A Robust RBF-FD Formulation based on Polyharmonic Splines and Polynomials

by

G. A. Barnett

B.S., Boise State University, 2009

M.S., University of Colorado, 2011

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Applied Mathematics

2015

This thesis entitled:
A Robust RBF-FD Formulation based on Polyharmonic Splines and Polynomials
written by G. A. Barnett
has been approved for the Department of Applied Mathematics

Dr. Natasha Flyer

Prof. Bengt Fornberg

Dr. Louis Wicker

Prof. Gunnar Martinsson

Prof. Keith Julien

Date

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Barnett, G. A. (Ph.D., Applied Mathematics)

A Robust RBF-FD Formulation based on Polyharmonic Splines and Polynomials

Thesis directed by Dr. Natasha Flyer and Prof. Bengt Fornberg

We introduce a local method based on radial basis function-generated finite differences (RBF-FD) for interpolation and the numerical solution of partial differential equations (PDEs). The method uses polyharmonic spline (PHS) RBFs together with polynomials to derive differentiation weights on different node configurations. The formulation is explored in three directions: (i) Interpolation and approximation of differential operators, (ii) Elliptic PDEs, and (iii) Hyperbolic PDEs.

In particular, the novel RBF-FD methodology is applied to standard test cases in numerical weather prediction, modeled by the compressible Navier-Stokes equations in 2D. Furthermore, the evaluation of the method on different node layouts, Cartesian, hexagonal, and scattered, is studied. The RBF-FD implementation acts as an extension of conventional finite-differences, achieving high accuracy on scattered nodes with no need for a computational mesh.

Contents

Chapter

Tables

Table

Figures

Figure

Chapter 1

Introduction

This thesis explores a new method based on the polyharmonic spline (PHS) radial basis function (RBF) with polynomial augmentation for local interpolation and derivative approximation. RBF-based methods are attractive because they are simple to apply and geometrically flexible, capable of handling complicated domains on scattered nodes with no need for a computational mesh [11]. Although RBF-generated finite-differences (RBF-FD) has been developed only over the last decade or so, the foundation of the RBF methodology has a long history, with important theoretical results stretching back 80 years [27, 4]. The discovery of the RBF interpolation method itself is usually attributed to Hardy in 1971, who used the Multiquadric (MQ) RBF for scattered data interpolation in 2-D [19] to solve problems in topography. Nearly 20 years later, the first use of RBFs for derivative approximation was attributed to Kansa in 1990, when he calculated the derivative of Hardy's MQ interpolant and used it to generate differentiation weights and solve simple PDEs [21, 22]. Global RBF approximations using Kansa's method are very accurate (especially on domains without boundaries like the surface of the sphere) and have been shown to compete favorably even for large scale simulations in geophysical fluid dynamics [31, 8]. On the other hand, the switch from global to local RBF methods decreases computational cost dramatically, making local methods more attractive for future applications to fully 3-D simulations.

The content in Chapter 6 and the three papers in Appendices A-C are novel, with the author's main contributions appearing in Chapter 6 (which remains to be organized into manuscript format) and the paper in Appendix C.

Chapter 2

RBF-generated Finite Differences

Radial basis function generated finite differences (RBF-FD) is a relatively new method, developed over the last decade, for creating differentiation weights on scattered nodes in more than one spatial dimension. Although the method has been implemented in various flavors and contexts in the last ten years, the first survey articles on RBF-FD are just now emerging [11, 12]. The method leverages the powerful nonsingularity results from RBF interpolation theory to produce a sparse differentiation matrix (DM). Once the DM is formed, it can be used repeatedly for spatial derivative approximation. In other words, the procedure for applying the RBF-FD method is conceptually identical to that of the conventional finite-difference method, but RBF-FD can be applied on scattered nodes in more than one spatial dimension. For example, Gaussian-based RBF-FD was used successfully in [8] for nonlinear transport using scattered nodes on a sphere, and multiquadric RBFs were used in [26] for the unsteady Navier-Stokes equations. However, it is difficult to achieve stable explicit time-stepping using the original RBF-FD method in the presence of boundaries. In [8], the computational domain is a sphere so there are no boundaries, and in [26], the authors use very small stencil-sizes ($n = 5$ and $n = 9$ in 2-D), which allows for stable time-stepping near the boundaries, but restricts the spatial approximations to low-order. In this thesis we advocate an improvement to the RBF-FD method which reduces edge effects and achieves formal polynomial accuracy, while still taking advantage of the guaranteed nonsingularity of the RBF collocation matrix (see Section 3.1).

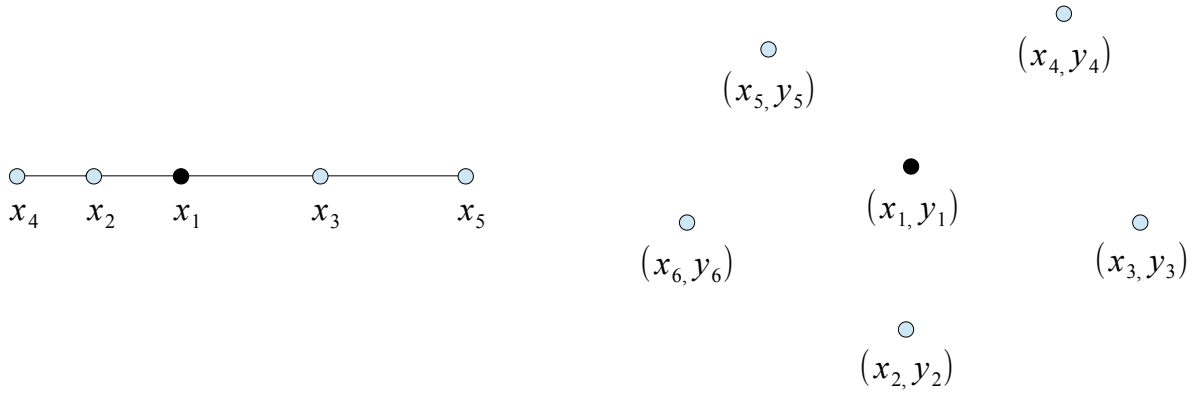

Figure 2.1: An appropriate stencil for polynomial-based FD (left), and RBF-FD (right). A circle represents a node and a black circle represents an evaluation point. Note that practical RBF-FD stencils in 2-D have more than 6 scattered nodes, and that conventional FD methods are usually applied on equi-spaced nodes, although this is not required.

In the RBF-FD method on scattered nodes, every stencil has a different shape. Approximation weights at some location $\mathbf{x}_e$ are derived locally, with radial functions centered only at the n nearest neighbors of $\mathbf{x}_e$ (See Figure 2.2). For a given differential operator, approximation weights are derived at every node. However, once these weights are known, they can be arranged in a differentiation matrix and used repeatedly, which makes this a pre-processing step for the numerical solution of PDEs. Additionally, the calculation of the differentiation weights for a given stencil is completely independent from similar calculations for the other stencils, so the weights can easily be generated in parallel.

In order to calculate the weights for local derivative approximations, the index of the n nearest neighbors needs to be found for each node. Luckily, this process has become extremely fast, especially in 2 and 3 spatial dimensions, and the nearest neighbors can be found quickly using available routines. For example, in MATLAB the k-d tree algorithm is accessible via the built-in knnsearch function. Along with calculating differentiation weights, the nearest-neighbor search is the other pre-processing step associated with RBF-FD.

