 28a_2a_4)\ . \tag{30}$$

Provided $(2a_2^2 - 5a_1a_3) \neq 0$ and $(15a_3^2 - 28a_2a_4) \neq 0$, $\psi_i(x)$, $i = 1, 2, 3$, and $\widetilde{\psi}_j(x)$, $j = 2, 3$, converge to the standard Hermite cardinal interpolating polynomials as $\varepsilon \to 0$. The weights in the limiting RBF-HFD formula for $u'(x_1)$ are thus the same as the classical, centered, fourth order compact FD scheme [23, p. 538]

$$u'(x_1) \approx \boxed{-\tfrac{1}{4}}\rule[0.5ex]{8em}{0.4pt}\boxed{-\tfrac{1}{4}}\ u' + \boxed{-\tfrac{3}{4}}\rule[0.5ex]{3em}{0.4pt}\boxed{\ 0\ }\rule[0.5ex]{3em}{0.4pt}\boxed{\tfrac{3}{4}}\ \frac{u}{h}\ .\square$$

Example 2 *Approximate $u''(x_1)$, i.e let $\mathcal{L} = \dfrac{d^2}{dx^2}$*

RBF-FD formula: The cardinal interpolants for this example are the same as the ones from Example 1. Thus again, the classical, centered, second order accurate FD scheme for $u''(x_1)$ is recovered in the $\varepsilon \to 0$ limit.

RBF-HFD formula: We again let $m = 2$ and $\sigma = \{2, 3\}$. The determinant in denominators of the cardinal interpolants (21) and (22) is given by

$$\det(A^{\mathrm{H}}) = 60bh^{12}F(a)\varepsilon^{20} + O(\varepsilon^{22}) \ , \tag{31}$$

while the determinants in the numerators are given by

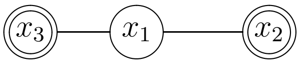
$$\begin{aligned}
\det(A^{\mathrm{H}}_1(x)) &= -12bh^8(x-h)(x+h)(x^2-5h^2)F(a)\varepsilon^{20} + O(\varepsilon^{22}) \ , \\
\det(A^{\mathrm{H}}_2(x)) &= 6bh^8x(x+h)(x^2-hx-5h^2)F(a)\varepsilon^{20} + O(\varepsilon^{22}) \ , \\
\det(A^{\mathrm{H}}_3(x)) &= 6bh^8(x-h)x(x^2+hx-5h^2)F(a)\varepsilon^{20} + O(\varepsilon^{22}) \ , \\
\det(A^{\mathrm{H}}_4(x)) &= bh^{10}(x-h)x(x+h)(3x+5h)F(a)\varepsilon^{20} + O(\varepsilon^{22}) \ , \\
\det(A^{\mathrm{H}}_5(x)) &= bh^{10}(3x-5h)(x-h)x(x+h)F(a)\varepsilon^{20} + O(\varepsilon^{22}) \ ,
\end{aligned}$$

where $b = 115200$ and $F(a)$ is given by (30). Provided again $F(a) \not\equiv 0$, we recover the standard cardinal Hermite interpolating polynomials in the limit of $\varepsilon \to 0$. The weights in the limiting RBF-HFD scheme for $u''(x_1)$ are

$$u''(x_1) \approx \boxed{-\tfrac{1}{10}}\rule[0.5ex]{6em}{0.4pt}\boxed{-\tfrac{1}{10}}\ u'' + \boxed{\tfrac{6}{5}}\rule[0.5ex]{2em}{0.4pt}\boxed{-\tfrac{12}{5}}\rule[0.5ex]{2em}{0.4pt}\boxed{\tfrac{6}{5}}\ \frac{u}{h^2} \ .$$

This is identical to the classical, centered, fourth order compact FD scheme of Collatz [23, p. 538]. □

The last example illustrates how symmetries in the stencil can make the order of accuracy relatively high compared to the number of nodes. For the three node stencil, the weights in the classical FD formula come from the second degree interpolating polynomial. This may lead us to conclude that, after differentiating twice, the order of accuracy of the formula is only one (i.e. it is exact for all polynomials of degree ≤ 2). However, due to the symmetry around x_1, the formula is in fact second order accurate (i.e. it is exact for all polynomials of degree ≤ 3). Similarly, for the classical compact three node stencil, we gain one order of accuracy from what is expected (fourth instead of third). In the next example, we study the limiting ($\varepsilon \to 0$) RBF-FD and HFD formulas for the "scattered" node stencil



,

where $\{x_1, x_2, x_3\} = \{0, 3h/2, -h\}$, and show how the accuracy is reduced.

Example 3 *Approximate $u''(x_1)$ using the above stencil.*

RBF-FD formula: Interpolating to only the function values in the above stencil yields the following results for the denominators and numerators of (11):

$$\begin{aligned}
\det(A) &= 15bh^6F(a)\varepsilon^6 + O(\varepsilon^8),\\
\det(A_1(x)) &= 5bh^4(2x-3h)(x+h)F(a)\varepsilon^6 + O(\varepsilon^8),\\
\det(A_2(x)) &= 4bh^4x(x+h)F(a)\varepsilon^6 + O(\varepsilon^8),\\
\det(A_3(x)) &= 3bh^4x(2x-3h)F(a)\varepsilon^6 + O(\varepsilon^8)\ ,
\end{aligned}$$

where $b = 45/4$ and $F(a)$ is given by (28). Again, provided $F(a) \not\equiv 0$, the cardinal RBF interpolants converge to the standard Lagrange interpolating polynomials as $\varepsilon \to 0$. The limiting RBF-FD formula is given by

$$u''(x_1) \approx \boxed{\tfrac{4}{5}}\rule[0.5ex]{1cm}{0.4pt}\boxed{-\tfrac{4}{3}}\rule[0.5ex]{1.5cm}{0.4pt}\boxed{\tfrac{8}{15}}\ \frac{u}{h^2}\ .$$

This FD scheme is exact for all polynomials of degree ≤ 2, leading us to conclude that it is only first order accurate.

RBF-HFD formula: We again let $m = 2$ and $\sigma = \{2, 3\}$. The determinant in the denominators of the cardinal interpolants (21) and (22) is given by

$$\det(A^{\mathrm{H}}) = -32768bh^{12}F(a)\varepsilon^{20} + O(\varepsilon^{22})\ , \tag{32}$$

while the determinants in the numerators are given by

$$\begin{aligned}
\det(A_1^{\mathrm{H}}(x)) &= -4650bh^8(2x-3h)(x+h)(4x^2-2hx-31h^2)F(a)\varepsilon^{20} + O(\varepsilon^{22})\ ,\\
\det(A_2^{\mathrm{H}}(x)) &= 14880bh^8x(x+h)(x^2-2hx+7h^2)F(a)\varepsilon^{20} + O(\varepsilon^{22})\ ,\\
\det(A_3^{\mathrm{H}}(x)) &= 2790bh^8(2x-3h)x(4x^2+2hx-33h^2)F(a)\varepsilon^{20} + O(\varepsilon^{22})\ ,\\
\det(A_4^{\mathrm{H}}(x)) &= -930bh^{10}(2x-3h)x(x+h)(7x+12h)F(a)\varepsilon^{20} + O(\varepsilon^{22})\ ,\\
\det(A_5^{\mathrm{H}}(x)) &= -465bh^{10}(2x-3h)x(x+h)(16x-39h)F(a)\varepsilon^{20} + O(\varepsilon^{22})\ ,
\end{aligned}$$

where $b = 3375/16$ and $F(a)$ is given by (30). Provided again $F(a) \not\equiv 0$, we recover the standard cardinal Hermite interpolating polynomials in the limit of $\varepsilon \to 0$. The weights in the limiting RBF-HFD scheme for $u''(x_1)$ are

$$u''(x_1) \approx \boxed{\tfrac{-3}{155}}\rule[0.5ex]{3cm}{0.4pt}\boxed{\tfrac{-22}{155}}\ u'' + \boxed{\tfrac{144}{155}}\rule[0.5ex]{0.8cm}{0.4pt}\boxed{\tfrac{-48}{31}}\rule[0.5ex]{1.3cm}{0.4pt}\boxed{\tfrac{96}{155}}\ \frac{u}{h^2}\ .$$

This FD formula is exact for all polynomials of degree ≤ 4, leading us to conclude that it is third order accurate. If an application requires that a three node stencil is used, then to increase the accuracy it is imperative that we use a compact stencil. □

While all the examples above involve centered FD formulas, we note that a similar recovery of the classical, one-sided formulas by the RBF-FD and

HFD formulas occurs. This is because of the convergence of the regular and Hermite RBF interpolants to the standard Lagrange and Hermite polynomial interpolants, respectively, in the $\varepsilon \to 0$ limit.

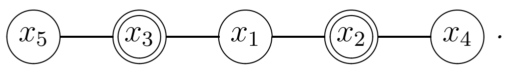
From the examples above, we see that for each of the RBF-FD formulations we obtain a set of requirements that the Taylor coefficients of the radial function must satisfy in order for the limiting interpolant to exist. We note that each of the infinitely smooth functions in Table 1 satisfies the requirements from each example. For larger stencil sizes (i.e. more nodes), the requirements that the Taylor coefficients must satisfy become more and more intricate. For the standard method in 1-D, the full set of requirements for any number of nodes was proven in [19] and extended to >1-D in [21] (in both cases, however, a slightly different formulation of the interpolant is considered). The full set of requirements for convergence in the $\varepsilon \to 0$ limit of the Hermite RBF interpolation method are still unknown. However, experiments suggest they are similar to the standard case.

The results for the RBF-HFD formulas in the above examples required quite extensive symbolic manipulation with *Mathematica*. Extending these results to more nodes is, at the present time, intractable. We can, however, numerically study the $\varepsilon \to 0$ formulas for larger stencils with the Contour-Padé algorithm [33]. Although originally developed for the RBF interpolation problem, this algorithm can be easily extended for computing the RBF-FD and HFD weights.

4.3 Numerical results using the Contour-Padé algorithm

In the first couple of examples, we further explore the connection between limiting ($\varepsilon \to 0$) RBF-HFD schemes and the classical compact FD schemes. We follow this with some results based on scattered node stencils. The results of each of the examples in this section are based on the MQ and GA radial functions.

Example 4 *Approximate $u'(x_1)$ and $u''(x_1)$ using the five-node equispaced stencil:*



The parameters for this example are $n = 5$, $m = 2$, and $\sigma = \{2, 3\}$. Computations with the Contour-Padé algorithm suggest that the RBF-HFD formulas in the $\varepsilon \to 0$ limit converges to the standard, sixth order accurate compact

FD formulas [23, p. 538]:

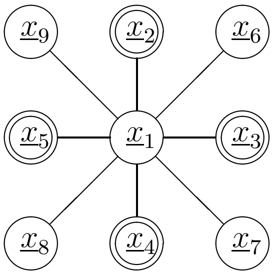
$$u'(x_1) \approx \boxed{\tfrac{-1}{3}}\text{———————}\boxed{\tfrac{-1}{3}}\; u' + \boxed{\tfrac{-1}{36}}\text{—}\boxed{\tfrac{-7}{9}}\text{—}\boxed{0}\text{—}\boxed{\tfrac{7}{9}}\text{—}\boxed{\tfrac{1}{36}}\;\frac{u}{h}$$

and

$$u''(x_1) \approx \boxed{\tfrac{2}{-11}}\text{———————}\boxed{\tfrac{2}{-11}}\; u'' + \boxed{\tfrac{3}{44}}\text{—}\boxed{\tfrac{48}{44}}\text{—}\boxed{\tfrac{102}{-44}}\text{—}\boxed{\tfrac{48}{44}}\text{—}\boxed{\tfrac{3}{44}}\;\frac{u}{h^2}.$$

□

Example 5 *Approximate $\Delta u(\underline{x}_1)$ using the nine-node equispaced stencil:*



.

The parameters for this example are $n = 9$, $m = 4$, and $\sigma = \{2, 3, 4, 5\}$. Computations with the Contour-Padé algorithm suggest that the RBF-HFD formula in the $\varepsilon \to 0$ limit converges to the classical, fourth order accurate compact FD formula [23, p. 542]:

$$\Delta u(x_1) \approx \begin{array}{ccc} & \boxed{-\frac{1}{8}} & \\ & | & \\ \boxed{-\frac{1}{8}}\text{——} & \text{—}\!\!\!+\!\!\!\text{—} & \text{——}\boxed{-\frac{1}{8}} \\ & | & \\ & \boxed{-\frac{1}{8}} & \end{array} \Delta u + \begin{array}{ccccc} \boxed{\frac{1}{4}} & & \boxed{1} & & \boxed{\frac{1}{4}} \\ & \diagdown & | & \diagup & \\ \boxed{1} & \text{—} & \boxed{-5} & \text{—} & \boxed{1} \\ & \diagup & | & \diagdown & \\ \boxed{\frac{1}{4}} & & \boxed{1} & & \boxed{\frac{1}{4}} \end{array} \;\frac{u}{h^2}.\ \square$$

The results for $u''(x_1)$ and $\Delta u(\underline{x}_1)$ in the previous two examples illustrate two important observations. First, not including the derivative information results in FD formulas that are only fourth and second order accurate, respectively. Second, unlike the compact schemes, these formulas would also not be diagonally dominant, as we would want from a discrete approximation of the Laplacian. In Section 5, we give evidence that for scattered nodes it is also more likely to find a diagonally dominant RBF-FD formula when we make it compact.