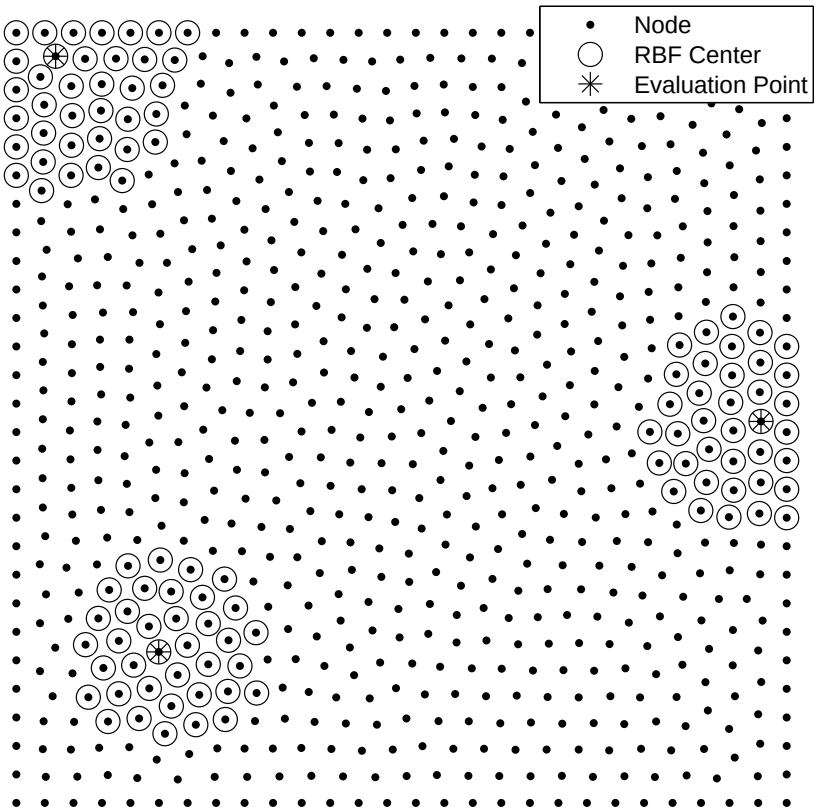

Figure 2.2: Three examples of 37-node RBF-FD stencils on a set of scattered nodes in 2-D. Each node has a corresponding stencil.

2.1 Derivation of Conventional FD Weights

As an introduction to the RBF-FD method, we review a derivation of the traditional polynomial-based finite-difference method. There are many possible derivations, but the one that most closely matches the RBF-FD process involves finding the analytical derivative of the Lagrange interpolating polynomial and evaluating it at the evaluation point, x_e. Suppose we have the five nodes $\{x_i\}_{i=1}^5$ (as in Figure 2.1), and the corresponding function values $\{f_i\}_{i=1}^5$. The goal is to find weights $\{w_i\}_{i=1}^5$ so that

$$f'(x_e) \approx \sum_{i=1}^{5} w_i f_i.$$

In order to derive these weights, assume there is some underlying interpolant $s(x)$, and that it is a linear combination of the first five monomial basis functions 1, x, x^2, x^3, and x^4:

$$s(x) = \lambda_1 + \lambda_2 x + \lambda_3 x^2 + \lambda_4 x^3 + \lambda_5 x^4.$$

Next, force the interpolant $s\,(x)$ to match the data at the node locations. Doing so induces a linear system for the unknowns $\{\lambda_i\}_{i=1}^5$:

$$\begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 & x_1^4 \\ 1 & x_2 & x_2^2 & x_2^3 & x_2^4 \\ 1 & x_3 & x_3^2 & x_3^3 & x_3^4 \\ 1 & x_4 & x_4^2 & x_4^3 & x_4^4 \\ 1 & x_5 & x_5^2 & x_5^3 & x_5^4 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{bmatrix} \quad \Longrightarrow \quad \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 & x_1^4 \\ 1 & x_2 & x_2^2 & x_2^3 & x_2^4 \\ 1 & x_3 & x_3^2 & x_3^3 & x_3^4 \\ 1 & x_4 & x_4^2 & x_4^3 & x_4^4 \\ 1 & x_5 & x_5^2 & x_5^3 & x_5^4 \end{bmatrix}^{-1} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{bmatrix}.$$

Finally, find the derivative of the interpolant at the evaluation point x_e:

$$\begin{aligned} s'\,(x_e) =& 0 + \lambda_2 + 2\lambda_3 x_e + 3\lambda_4 x_e^2 + 4\lambda_5 x_e^3 \\ =& \begin{bmatrix} 0 & 1 & 2x_e & 3x_e^2 & 4x_e^3 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \end{bmatrix} \\ =& \left(\begin{bmatrix} 0 & 1 & 2x_e & 3x_e^2 & 4x_e^3 \end{bmatrix} \begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 & x_1^4 \\ 1 & x_2 & x_2^2 & x_2^3 & x_2^4 \\ 1 & x_3 & x_3^2 & x_3^3 & x_3^4 \\ 1 & x_4 & x_4^2 & x_4^3 & x_4^4 \\ 1 & x_5 & x_5^2 & x_5^3 & x_5^4 \end{bmatrix}^{-1} \right) \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{bmatrix}. \end{aligned}$$

Once multiplied, the resulting row-vector in parentheses gives the FD weights for approximating $f'\,(x_e)$. The applicability of the method relies on the nonsingularity of the well-known Vandermonde matrix, which is guaranteed nonsingular so long as the nodes are distinct. The method produces

weights which are exact for polynomials up to some degree, which implies via Taylor's theorem an algebraic order of convergence to the true value, provided f is sufficiently smooth. The above procedure can be applied to generate weights on any set of distinct nodes in 1-D. In particular, it can be used to derive the well-known $O\left(h^2\right)$ weights for approximating the first and second derivative on equi-spaced nodes:

$$\begin{aligned} f'(x) =& \frac{-\frac{1}{2}f(x-h) + \frac{1}{2}f(x+h)}{h} + O\left(h^2\right), \\ f''(x) =& \frac{f(x-h) - 2f(x) + f(x+h)}{h^2} + O\left(h^2\right). \end{aligned}$$

The current RBF-FD method is derived using a very similar process. The weights are extracted by finding the analytical derivative of the interpolant (based on PHS RBFs and polynomials in the RBF-FD case) at some evaluation point $\mathbf{x}_e \in \mathbb{R}^d$.

2.2 Derivation of RBF-FD Weights

Traditional finite difference (FD) methods are very effective in one spatial dimension, or when the geometry is sufficiently simple that it can be treated one dimension at a time. However, polynomial interpolation on scattered nodes is notoriously misbehaved in higher dimensions, so the conventional FD method does not easily generalize. Given data on some very reasonable set of nodes 2-D, it may still be impossible to find a unique linear combination of polynomial basis functions which passes through each data point. On the other hand, no such restriction exists for RBF interpolation, which is guaranteed (for certain choices of the RBF) to work on any collection of distinct nodes and in any number of spatial dimensions [6].

Except for the PHS RBFs, the others in Table 2.1 are infinitely smooth and depend on ε, the so-called shape parameter, which controls how flat the function is. Smaller values of ε lead to flatter functions and usually a more accurate interpolant. However, the standard approximation procedure becomes numerically ill-conditioned as ε decreases. The method being advocated here does not rely on a shape parameter. We use a finitely-differentiable PHS RBF, together with some polynomial functions, to derive the approximation weights.

Name (Acronym)	*Function* $\phi(r)$	*Condition*	C^∞
Polyharmonic Spline (PHS)	r^m	$m \in 2\mathbb{N}-1$	No
	$r^m \log r$	$m \in 2\mathbb{N}$	No
Inverse Quadratic (IQ)	$\frac{1}{1+(\varepsilon r)^2}$	$\varepsilon \in \mathbb{R}$	Yes
Multiquadric (MQ)	$\sqrt{1+(\varepsilon r)^2}$	$\varepsilon \in \mathbb{R}$	Yes
Gaussian (GA)	$e^{-(\varepsilon r)^2}$	$\varepsilon \in \mathbb{R}$	Yes
Inverse Multiquadric (IMQ)	$\frac{1}{\sqrt{1+(\varepsilon r)^2}}$	$\varepsilon \in \mathbb{R}$	Yes

Table 2.1: Some relevant RBFs.