							m					
		0	1	2	3	4	5	6	7	8	9	10
	1	0	0	1	1	1	2	2	2	2	3	3
	2	0	1	1	1	2	2	2	2	3	3	3
	3	1	1	1	2	2	2	2	3	3	3	3
	4	1	1	2	2	2	2	3	3	3	3	3
	5	1	2	2	2	2	3	3	3	3	3	4
n	6	2	2	2	2	3	3	3	3	3	4	4
	7	2	2	2	3	3	3	3	3	4	4	4
	8	2	2	3	3	3	3	3	4	4	4	4
	9	2	3	3	3	3	3	4	4	4	4	4
	10	3	3	3	3	3	4	4	4	4	4	4
	11	3	3	3	3	4	4	4	4	4	4	5

Table 2
Numerical results for the maximum degree 2-D polynomial the limiting ($\varepsilon \to 0$) RBF-HFD formulas for Δu are exact for using a scattered node stencil. Here n and m are the the number of values of u and Δu used in the formulas, respectively.

We now focus on the case of a scattered node stencil (cf. Figure 1). As an experiment, we considered 11 scattered nodes in 2-D and computed the limiting ($\varepsilon \to 0$) RBF-HFD formulas for approximating Δu using n values of u and m values of Δu such that $1 \leq n \leq 11$ and $0 \leq m \leq 10$. We then tested the resulting formulas on all polynomials in the standard basis for polynomials of degree ≤ 6. Table 2 displays the maximum degree polynomial the formulas were observed to become exact for. If we let Q equal the maximum degree polynomial, then the results from the table seem to indicate that when $\frac{1}{2}(Q+2)(Q+1) \leq n+m < \frac{1}{2}(Q+3)(Q+2)$, the limiting RBF-HFD formula is exact for all polynomials of degree Q. These are the same numbers we would expect if using regular (2-D) polynomial interpolation (with the standard basis) to generate the FD formulas (assuming these polynomial interpolants exist).

5 Application and implementation: elliptic PDEs

In all the numerical experiments that follow, we use the MQ radial function because of its popularity in applications. All of the model problems involve the Laplace linear differential operator, i.e. $\mathcal{L} = \Delta$. We note that the d-dimensional Laplacian and biharmonic operators applied to any radially symmetric function $\phi(r)$ are given by

$$\begin{aligned} \Delta\phi &= (d-1)\frac{1}{r}\frac{d\phi}{dr} + \frac{d^2\phi}{dr^2}\,, \\ \Delta^2\phi &= \frac{d^2-4d+3}{r^2}\left[\frac{d^2\phi}{dr^2} - \frac{1}{r}\frac{d\phi}{dr}\right] + 2(d-1)\frac{1}{r}\frac{d^3\phi}{dr^3} + \frac{d^4\phi}{dr^4}\,. \end{aligned}$$

Also, $\mathcal{L}_1\phi(\|\underline{x}-\underline{y}\|) = \mathcal{L}_2\phi(\|\underline{x}-\underline{y}\|)$ for the Laplacian.

5.1 Poisson equation on the unit square: uniform discretization

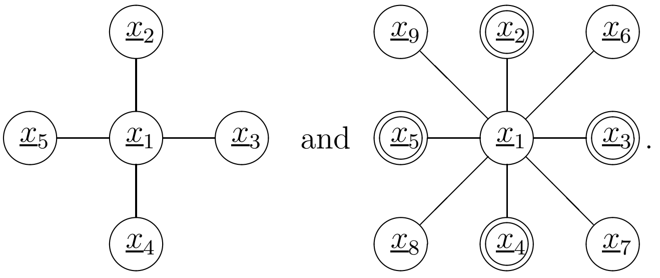
In this first example, we illustrate the convergence rate of the RBF-FD method for the standard and compact formulas as the grid is refined. To do this, we consider the model problem

$$\Delta u = f \;\text{ in } \Omega = \{(x,y)\,|\, 0 \le x, y \le 1\}\,, \; u = g \;\text{ on } \partial\Omega\,, \tag{33}$$

where f and g are computed from the known solution

$$u(\underline{x}) = u(x,y) = e^{-(x-1/4)^2-(y-1/2)^2}\cos(2\pi y)\sin(\pi x)\ .$$

The solution is approximated on an equispaced grid of spacing h using both an RBF-FD formula with $n = 5$ nodes and an RBF-HFD formula with $n = 9$ and $m = 4$ nodes, i.e. using the respective stencils



(34)

In this case, it is only necessary to compute the standard and compact formulas once for these stencils and apply them all over the discretization. From the results of Section 4.3, we expect the RBF-FD and HFD formulas for these stencils to converge in the $\varepsilon \to 0$ limit to the classical, second order accurate and compact, fourth order accurate FD formulas (FD2 and compact FD4), respectively.

Figure 2 displays the max norm error of the RBF-FD and HFD solutions for different values of ε and h. Moving from top to bottom in each of the plots, the error curves correspond to the solutions with grid spacing $h = 0.2, 0.1, 0.05, 0.02, 0.01, 0.005, 0.002$. We can see from the figure that for each h, the solution with the minimum error occurs at a non-zero value of ε. Furthermore, for each ε the error appears to be decreasing at a fairly constant rate as h decreases. In Table 3 and 4 we display the max norm error for different values of ε and h and the observed rate of convergence for the RBF-FD and HFD formulas. We can see from the table that even for non-zero values of ε, both of the formulas demonstrate the same convergence rate as the standard

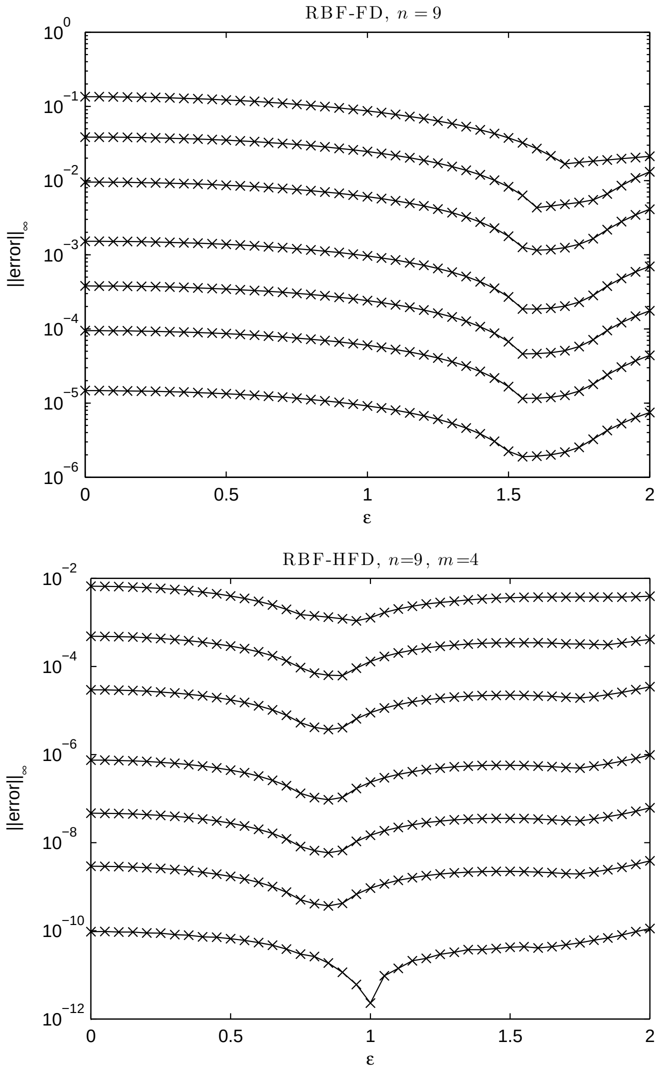


Fig. 2. The error as a function of ε and h for the solution of (33) using the RBF-FD and HFD formulas based on the stencils in (34). Each error curve corresponds to a different value of h with varying ε.

FD2 and compact FD4 formulas (i.e. the results for $\varepsilon = 0$). However, for larger values of ε, the results indicate that h is required to be much smaller before the standard rates of convergence are observed.

We conclude this example by noting that for the full range of ε considered here, the weights in the RBF-FD and HFD formulas satisfied the diagonal dominance condition

$$c_1 < 0,\ c_j > 0,\ j = 2, \ldots, n, \text{ and } \sum_{j=1}^{n} c_j = 0\ . \qquad (35)$$

Thus, the resulting symmetric, banded linear system for determining the approximate solution (for each ε and h) was guaranteed to be negative definite.

To solve these systems we used a standard banded Gaussian elimination solver. However, any iterative method applicable to the the classical FD2 and compact FD4 methods (e.g. multigrid) could also have been used.

				h–grid spacing			
	$2.0 \cdot 10^{-1}$	$1.0 \cdot 10^{-1}$	$5.0 \cdot 10^{-2}$	$2.0 \cdot 10^{-2}$	$1.0 \cdot 10^{-2}$	$5.0 \cdot 10^{-3}$	$2.0 \cdot 10^{-3}$
$\varepsilon = 0.0$	$1.3 \cdot 10^{-1}$	$3.8 \cdot 10^{-2}$	$9.5 \cdot 10^{-3}$	$1.5 \cdot 10^{-3}$	$3.8 \cdot 10^{-4}$	$9.5 \cdot 10^{-5}$	$1.5 \cdot 10^{-5}$
rate	—	1.8	2.0	2.0	2.0	2.0	2.0
$\varepsilon = 0.25$	$1.3 \cdot 10^{-1}$	$3.8 \cdot 10^{-2}$	$9.3 \cdot 10^{-3}$	$1.5 \cdot 10^{-3}$	$3.7 \cdot 10^{-4}$	$9.2 \cdot 10^{-5}$	$1.4 \cdot 10^{-5}$
rate	—	1.8	2.0	2.0	2.0	2.0	2.0
$\varepsilon = 1.0$	$8.7 \cdot 10^{-2}$	$2.5 \cdot 10^{-2}$	$6.0 \cdot 10^{-3}$	$9.6 \cdot 10^{-4}$	$2.4 \cdot 10^{-4}$	$6.0 \cdot 10^{-5}$	$9.2 \cdot 10^{-6}$
rate	—	1.8	2.0	2.0	2.0	2.0	2.1
$\varepsilon = 1.6$	$2.7 \cdot 10^{-2}$	$4.3 \cdot 10^{-3}$	$1.1 \cdot 10^{-3}$	$1.8 \cdot 10^{-4}$	$4.6 \cdot 10^{-5}$	$1.2 \cdot 10^{-5}$	$1.9 \cdot 10^{-6}$
rate	—	2.6	1.9	2.0	2.0	2.0	2.0
$\varepsilon = 2.0$	$2.1 \cdot 10^{-2}$	$1.3 \cdot 10^{-2}$	$4.1 \cdot 10^{-3}$	$7.0 \cdot 10^{-4}$	$1.8 \cdot 10^{-4}$	$4.4 \cdot 10^{-5}$	$7.5 \cdot 10^{-6}$
rate	—	0.7	1.7	1.9	2.0	2.0	1.9

Table 3
Max norm error and observed convergence rate for the approximate solution of (33) using the $n = 5$ node RBF-FD formula.

				h–grid spacing			
	$2.0 \cdot 10^{-1}$	$1.0 \cdot 10^{-1}$	$5.0 \cdot 10^{-2}$	$2.0 \cdot 10^{-2}$	$1.0 \cdot 10^{-2}$	$5.0 \cdot 10^{-3}$	$2.0 \cdot 10^{-3}$
$\varepsilon = 0.0$	$6.6 \cdot 10^{-3}$	$4.8 \cdot 10^{-4}$	$2.9 \cdot 10^{-5}$	$7.5 \cdot 10^{-7}$	$4.7 \cdot 10^{-8}$	$2.9 \cdot 10^{-9}$	$9.5 \cdot 10^{-11}$
rate	—	3.8	4.0	4.0	4.0	4.0	3.7
$\varepsilon = 0.25$	$5.9 \cdot 10^{-3}$	$4.3 \cdot 10^{-4}$	$2.6 \cdot 10^{-5}$	$6.7 \cdot 10^{-7}$	$4.2 \cdot 10^{-8}$	$2.6 \cdot 10^{-9}$	$8.9 \cdot 10^{-11}$
rate	—	3.8	4.0	4.0	4.0	4.0	3.7
$\varepsilon = 0.85$	$1.3 \cdot 10^{-3}$	$6.3 \cdot 10^{-5}$	$3.7 \cdot 10^{-6}$	$9.4 \cdot 10^{-8}$	$5.9 \cdot 10^{-9}$	$3.7 \cdot 10^{-10}$	$1.8 \cdot 10^{-11}$
rate	—	4.4	4.1	4.0	4.0	4.0	3.3
$\varepsilon = 1.6$	$3.7 \cdot 10^{-3}$	$3.4 \cdot 10^{-4}$	$2.1 \cdot 10^{-5}$	$5.5 \cdot 10^{-7}$	$3.4 \cdot 10^{-8}$	$2.1 \cdot 10^{-9}$	$4.1 \cdot 10^{-11}$
rate	—	3.4	4.0	4.0	4.0	4.0	4.3
$\varepsilon = 2.0$	$3.9 \cdot 10^{-3}$	$4.1 \cdot 10^{-4}$	$3.5 \cdot 10^{-5}$	$9.8 \cdot 10^{-7}$	$6.2 \cdot 10^{-8}$	$3.9 \cdot 10^{-9}$	$1.1 \cdot 10^{-10}$
rate	—	3.3	3.6	3.9	4.0	4.0	3.9

Table 4
Max norm error and observed convergence rate for the approximate solution of (33) using the $n = 9$, $m = 4$ node RBF-HFD formula.