Let $\phi : [0,\infty) \to \mathbb{R}$ be an RBF and let $\{p_j(\mathbf{x})\}_{j=1}^k$ be a basis for polynomials up to some degree in $\mathbb{R}^d$. Assume we are constructing weights for a differential operator L using the n distinct nodes $\mathbf{x}_1, \mathbf{x}_2 \ldots, \mathbf{x}_n \in \mathbb{R}^d$ with corresponding function values $f_1, f_2, \ldots, f_n$ (either known or unknown), and that we are approximating Lf at the evaluation point $\mathbf{x}_e$. In other words, we seek weights $w_1, w_2, \ldots, w_n$ such that

$$Lf(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_e} \approx \sum_{i=1}^{n} w_i f_i.$$

Finally, let $\|\cdot\| = \|\cdot\|_2$ be the Euclidean norm on $\mathbb{R}^d$. The combined RBF and polynomial interpolant is assumed to be a linear combination of the radial and polynomial functions, thus it takes the form

$$s(\mathbf{x}) = \sum_{j=1}^{n} \lambda_j \phi(\|\mathbf{x}-\mathbf{x}_j\|) + \sum_{j=1}^{k} \mu_j p_j(\mathbf{x}). \tag{2.1}$$

To solve for the unknowns, force the interpolant to match the data at the node locations:

$$s(\mathbf{x}_i) = \sum_{j=1}^{n} \lambda_j \phi(\|\mathbf{x}_i-\mathbf{x}_j\|) + \sum_{j=1}^{k} \mu_j p_j(\mathbf{x}_i) = f_i, \qquad i = 1,2,3,\ldots,n, \tag{2.2}$$

and enforce k matching constraints to maintain a square symmetric linear system, and also to minimize far-field growth [10]:

$$\sum_{j=1}^{n} \lambda_j p_i(\mathbf{x}_j) = 0, \qquad i = 1,2,3,\ldots,k. \tag{2.3}$$

The above are arranged in a linear system of $n+k$ equations for the $n+k$ unknowns $\{\lambda_j\}_{j=1}^{n} \cup \{\mu_j\}_{j=1}^{k}$. However, before writing down the linear system, define the standard A-matrix, the polynomial matrix P, the row-vectors $\underline{b}$ and $\underline{c}$, and the column-vectors $\underline{\lambda}$, $\underline{\mu}$ and $\underline{f}$ by

$$A = \begin{bmatrix} \phi\left(\|\mathbf{x}_1 - \mathbf{x}_1\|\right) & \phi\left(\|\mathbf{x}_1 - \mathbf{x}_2\|\right) & \cdots & \phi\left(\|\mathbf{x}_1 - \mathbf{x}_n\|\right) \\ \phi\left(\|\mathbf{x}_2 - \mathbf{x}_1\|\right) & \phi\left(\|\mathbf{x}_2 - \mathbf{x}_2\|\right) & \cdots & \phi\left(\|\mathbf{x}_2 - \mathbf{x}_n\|\right) \\ \vdots & \vdots & \ddots & \vdots \\ \phi\left(\|\mathbf{x}_n - \mathbf{x}_1\|\right) & \phi\left(\|\mathbf{x}_n - \mathbf{x}_2\|\right) & \cdots & \phi\left(\|\mathbf{x}_n - \mathbf{x}_n\|\right) \end{bmatrix},$$

$$P = \begin{bmatrix} p_1\left(\mathbf{x}_1\right) & p_2\left(\mathbf{x}_1\right) & \cdots & p_k\left(\mathbf{x}_1\right) \\ p_1\left(\mathbf{x}_2\right) & p_2\left(\mathbf{x}_2\right) & \cdots & p_k\left(\mathbf{x}_2\right) \\ \vdots & \vdots & \ddots & \vdots \\ p_1\left(\mathbf{x}_n\right) & p_2\left(\mathbf{x}_n\right) & \cdots & p_k\left(\mathbf{x}_n\right) \end{bmatrix},$$

$$\underline{b} = \begin{bmatrix} L\phi\left(\|\mathbf{x} - \mathbf{x}_1\|\right)\big|_{\mathbf{x}=\mathbf{x}_e} & L\phi\left(\|\mathbf{x} - \mathbf{x}_2\|\right)\big|_{\mathbf{x}=\mathbf{x}_e} & \cdots & L\phi\left(\|\mathbf{x} - \mathbf{x}_n\|\right)\big|_{\mathbf{x}=\mathbf{x}_e} \end{bmatrix},$$
$$\underline{c} = \begin{bmatrix} Lp_1\left(\mathbf{x}\right)\big|_{\mathbf{x}=\mathbf{x}_e} & Lp_2\left(\mathbf{x}\right)\big|_{\mathbf{x}=\mathbf{x}_e} & \cdots & Lp_k\left(\mathbf{x}\right)\big|_{\mathbf{x}=\mathbf{x}_e} \end{bmatrix},$$

$$\underline{\lambda} = \begin{bmatrix} \lambda_1 & \lambda_2 & \cdots & \lambda_n \end{bmatrix}^T, \qquad \underline{\mu} = \begin{bmatrix} \mu_1 & \mu_2 & \cdots & \mu_k \end{bmatrix}^T, \qquad \underline{f} = \begin{bmatrix} f_1 & f_2 & \cdots & f_n \end{bmatrix}^T.$$

Enforcing the n collocation conditions (2.2) and k matching constraints (2.3) yields the linear system

$$\begin{bmatrix} A & P \\ P^T & O \end{bmatrix} \begin{bmatrix} \underline{\lambda} \\ \underline{\mu} \end{bmatrix} = \begin{bmatrix} \underline{f} \\ \underline{0} \end{bmatrix} \quad \Longrightarrow \quad \begin{bmatrix} \underline{\lambda} \\ \underline{\mu} \end{bmatrix} = \begin{bmatrix} A & P \\ P^T & O \end{bmatrix}^{-1} \begin{bmatrix} \underline{f} \\ \underline{0} \end{bmatrix}.$$

Evaluating the derivative of the interpolant at $\mathbf{x}_e$ gives

$$\begin{aligned} Ls\left(\mathbf{x}\right)\big|_{\mathbf{x}=\mathbf{x}_e} &= \sum_{j=1}^{n} \lambda_j \; L\phi\left(\|\mathbf{x}-\mathbf{x}_j\|\right)\big|_{\mathbf{x}=\mathbf{x}_e} + \sum_{j=1}^{k} \mu_j \; Lp_j\left(\mathbf{x}\right)\big|_{\mathbf{x}=\mathbf{x}_e} \\ &= \begin{bmatrix} \underline{b} & \underline{c} \end{bmatrix} \begin{bmatrix} \underline{\lambda} \\ \underline{\mu} \end{bmatrix} \\ &= \left(\begin{bmatrix} \underline{b} & \underline{c} \end{bmatrix} \begin{bmatrix} A & P \\ P^T & O \end{bmatrix}^{-1} \right) \begin{bmatrix} \underline{f} \\ \underline{0} \end{bmatrix}. \end{aligned}$$

Once multiplied, the resulting row-vector in parentheses gives the weights $\begin{bmatrix} \underline{w} & \underline{v} \end{bmatrix}$ for approximating $Lf\left(\mathbf{x}\right)\big|_{\mathbf{x}=\mathbf{x}_e}$, where $\underline{w} \in \mathbb{R}^{1\times n}$ and $\underline{v} \in \mathbb{R}^{1\times k}$. However, we throw out the last k weights contained in $\underline{v}$, because these are multiplied by zero. Taking the transpose, the weights are seen to be contained in the solution to the linear system

$$\begin{bmatrix} A & P \\ P^T & O \end{bmatrix} \begin{bmatrix} \underline{w}^T \\ \underline{v}^T \end{bmatrix} = \begin{bmatrix} \underline{b}^T \\ \underline{c}^T \end{bmatrix}$$

where $\underline{v}$ is disregarded after solving. Similar to the case of conventional finite differences, the applicability of the present RBF-FD method depends on the nonsingularity of the matrix $\begin{bmatrix} A & P \\ P^T & O \end{bmatrix}$, which will be established in Chapter 3.

Any function of the form (2.1) that satisfies all of the constraints in (2.3) will be reproduced exactly by the interpolation procedure. In particular, if the underlying function is purely polynomial, and can be written as a combination of the k polynomial basis functions, then there will be no error associated with the combined RBF and polynomial approximation. Weights that are derived using the process above will be exact for all polynomials up to some degree. Again via Taylor's theorem, we should expect algebraic convergence of the RBF-FD approximation to the true value, provided f is smooth enough. The formal accuracy that results from polynomial reproduction might seem like a crucial characteristic for local methods, but previous implementations of RBF-FD typically have no associated formal convergence rate.