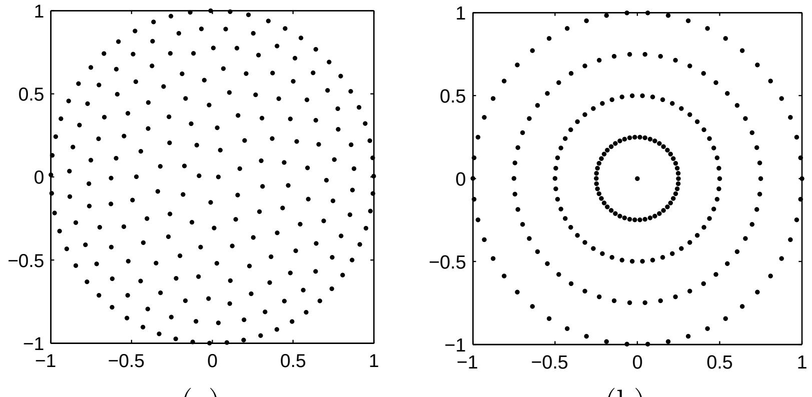
5.2 Poisson equation on the unit disk: unstructured discretization

We consider the model problem

$$\Delta u = f \ \text{ in } \Omega = \Big\{(x, y) \mid x^2 + y^2 \leq 1\Big\}, \ u = g \ \text{ on } \partial\Omega \ , \tag{36}$$

where f and g are computed from the known solution

$$u(\underline{x}) = u(x, y) = \frac{25}{25 + (x - 0.2)^2 + 2y^2} \ .$$



(a)

(b)

Fig. 3. (a) 200 point unstructured discretization and (b) 201 point structured discretization of the unit disk for the problem in Section 5.2.

The domain is discretized using the $N = 200$ points shown in Figure 3 (a). The purpose of this example is to compare the RBF-FD and HFD methods on an unstructured grid as ε and the number of nodes in the stencil are varied.

We begin by describing the *greedy* algorithm used for selecting the nodes for each of the stencils from the larger unstructured discretization. The essential factors used for determining when an acceptable stencil is found are that the weights in the resulting RBF-FD or RBF-HFD formula satisfy the diagonal dominance condition (35) and

$$\frac{1}{h^2} \le |c_1| \le \frac{2(n-1)}{h^2}, \quad h = \max_{i=1,\ldots,n}\left[\min_{\substack{j=1,\ldots,n\\ i\neq j}} \|\underline{x}_i - \underline{x}_j\|\right], \tag{37}$$

where $\underline{x}_i$, $i = 1, 2, \ldots, n$ are the nodes in the stencil. The first condition guarantees the linear system for discretizing (36) is negative definite, while the second condition is meant to balance the influence of the point the approximation is about. This latter condition would be satisfied by either the standard FD or compact FD formulas on an equispaced grid.

The stencil selection algorithm is as follows: Let N be the number of points in the unstructured discretization of the domain Ω and let $N_{\mathcal{I}}$ be the number of points at which to compute all the n-node RBF-FD or HFD formulas. We denote each point as $\underline{x}_{i,1}$, $i = 1, \ldots, N$, and the nodes that make up the stencil at each point as $\underline{x}_{i,j}$, $j = 1, \ldots, n$ (note that $\underline{x}_{i,1}$ is always part of the stencil). Using the notation from Section 3, we let $0 \le m < n$ be the number of nodes that contain both function and derivative information. The steps in the algorithm are as follows:

(1) For each $\underline{x}_{i,1}$, $i = 1, \ldots, N_{\mathcal{I}}$, determine its η nearest neighbors, where η is on the order of n.
(2) Compute all possible combinations of choosing $n - 1$ nodes from these η

nearest neighbors and sort them according to their average distance from node $\underline{x}_{i,1}$. These nodes will be where the function values of the stencil are given.

(3) For each set of nodes from the previous step, compute all possible combinations of choosing m nodes and again sort them by their average distance from $\underline{x}_{i,1}$. These nodes will be where the derivative values of the function are also given.

(4) Looping first over the sorted set of nodes from step 2 followed by the sorted set from step 3, compute the RBF-FD or RBF-HFD formula. When the direct computation becomes unstable for small ε, use the Contour-Padé algorithm [33]. Terminate the nested loops when the weights satisfy (35) and (37).

We make a few comments regarding this algorithm:

- If an acceptable stencil can not be found, one could choose to increase or decrease n or m. Based on the numerical results from Section 4.3, we recommend trying to keep $n + m$ constant for all the stencils. Additionally, in many cases, the points in the discretization can be chosen freely. Thus, it may also be possible to move or add some nodes until an acceptable stencil is found.
- In cases where the number of points N in the discretization is large, an efficient search algorithm will be required for determining the points close to each $\underline{x}_{i,1}$; the *binning* algorithm described by Liu [35, Chap. 15] may be appropriate for this.
- Determining the stencil from the η closest points may not result in the best stencil choice, especially if there are large discrepancies in the discretization points. A possible improvement may be to compute a local Delaunay triangulation about the points surrounding $\underline{x}_{i,1}$ to determine the points natural neighbors. This idea is also mentioned in [10].
- If the above algorithm was applied to the equispaced example from the previous section, then it would immediately select the standard stencils (34) as acceptable. In fact, this default behavior is what guided the algorithms development.

Using the above algorithm, we computed RBF-FD formulas with $n = 9$ nodes for all the interior points of Figure 3 (a) and for several different values of ε. Similarly we computed two sets of RBF-HFD formulas; one with $n = 9$, $m = 5$ and the other with $n = 10$, $m = 9$. In all cases, acceptable stencils were found after very few iterations of the stencil selection algorithm. Figure 4 displays the max norm error in the solution to (36) with these formulas as a function of ε. A point on any of the curves corresponds to the error in the solution using the same value of ε in all the stencils. The figure clearly shows that the accuracy is vastly improved by using the compact stencils. Furthermore, as we expect, the accuracy can further be improved by increasing the size of the

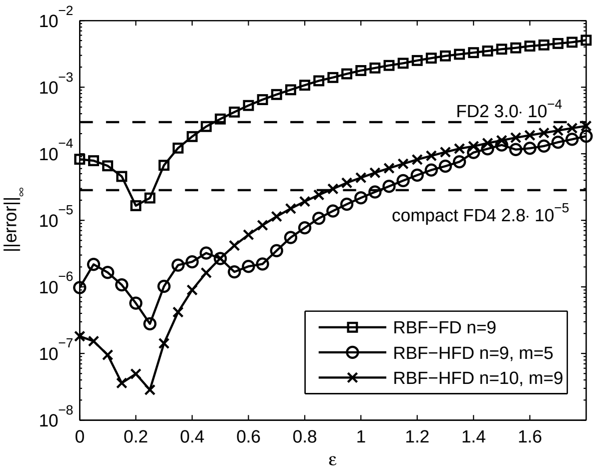
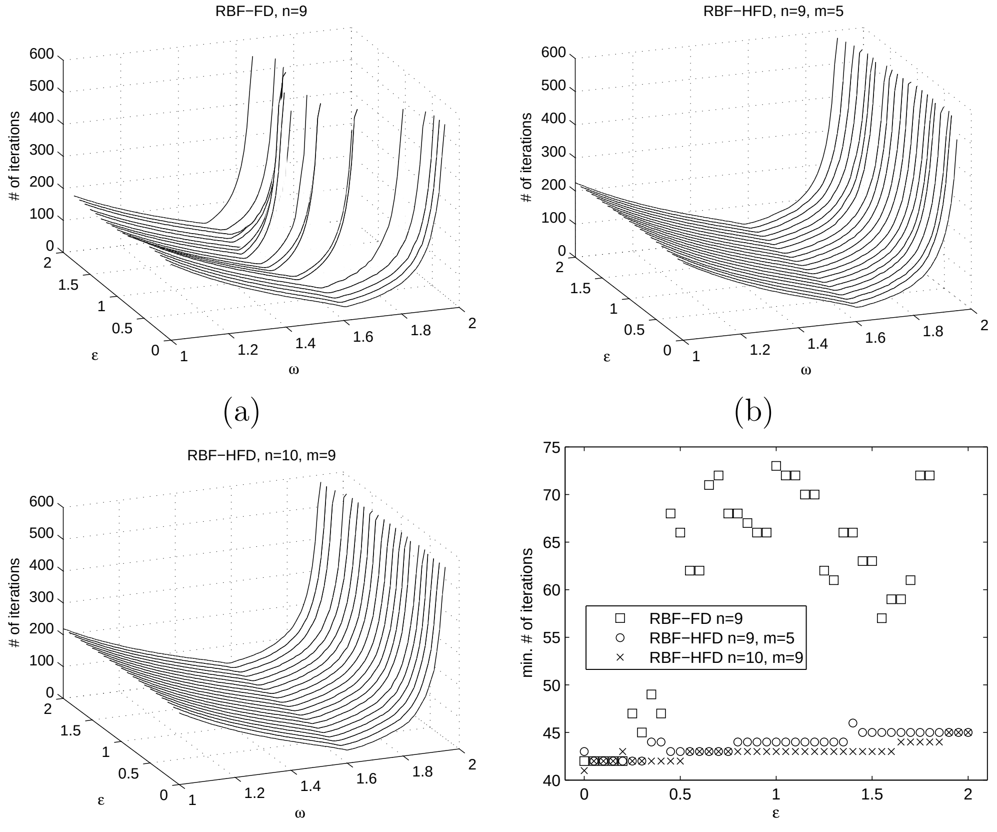


Fig. 4. Max norm error, as a function of ε, for the RBF-FD and HFD solutions of (36). The dashed lines correspond to the error in the standard FD2 and compact FD4 solutions of (36) using the polar mesh in Figure 3 (b).

compact stencil. However, for this example, any improvements appear to be lost for ε approximately > 0.55. The figure also illustrates that the optimal value of ε (the ε where the error reaches a minimum) is small (in magnitude), and nonzero, as is often the case for the RBF interpolation problem [33]. From the numerical results for limiting ($\varepsilon \to 0$) scattered node formulas in Table 2 we expect that the $n = 9$ node RBF-FD solution would be first order accurate, the $n = 9$, $m = 5$ node RBF-HFD solution would be second order accurate, and the $n = 10$, $m = 9$ node RBF-HFD solution would be third order accurate.

One very important observation we make from this example is that no RBF-FD formula satisfying the diagonal dominance property (35) could be found for $n \geq 10$ and $m = 0$ using the stencil selection algorithm above. However, if we choose $2 \leq m \leq 9$, such stencils could be found.

We next compare the RBF solutions to the standard FD solutions based on a uniform polar mesh. For the comparison, we require that the methods contain approximately the same number of boundary and interior nodes. The unstructured discretization in Figure 3 (a) contains 56 boundary and 144 interior nodes, while the numbers for the structured polar mesh in (b) are 50 and 151, respectively. We compare the results to the standard, five-node, second order FD scheme (FD2) and the standard, compact, nine-node, fourth order FD scheme (compact FD4) (see, for example, [36]). These solutions are included in Figure 4 as dashed lines. Comparing the FD2 and standard $n = 9$ RBF-FD solution, we see that for approximately $\varepsilon < 0.6$, the RBF solution is the clear winner. In fact, at the optimal value of ε, the RBF solution is over one order of magnitude more accurate. Comparing the compact FD4 solution, which uses nine function values and four derivative values, and the $n = 9$, $m = 5$ RBF-HFD solution, we see that the RBF solution is better for all



(a)

(b)

(c)

(d)

Fig. 5. (a)–(c) Convergence results of the SOR method for solving the linear systems associated with the Poisson model problem as a function of the shape parameter ε and SOR parameter ω. (d) Minimum number of iterations required for different values of ε.

values of ε approximately < 1.45. Again, at the optimal ε, the RBF solution is over one order of magnitude more accurate. Furthermore, the RBF-FD and HFD techniques generalize to more complex domains, while the standard FD2 and compact FD4 methods used here are specific to a disk.

We finally make comments on computing the RBF-FD and HFD solutions. The approximate RBF-FD or HFD solution $\mathbf{u}^*$ to (36) at the interior points of Ω can be expressed in terms of the linear system

$$L^c\mathbf{u}^* = (I - \widetilde{L^c})\mathbf{f} + \mathbf{b}, \tag{38}$$

where L^c is the matrix containing the RBF-FD or RBF-HFD weights for $\mathbf{u}$, $\widetilde{L^c}$ is the matrix containing the RBF-HFD weights for $\Delta\mathbf{u}$ (or zeros in the case of the RBF-FD formulas), and $\mathbf{b}$ contains the contributions from the boundary. L^c is sparse, non-symmetric, and since both the RBF-FD and HFD formulas are required to satisfy (35), it is also diagonally dominant. Many iterative techniques can be used for solving this system [37, p. 321], we have chosen the the classical successive over-relaxation (SOR) method. In Figure 5 (a)–(c) we display how the number of iterations necessary for convergence of SOR depend on ε and the relaxation parameter ω for the different RBF-FD methods. The iterations were terminated when the solution at the i^{th} iteration, $\mathbf{u}_i^*$, satisfied

$$\|L^c\mathbf{u}_i^* - ((I - L^{\widetilde{c}})\mathbf{f} + \mathbf{b})\|_\infty \leq 10^{-9}\left(\|L^c\|_\infty\|\mathbf{u}_i^*\|_\infty + \|(I - L^{\widetilde{c}})\mathbf{f} + \mathbf{b}\|_\infty\right) \; . \quad (39)$$