Chapter 3

Polyharmonic Splines

For global RBF approximations, if the RBF is smooth and the underlying function resides in an appropriate native space, then spectral convergence can be achieved [6]. For this reason, the first RBF-based derivative approximations used the infinitely smooth multiquadric RBF [21, 22], rather than a finitely-differentiable one, such as a polyharmonic spline. The logic behind this choice makes sense in the global case, but it breaks down when one transitions to local methods (RBF-FD). With a fixed stencil-size, one should not expect that increasing the total number of nodes will yield spectral convergence, but rather algebraic convergence. Therefore, unless the stencil-size is very large, using an infinitely-differentiable RBF is unnecessary. The conditioning and simplicity of the method can be improved with no cost to accuracy by switching to a finitely-differentiable PHS RBF and including an appropriate number of polynomial functions in the interpolant (see Theorem 2). Switching to a PHS RBF removes the need for a shape parameter, so approximations can be made on reasonable node-sets without worrying about choosing the right ε for good conditioning. Additionally, while polynomials in global approximations can induce the Runge phenomenon, local methods are somewhat protected since evaluations are typically made near the center of the stencil. Finally, approximations based strictly on RBFs suffer from stagnation error, but the inclusion of polynomials with PHS RBFs bypasses this issue entirely (see Appendix A).

3.1 Guaranteed Nonsingularity of the Collocation Matrix

Suppose that $\{(\mathbf{x}_k, f_k)\}_{k=1}^n$ is a collection of n distinct nodes in $\mathbb{R}^d$ with corresponding real data. For many choices of the RBF $\phi(r)$, the linear system

$$A\underline{\lambda} = \underline{f}, \tag{3.1}$$

with A defined element-wise by

$$a_{ij} = \phi\left(\|\mathbf{x}_i - \mathbf{x}_j\|_2\right), \qquad i, j = 1, 2, \ldots, n, \tag{3.2}$$

is uniquely solvable. Although this has not been shown for polyharmonic spline RBFs, a minor adjustment to the linear system in (3.1) restores unique solvability and also improves the interpolant (especially for local approximations).

Degree	1-D $[\mathbf{x} = x]$	2-D $[\mathbf{x} = (x, y)]$	3-D $[\mathbf{x} = (x, y, z)]$
0	1	1	1
1	x	x, y	x, y, z
2	x^2	x^2, xy, y^2	$x^2, xy, xz, y^2, yz, z^2$
3	x^3	x^3, x^2y, xy^2, y^3	$x^3, x^2y, x^2z, xy^2, xyz, xz^2, y^3, y^2z, yz^2, z^3$

Table 3.1: Simple basis for polynomials up to third degree in 1, 2, and 3 spatial dimensions. Unless polynomials of very high degree are included, this simple style of basis suffices to augment RBFs with no ill effects on conditioning.

Let $\{p_j(\mathbf{x})\}_{j=1}^{\binom{\ell+d}{d}}$ be a basis for polynomials up to degree ℓ in $\mathbb{R}^d$. Although any basis will do, for local approximations a basis of the form shown in Table 3.1 is convenient and effective. The linear system for interpolation, modified to include polynomial support, looks like

$$\begin{bmatrix} A & P \\ P^T & O \end{bmatrix} \begin{bmatrix} \underline{\lambda} \\ \underline{\mu} \end{bmatrix} = \begin{bmatrix} \underline{f} \\ \underline{0} \end{bmatrix}, \tag{3.3}$$

where A is defined as in (3.2), P is defined element-wise by

$$p_{ij} = p_j(\mathbf{x}_i), \qquad i = 1, 2, \ldots, n, \qquad j = 1, 2, \ldots \binom{\ell + d}{d},$$

and O is a square matrix of zeros with size $\binom{\ell+d}{d}$.

To cover briefly the theoretical background for the solvability of (3.3), the following definitions and theorems are taken directly from [6], or written less generally to accomodate only the relevant results for polyharmonic splines.

Definition 1. *A real-valued continuous even function Φ is called conditionally positive definite of order s on $\mathbb{R}^d$ if*

$$\sum_{j=1}^{N} \sum_{j=1}^{N} c_j \overline{c_k} \Phi(\mathbf{x}_j - \mathbf{x}_k) \geq 0 \tag{3.4}$$

for any N distinct points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N \in \mathbb{R}^d$, and $c_1, c_2, \ldots, c_N \in \mathbb{R}$ satisfying

$$\sum_{j=1}^{N} c_j p(\mathbf{x}_j) = 0$$

for any real-valued polynomial p of degree at most $s-1$. Φ is called strictly conditionally positive definite of order s on $\mathbb{R}^d$ if the quadratic form (3.4) is zero only for $c_1 = c_2 = \cdots = c_N = 0$.

Lemma 1. *A function that is strictly conditionally positive definite of order s on $\mathbb{R}^d$ is also strictly conditionally positive definite of any higher order.*

Theorem 1. *Suppose the complex-valued function Φ possesses a generalized Fourier transform $\hat{\Phi}$ of order s which is continuous on $\mathbb{R}^d \setminus \{\mathbf{0}\}$. Then Φ is strictly conditionally positive definite of order s if and only if $\hat{\Phi}$ is non-negative and non-vanishing.*

The function

$$\Phi(\mathbf{x}) = \|\mathbf{x}\|^m, \quad \mathbf{x} \in \mathbb{R}^d, \quad m \in 2\mathbb{N} - 1,$$

has generalized Fourier transform

$$\hat{\Phi}(\omega) = \frac{2^{m+d/2} \Gamma\left(\frac{d+m}{2}\right)}{\Gamma(-m/2)} \|\omega\|^{-m-d}, \quad \omega \neq \mathbf{0},$$

of order $(m+1)/2$. Therefore, by Theorem 1, Φ is strictly conditionally positive definite of order $(m+1)/2$. As a consequence of Lemma 1, Φ is also strictly conditionally positive definite of any higher order.

Theorem 2. *If the real-valued even function Φ is strictly conditionally positive definite of order s on $\mathbb{R}^d$ and the points $\mathbf{x}_1, \mathbf{x}_2 \ldots, \mathbf{x}_N$ form an $(s-1)$-unisolvent set, then the system of linear equations* (3.3) *is uniquely solvable.*

As a result of Theorem 2, we can conclude that using $\phi(r) = r^m$ with polynomials up to degree $\ell \geq (m-1)/2$ implies that (3.3) is uniquely solvable, provided the nodes are ℓ-unisolvent. For example, if $\phi(r) = r^3$, then polynomials up to degree 1 or higher should be included, and if $\phi(r) = r^5$, then polynomials up to degree 2 or higher should be included, and so on.

It should be noted that the matrix P which appears in (3.3) is not a square interpolation matrix, but rather a rectangular least-squares matrix. The number of rows of P is determined by the stencil-size n, while the number of columns of P is determined by the size of the polynomial basis, which always contains $\binom{\ell+d}{d}$ polynomials up to degree ℓ in $\mathbb{R}^d$. By keeping the level of polynomial support fixed while increasing the stencil-size, the likelihood of failing to meet the unisolvency requirement in Theorem 2 decreases to the point that it is no longer an issue, even on thousands of RBF-FD stencils on scattered nodes.

3.2 Global Approximations

Although the current study focuses on local methods, it is worthwhile to review the key properties of global PHS approximations versus infinitely smooth alternatives such as the MQ approach of Hardy and Kansa [19, 21, 22]. Reviewing the convergence properties for global interpolation with various RBFs gives insights into why polyharmonic splines have thus far not been popular for derivative approximations. Global PHS approximations converge algebraically to the target function, with the rate of convergence increasing as the exponent in the RBF increases. Similarly, the condition number of the collocation matrix diverges algebraically, and this rate also increases

with the exponent in the RBF (see Figure 3.2). On the other hand, global approximations based on infinitely smooth RBFs, such as the MQ, converge spectrally, and their condition numbers diverge exponentially.

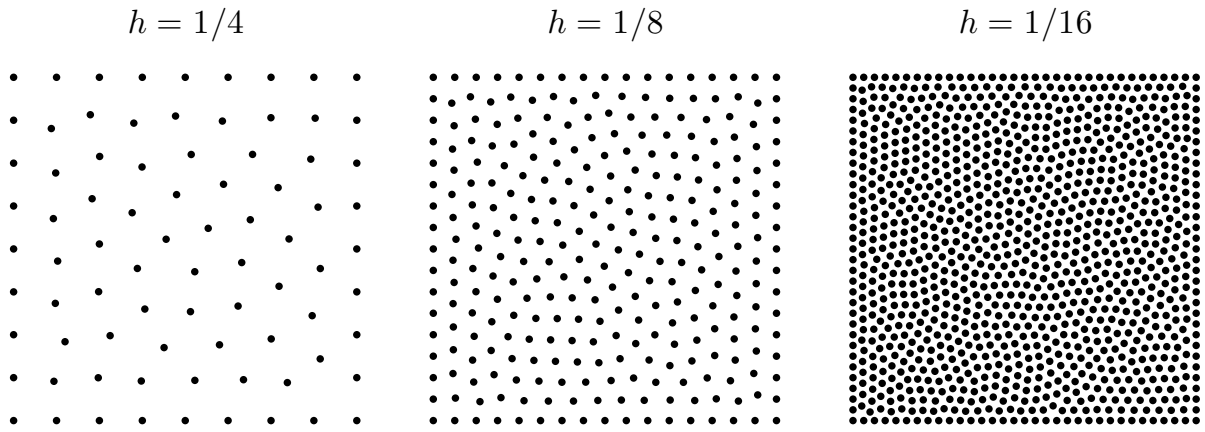

Figure 3.1: Three of the scattered node-sets for global RBF interpolation on $[-1, 1] \times [-1, 1]$.

For a test problem to explore different global methods, we interpolate the simple function

$$f(x, y) = \cos(\pi x) \sin(\pi y),$$

for $(x, y) \in [-1, 1] \times [-1, 1]$, using scattered nodes of increasing density, and calculate the relative ℓ_2 error over $10,000$ semi-uniformly scattered Halton nodes. In the case of the PHS RBFs, we include only the number of polynomials required for nonsingularity (See Theorem 2).

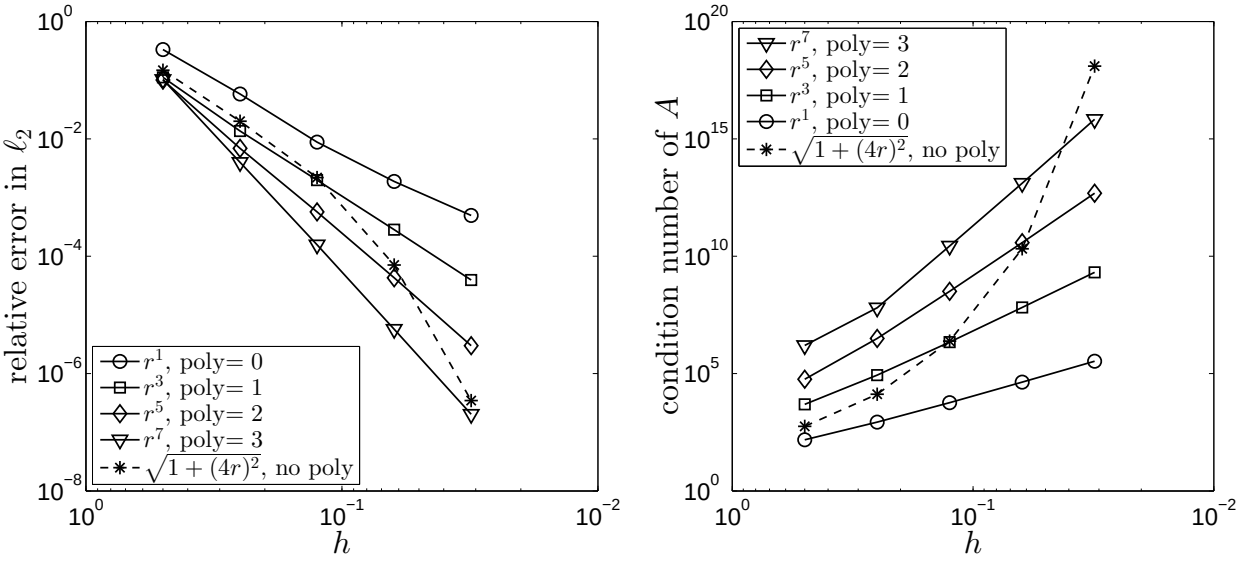

Figure 3.2: Error and condition number for global PHS and global MQ interpolation on scattered nodes. $h = \frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$, and $\frac{1}{32}$. Three of the node-sets are shown in Figure 3.1.

Looking at Figure 3.2, we see that the MQ interpolant converges spectrally, while the PHS interpolant with polynomial augmentation converges only algebraically, according to the exponent in the RBF. In the global case, in order to achieve the best convergence rate, it makes sense to use an infinitely differentiable RBF, but we will see that the convergence rate in local RBF approximations is not affected by the type of RBF. Since Kansa's original RBF-based derivative approximation was global, the MQ RBF was a good choice. However, once we transition to local methods, the choice of RBF becomes less important (See Chapter 4). In this case, the PHS RBFs become more attractive because they are simple, having no shape parameter to choose for good conditioning. As long as the stencil-size remains fixed, the RBF-FD method with PHS and polynomials is applied in exactly the same way, regardless of the density of the nodes.

Chapter 4

PHS+Polynomials in the RBF-FD Method I: Approximation

The most basic application of RBF-FD is the local interpolation of some given data $\{(\mathbf{x}_i, f_i)\}_{i=1}^N$, where each $\mathbf{x}_i \in \mathbb{R}^d$. Given this data, the problem is to approximate the value of the underlying function $f(\mathbf{x})$ at some point $\mathbf{x}_e$, the evaluation point. In the RBF-FD method we find and evaluate the combined RBF and polynomial interpolant based only on the n nodes which are closest to $\mathbf{x}_e$. If there is more than one evaluation point, then we repeat the process, finding a new set of n nearest neighbors (stencil) for the new evaluation point. In other words, the RBF-FD method for interpolation and approximation of derivatives is just the repeated application of the global RBF method on small subsets of the nodes. However, shifting from the global method to RBF-FD changes the mechanism for convergence. In the global RBF case the differentiability of the RBF is crucial, but we show below that the level of polynomial support is the most important aspect of the present RBF-FD method, because it determines the convergence rate.

4.1 Stagnation Error

In purely RBF-based local methods with an infinitely differentiable RBF like the GA or MQ, when the node-spacing is decreased it is common practice to increase the shape parameter ε by the same factor, thus keeping the condition numbers of the collocation matrices roughly constant. However, with no polynomial support, there comes a point where decreasing the node-spacing (and consequently increasing the shape parameter) no longer yields a more accurate result. Although there are stable algorithms that have been developed for small ε to avoid this issue [16, 15, 14], we

find that perhaps the simplest way to avoid the problem is to augment the local RBF functions with polynomials. After this simple adjustment, each local interpolant is a linear combination of both RBFs and polynomials, so the interpolant is capable of matching the data whether it is under-resolved or extremely well-resolved. In other words, when polynomials are included along with the RBFs, instead of reaching the point of stagnation error, the method simply continues to converge according to the highest degree polynomial that is included.

Although not recommended (since nonsingularity of the collocation matrix has not been shown), PHS RBFs can also be used for local approximations with no polynomial support. In this situation, the mechanism for stagnation error is quite different than for infinitely differentiable RBFs, which was explained above. In Appendix A, the novel explanation of the mechanism for stagnation error in the PHS case is covered in detail.

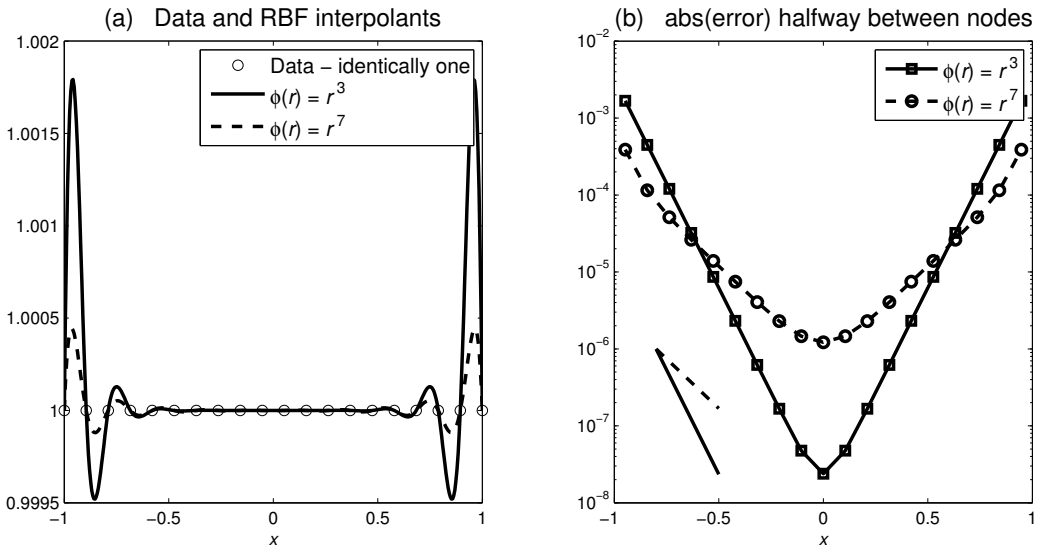

Figure 4.1: (a) The $\phi(r) = r^3$ and $\phi(r) = r^7$ RBF interpolants for 18 nodes of equally spaced unit data. (b) The errors at the halfway points between the nodes. The errors decrease at the theoretical rates as you move towards the center of the interval. See Appendix A for a more complete interpretation of this figure.