We can see from the figure that the optimal relaxation parameter depends quite significantly on the value of ε for the $n = 9$ RBF-FD solution. However, for both compact solutions, the optimal ω appears to be rather stable with respect to ε. The choice of $\omega \approx 1.6$ seems to be good for both compact methods. Figure 5 (d) shows the number of iterations necessary for convergence at the optimal value of ω for different values of ε. We can see from the figure that the minimum number of iterations is quite consistent for the compact methods but jumps around for the non-compact method. There also appears to be a slight increase in the number of iterations as ε increases. Efficient direct solvers based on the FFT may be used for the standard FD2 and compact FD4 polar methods [38] (but only in the specialized case a circular domain). If we had, however, used SOR then the minimum number of iterations would be 162 for FD2 and 328 for compact FD4, which are significantly higher than the standard RBF-FD and HFD methods.

5.3 Nonlinear equation: hybrid discretization

As a final example, we consider the nonlinear equation

$$\begin{aligned} \Delta u &= e^{-2x}u^3 \text{ in } \Omega = \left\{(x,y) \mid x^2 + y^2 \geq 0.4 \;\&\; -1 \leq x, y \leq 1\right\}, \\ u &= g \;\text{ on } \partial\Omega \;, \end{aligned} \quad (40)$$

where g is computed from the known solution

$$u(\underline{x}) = u(x,y) = e^x \tanh\frac{y}{\sqrt{2}}.$$

Note that Ω is square with a circular whole in the middle. To discretize this domain, we use a *hybrid* approach that combines an unstructured discretization near the hole with an equispaced discretization away from it. Figure 6 illustrates this discretization. We anticipate that this type of hybrid approach—using scattered nodes only where the geometry is more complex—will be a powerful application of the RBF-FD and HFD methods.

For each node on, and interior to the square enclosed with solid lines in Figure 6, we use scattered node RBF-HFD formulas with $n = 10$ and $m = 9$ to approximate the Laplacian. The scattered node stencils are chosen according to the stencil selection algorithm of the previous section. For all the other

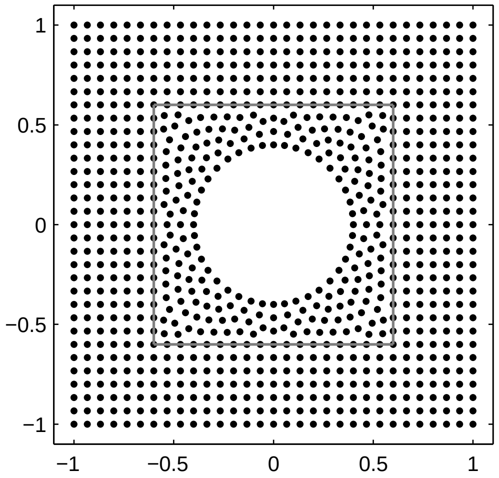


Fig. 6. Hybrid discretization for the problem in Section 5.3

interior nodes, we use the same $n = 9$, $m = 4$, RBF-HFD formula based on the second stencil in (34). The scattered node stencils total 252, while the regular stencils total 482.

Let $\underline{x}_i = (x_i, y_i)$, $i = 1, \ldots, N_{\mathcal{I}}$ denote all of the interior nodes of the discretization. Then, using the same notation as (38), we can write the approximation of (40) as

$$L^c \mathbf{u}^* = D(I - \widetilde{L^c})(\mathbf{u}^*)^3 + \mathbf{b},$$

where D is a diagonal matrix with entries $D_{ii} = e^{-2x_i}$, $i = 1, \ldots, N_{\mathcal{I}}$ and $\mathbf{b}$ incorporates the boundary conditions. To solve this nonlinear system of equations we use Newton-SOR [39, Sec.7.4]. Figure 7 displays the max norm error in the approximate solution as a function of ε. Similar to the previous example, each point on the curve corresponds to the error using the same value of ε in all the RBF-HFD stencils. We can see from the figure that the error decreases with ε until $\varepsilon = 0.15$, where it reaches a minimum value of $1.99 \cdot 10^{-8}$.

For the Newton-SOR method, the SOR relaxation parameter was fixed at $\omega = 1.635$ for all the Newton iterations and for all values of ε. With this value, the number of SOR sub-iterations required for solving the Jacobian system in any one of the Newton iterations varied between 63 and 72. The same stopping criterion as (39) was used for the SOR iterations, but with a tolerance of 10^{-10}. Finally, we used an initial guess of $u = 1$ for Newton's method. With this value, five Newton iterations were required for the residual to reach a tolerance $< 10^{-12}$ for all values of ε.

Remark 4 *In our experiments, we fixed ε in all of the RBF-FD and HFD formulas and then computed a solution. This may not, however, be the best strategy. If, for example, the scales associated with each of the RBF-FD or RBF-HFD stencils is significantly different, we may wish normalize ε for each*

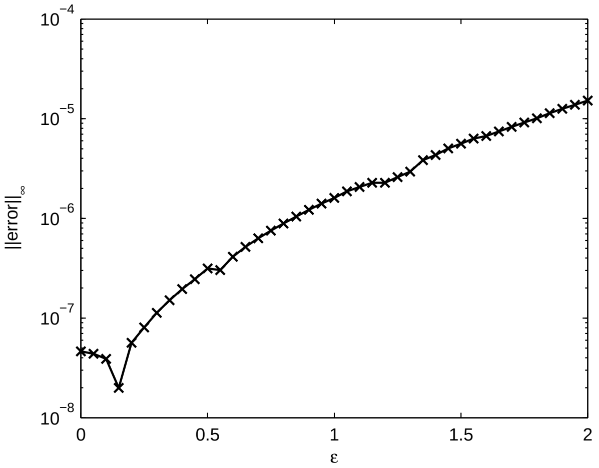


Fig. 7. Max norm error, as a function of ε, for the $n = 10$, $m = 9$ RBF-HFD solution of (40) using the hybrid discretization shown in Figure 6.

stencil. Two ideas for such a normalization are discussed by Shu et al. [7] and Cecil et al. [10].