4.2 Example Problem: Interpolation and Differentiation

Once a node-set is chosen, there are three parameters left to consider when applying the present RBF-FD method: (1) the highest degree polynomial to include, (2) the RBF, and (3) the stencil-size. In the test problem below, we use two different stencil-sizes ($n = 40$ and $n = 80$) and two different PHS RBFs $\left(\phi(r) = r^3 \text{ and } \phi(r) = r^7\right)$ to approximate a test function and its derivatives with various levels of polynomial support (highest polynomial degrees of 3, 4, 5, and 6). We interpolate, approximate $\partial/\partial x$, and approximate $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ on the nodes shown in Figure 4.2, taking the worst error over all of the stencils. The exact function and its derivatives are

$$\begin{aligned} f =& 1 + \sin(4x) + \cos(3x) + \sin(2y), \\ \frac{\partial f}{\partial x} =& 4\cos(4x) - 3\sin(3x), \\ \Delta f =& -9\cos(3x) - 16\sin(4x) - 4\sin(2y). \end{aligned} \tag{4.1}$$

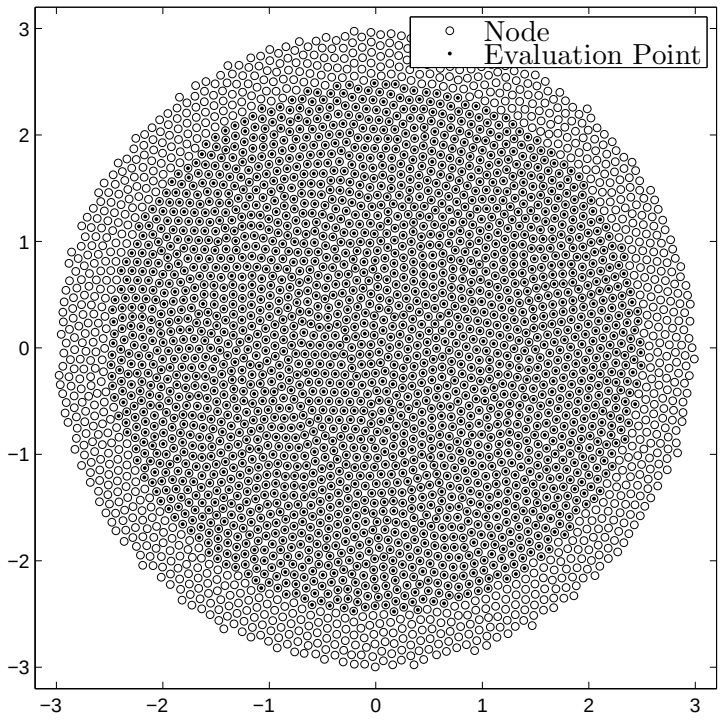

Figure 4.2: Set of scattered nodes and evaluation points for $h = 0.1$. For this test problem, the nodes overlap the evaluation points to eliminate boundary effects.

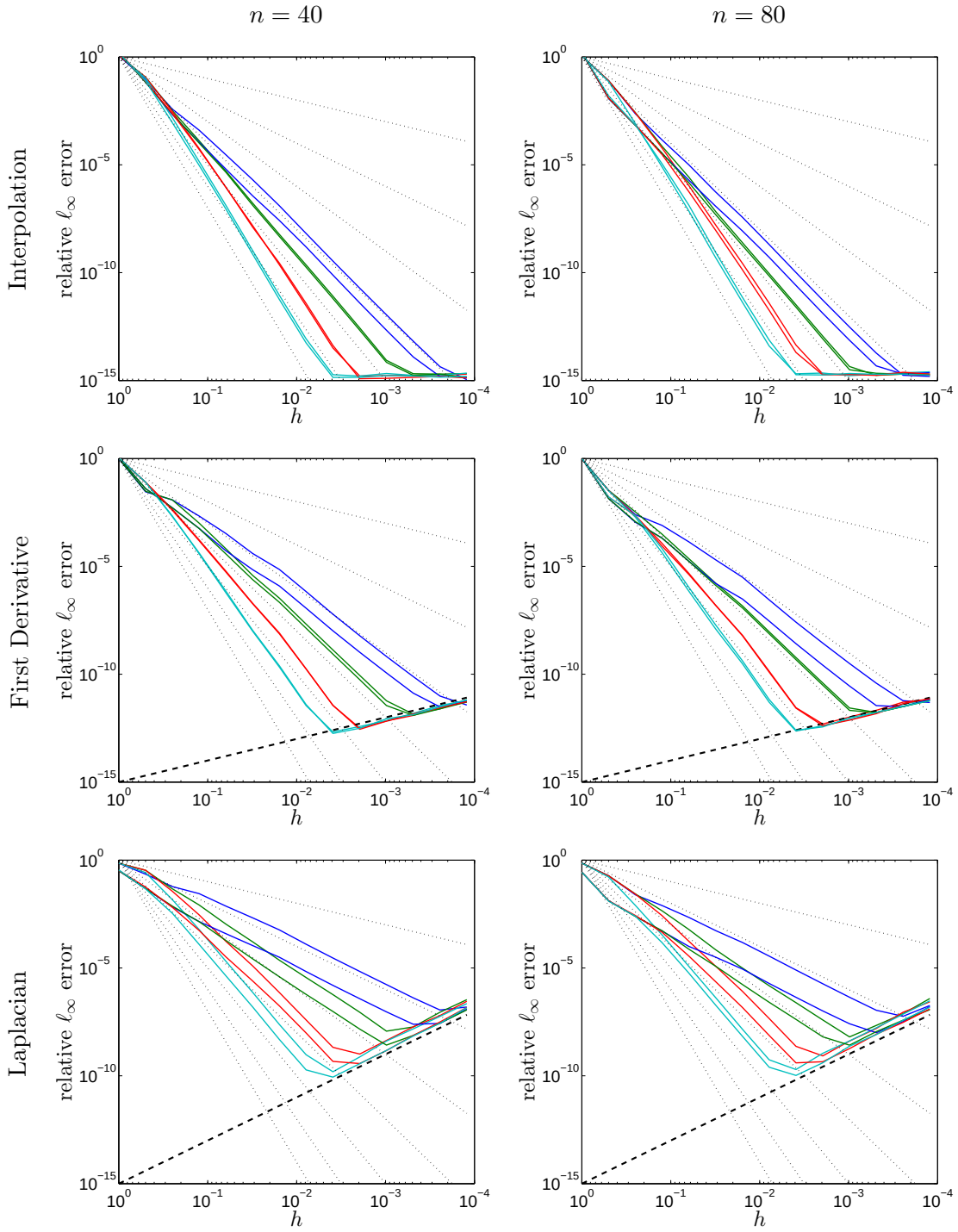

Figure 4.3: Convergence results for interpolation, $\frac{\partial}{\partial x}$, and $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ on the two stencil-sizes $n = 40$ and $n = 80$. The colors blue, green, red, and teal correspond to highest polynomial degrees of 3, 4, 5, and 6, respectively. A pair of lines of the same color indicate errors calculated using $\phi(r) = r^3$ and $\phi(r) = r^7$, with r^7 giving slightly less error. Dotted lines have slopes 1-7.

Looking at the results in Figure 4.3, we see that it is the polynomial support that determines the order of convergence to the true solution. Adjusting the stencil-size or changing the exponent in the RBF might shift the error curve up or down, but it doesn't change the slope. Also, as might have been expected, the same level of polynomial support gives an improved (by one) rate of convergence for interpolation than for approximating the first derivative, and likewise the first derivative approximation converges faster (also by one) than the Laplacian approximation. In Figure 4.3, the dashed lines show where the truncation error meets the round-off error in standard double-precision arithmetic. The dashed lines are error= $10^{-15}/h$ for the first derivative, and error= $10^{-15}/h^2$ for the Laplacian.

4.3 PHS vs. GA RBFs with Polynomial Support

Next, we explore the parameter space in more detail, comparing PHS and GA interpolation for various values of the exponent m in $\phi(r) = r^m$ and the shape parameter ε in $\phi(r) = e^{-(\varepsilon r)^2}$. In the contour plots below (Figures 4.6 and 4.7), we compare the two types of RBFs for interpolation of the same test function f from (4.1). In this case however, we use a scattered 56-node stencil to allow for higher degree polynomials, and we compute errors for three different values of the stencil radius R (Figure 4.4). As R decreases, the test function f in (4.1) looks more and more like a low-order polynomial (Figure 4.5), which is exactly what happens in RBF-FD calculations when the total number of nodes is increased while the stencil-size remains fixed. The goal of the numerical test is to compare the mechanisms for convergence using the two different types of RBF (PHS and GA), and to determine the role that polynomials play in each case. In the experiment that follows, we record the worst interpolation error over $2,000$ scattered evaluation points near the stencil center.

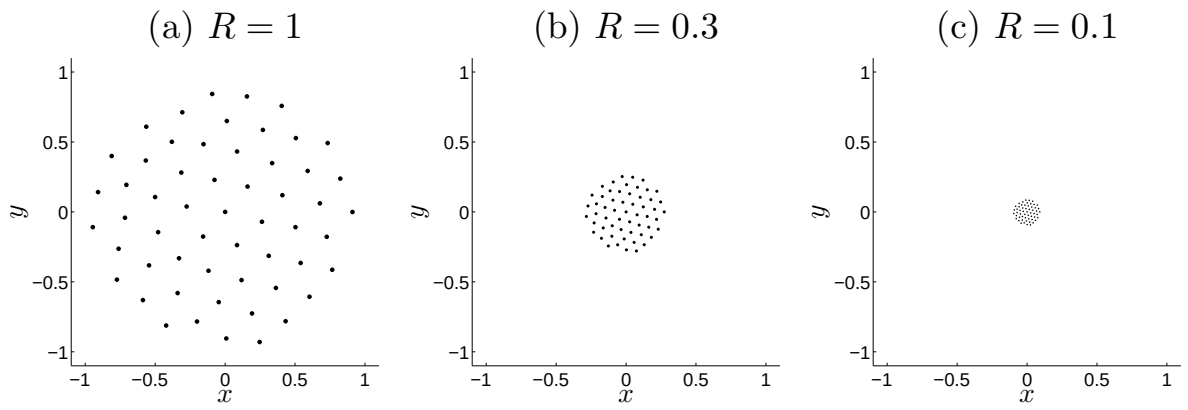

Figure 4.4: 56 Nodes (RBF-centers) for three different values of the radius R.

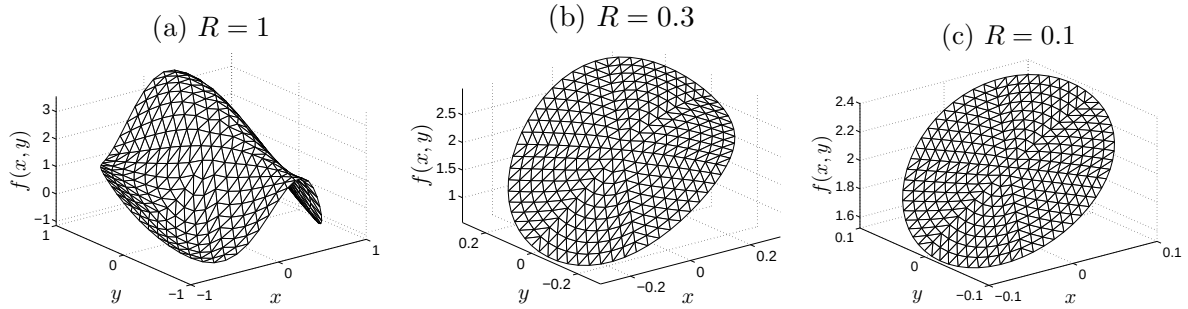

Figure 4.5: Test function (4.1) displayed for $x^2 + y^2 \leq R^2$.

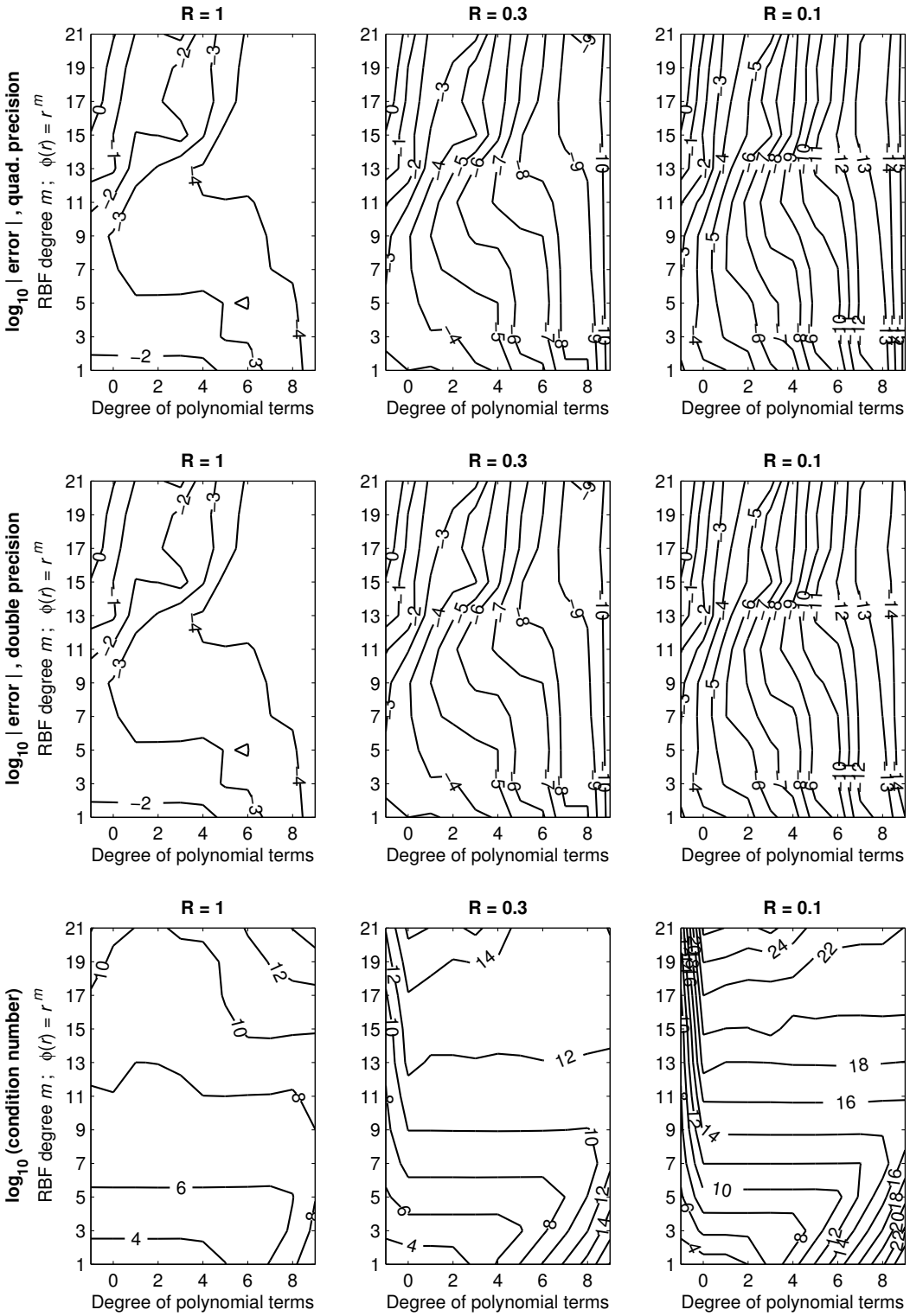

Figure 4.6: The top two rows show $\log_{10}|\text{error}|$ as a function of highest degree polynomial included (x-axis) and exponent in the PHS RBF (y-axis). The bottom row shows $\log_{10}(\text{condition number})$ as a function of the same two parameters.