6 Conclusions

In a similar style to how polynomials are used to generate FD and compact FD stencils for 1-D, we have here shown how RBFs can be used to create analogous formulas also for multidimensional scattered node layouts. We have also demonstrated that, when the stencil nodes are arranged accordingly, RBF-FD and HFD formulas in the $\varepsilon \to 0$ limit are equivalent to standard FD and compact FD formulas. In contrast to many methods, such as finite elements, the RBF-FD or HFD methods do not require the generation of global meshes. Furthermore, the number of space dimensions and the geometric complexity of the methods can all be arbitrary without adversely affecting either computational speed or algorithmic complexity. Tests with elliptic equations show that accuracy can be improved dramatically by using RBF-HFD formulas, and that it is imperative to use these formulas to preserve diagonal dominance. The latter property permits the use of fast iterative methods for computing the numerical solution. We believe—but it is yet to be fully explored—that the RBF-HFD approach will prove successful also for many further classes of PDEs.

Acknowledgements

The authors would like to thank the anonymous referees for their helpful suggestions in improving this paper.

References

[1] E. J. Kansa, Multiquadrics – a scattered data approximation scheme with applications to computational fluid-dynamics – II: Solutions to parabolic, hyperbolic and elliptic partial differential equations, Comput. Math. Appl. 19 (1990) 147–161.

[2] Y. C. Hon, X. Z. Mao, An efficient numerical scheme for Burgers' equation, Appl. Math. Comput. 95 (1998) 37–50.

[3] G. E. Fasshauer, Solving partial differential equations with radial basis functions: multilevel methods and smoothing, Adv. Comput. Math. 11 (1999) 139–159.

[4] E. Larsson, B. Fornberg, A numerical study of some radial basis function based solution methods for elliptic PDEs, Comput. Math. Appl. 46 (2003) 891–902.

[5] E. J. Kansa, Y. C. Hon, Circumventing the ill-conditioning problem with multiquadric radial basis functions: Applications to elliptic partial differential equations, Comput. Math. Appl. 39 (2000) 123–137.

[6] L. Ling, E. J. Kansa, A least-squares preconditioner for radial basis functions collocation methods., Adv. Comput. Math. 23 (2005) 31–54.

[7] C. Shu, H. Ding, K. S. Yeo, Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier-Stokes equations, Comput. Methods Appl. Mech. Engrg. 192 (2003) 941–954.

[8] C. Shu, H. Ding, Numerical comparison of least square-based finite difference (LSFD) and local multiquadric-differential quadrature (LMQDQ) methods, Comput. Math. Appl. submitted (2004).

[9] A. I. Tolstykh, M. V. Lipavskii, D. A. Shirobokov, High-accuracy discretization methods for solid mechanics, Arch. Mech. 55 (2003) 531–553.

[10] T. Cecil, J. Qian, S. Osher, Numerical methods for high dimensional Hamilton-Jacobi equations using radial basis functions, J. Comput. Phys. 196 (2004) 327–347.

[11] G. Wright, Radial basis function interpolation: Numerical and analytical developments, Ph.D. thesis, University of Colorado, Boulder (2003).

[12] B. Fornberg, A Practical Guide to Pseudospectral Methods, Cambridge University Press, Cambridge, 1996.

[13] B. Fornberg, Calculation of weights in finite difference formulas, SIAM Rev 40 (3) (1998) 685–691.

[14] R. Abgrall, On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation, J. Comput. Phys. 114 (1994) 45–58.

[15] W. Schönauer, T. Adolph, How we solve PDEs, J. Comput. Appl. Math. 131 (2001) 473–492.

[16] R. J. Y. McLeod, M. L. Baart, Geometry and Interpolation of Curves and Surfaces, Cambridge University Press, Cambridge, 1998.

[17] W. R. Madych, Miscellaneous error bounds for multiquadric and related interpolants, Comput. Math. Appl. 24 (1992) 121–138.

[18] B. Fornberg, N. Flyer, Accuracy of radial basis function interpolation and derivative approximations on 1-D infinite grids, Adv. Comput. Math. 23 (2005) 5–20.

[19] T. A. Driscoll, B. Fornberg, Interpolation in the limit of increasingly flat radial basis functions, Comput. Math. Appl. 43 (2002) 413–422.

[20] B. Fornberg, G. Wright, E. Larsson, Some observations regarding interpolants in the limit of flat radial basis functions, Comput. Math. Appl. 47 (2004) 37–55.

[21] E. Larsson, B. Fornberg, Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions, Comput. Math. Appl. 49 (2005) 103–130.

[22] R. Schaback, Multivariate interpolation by polynomials and radial basis functions, Constr. Approx. 21 (2005) 293–317.

[23] L. Collatz, The Numerical Treatment of Differential Equations, Springer Verlag, Berlin, 1960.

[24] S. K. Lele, Compact finite difference schemes with spectral-like resolution, J. Comput. Phys. 103 (1992) 16–42.

[25] C. A. Micchelli, Interpolation of scattered data: distance matrices and conditionally positive definite functions, Constr. Approx. 2 (1986) 11–22.

[26] R. Schaback, Error estimates and condition numbers for radial basis function interpolants, Adv. Comput. Math. 3 (1995) 251–264.

[27] J. Yoon, Spectral approximation orders of radial basis function interpolation on the Sobolev space, SIAM J. Math. Anal. 23 (4) (2001) 946–958.

[28] R. L. Hardy, Multiquadric equations of topograpy and other irregular surfaces, J. Geophy. Res. 76 (1971) 1905–1915.

[29] R. L. Hardy, Theory and applications of the multiquadric-biharmonic method: 20 years of discovery, Comput. Math. Appl. 19 (1990) 163–208.

[30] G. E. Fasshauer, Hermite interpolation with radial basis functions on spheres, Adv. Comput. Math. 10 (1999) 81–96.

[31] X. Sun, Scattered Hermite interpolation using radial basis functions, Linear Algebra Appl. 207 (1994) 135–146.

[32] Z. Wu, Hermite-Birkhoff interpolation of scattered data by radial basis functions, Approx. Theory Appl. 8 (2) (1992) 1–10.

[33] B. Fornberg, G. Wright, Stable computation of multiquadric interpolants for all values of the shape parameter, Comput. Math. Appl. 48 (2004) 853–867.

[34] B. Fornberg, E. Larsson, G. Wright, A new class of oscillatory radial basis functions, Comput. Math. Appl.To appear (2005).

[35] G. R. Liu, Mesh Free Methods: moving beyond the finite element method, CRC Press, New York, 2002.

[36] R. C. Mittal, S. Gahlaut, High-order finite-difference schemes to solve Poisson's equation in polar coordinates, IMA J. Num. Anal. 11 (1991) 261–270.

[37] J. W. Demmel, Numerical Linear Algebra, SIAM, Philadelphia, 1997.

[38] M.-C. Lai, W.-C. Wang, Fast direct solvers for Poisson equation on 2D polar and spherical geometries, Numer. Methods for Partial Differential Equations 18 (2002) 56–68.

[39] J. M. Ortega, W. C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, SIAM, Philadelphia, 2000.