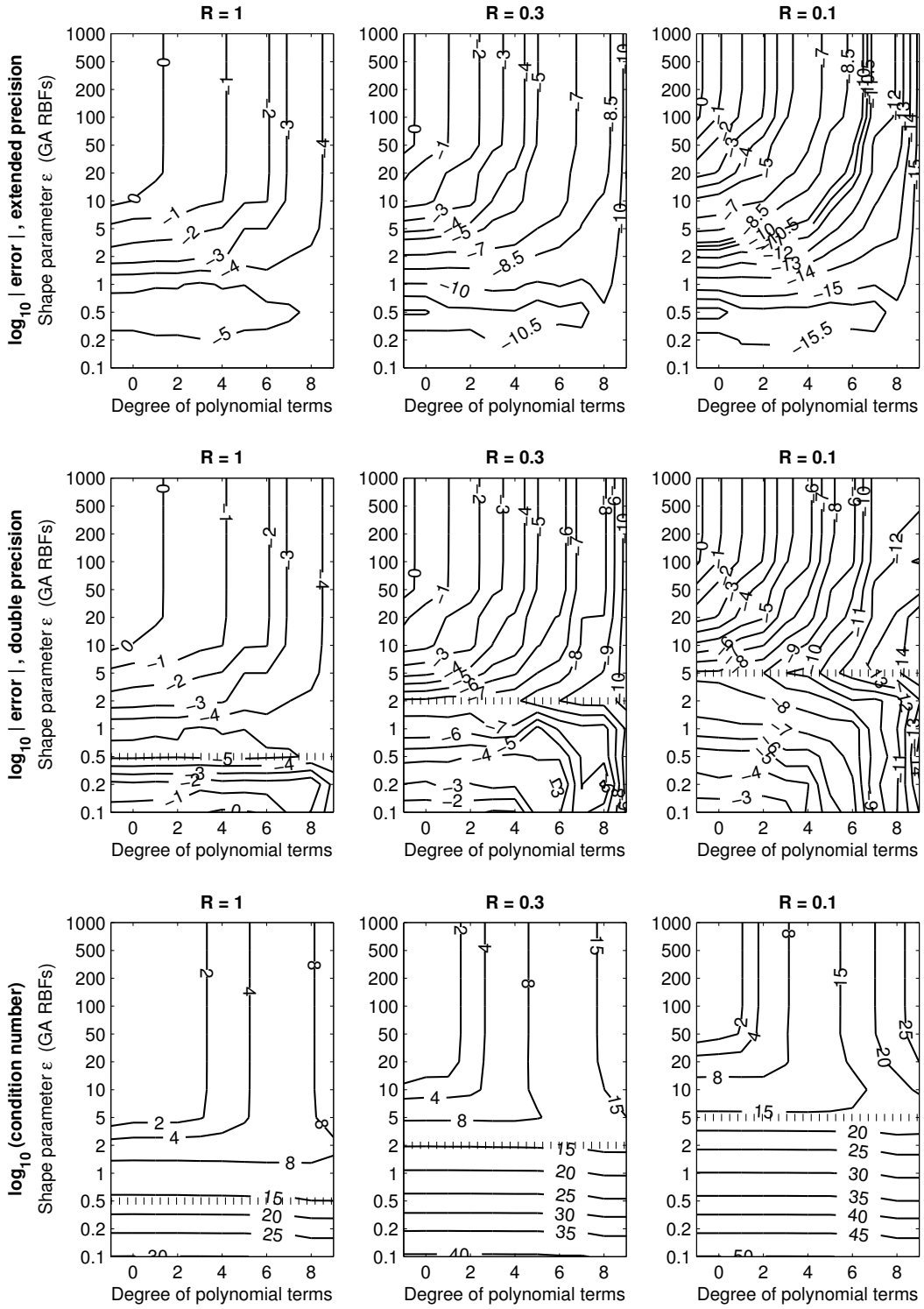

Figure 4.7: The top two rows show $\log_{10}|\text{error}|$ as a function of highest degree polynomial included (x-axis) and shape parameter (y-axis). The bottom row shows $\log_{10}(\text{condition number})$ as a function of the same two parameters.

Figures 4.6 and 4.7 convey a lot of information regarding local RBF interpolation with polynomial augmentation. Looking at Figure 4.6, the first thing to notice is that the top two rows of subplots are identical, even though there are very large condition numbers that appear in the bottom row, especially for $R = 0.1$. This suggests that although the condition number of the PHS and polynomial collocation matrix increases as the nodes get closer together, this increase has no effect on the accuracy of the approximation. Indeed, the condition number can be made to stay constant as R decreases using a re-scaling of the polynomials [20], but there is no practical need for this re-scaling. Secondly, the nearly vertical contour lines in the top two rows confirm what we already hinted at in Figure 4.3, namely that the degree of polynomial support is the most important parameter, especially as the nodes get closer together (the polynomials determine the convergence rate).

In Figure 4.7, the top two rows of subplots are vastly different below the dashed lines (which mark where the condition number is 10^{15}), showing that large condition numbers in the GA case should not be ignored. In the regime where double-precision arithmetic can be applied, GA RBFs do not perform better than PHS. Since PHS has the advantage of having no shape parameter to worry about, and is equally simple to apply regardless of how close the nodes become, the plots suggest that PHS with polynomials provides the best combination of simplicity and accuracy.

It should be noted that, since we are using a 56-node stencil and there are $\binom{9+2}{2} = 55$ polynomial basis functions up to degree 9 in $\mathbb{R}^2$, polynomials up to degree 9 is the maximum we should expect to get away with, and here we do. While the simple interpolation problem allows very high degree polynomials, we will see that for more complicated applications the polynomial degree should be substantially more modest. For a more detailed discussion of Figures 4.6 and 4.7, please see Appendix A.

4.4 RBF and Polynomial Interpolation Coefficients

Interpolating on the nodes $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ with RBFs and polynomials gives an interpolant of the form

$$\begin{aligned} s(\mathbf{x}) =& \lambda_1 \phi(\|\mathbf{x}-\mathbf{x}_1\|) + \lambda_2 \phi(\|\mathbf{x}-\mathbf{x}_2\|) + \cdots + \lambda_n \phi(\|\mathbf{x}-\mathbf{x}_n\|) \\ &+ \mu_1 p_1(\mathbf{x}) + \mu_2 p_2(\mathbf{x}) + \cdots + \mu_k p_k(\mathbf{x}), \end{aligned} \tag{4.2}$$

where $\phi(r)$ is an RBF and $\{p_1(\mathbf{x}), p_2(\mathbf{x}), \ldots, p_k(\mathbf{x})\}$ is a basis for polynomials up to some degree. Using the same test function f from (4.1), we conduct a numerical experiment to determine the relative importance of each piece of the interpolant (4.2) under refinement.

Using the 56 nodes shown in Figure 4.4 for various values of the radius R, we interpolate using RBFs and a simple basis for polynomials up to degree 4 (see Table 3.1). The coefficients are separated according to the type of function they correspond to. The RBF coefficients are $\lambda_1, \lambda_2, \ldots, \lambda_{56}$, the degree 0 coefficient is μ_1, the degree 1 coefficients are μ_2, μ_3, the degree 2 coefficients are μ_4, μ_5, μ_6, and so on (see (4.3) for a better look at the form of the combined PHS and polynomial interpolant $s(x, y)$ in $\mathbb{R}^2$). The point is to see how important a role these different parts of the interpolant play as the value of R decreases (as the total number of nodes increases while the stencil-size remains fixed). To this end, we keep track of the 2-norm of each vector of coefficients for smaller and smaller values of the radius R, and we also plot the interpolation error (see Figures 4.8 and 4.9).

$$\begin{aligned} s(x,y) =& \lambda_1 \phi(\|(x-x_1, y-y_1)\|) + \cdots + \lambda_n \phi(\|(x-x_n, y-y_n)\|) \\ &+ \mu_1 \\ &+ \mu_2 x + \mu_3 y \\ &+ \mu_4 x^2 + \mu_5 xy + \mu_6 y^2 \\ &+ \cdots \end{aligned} \tag{4.3}$$

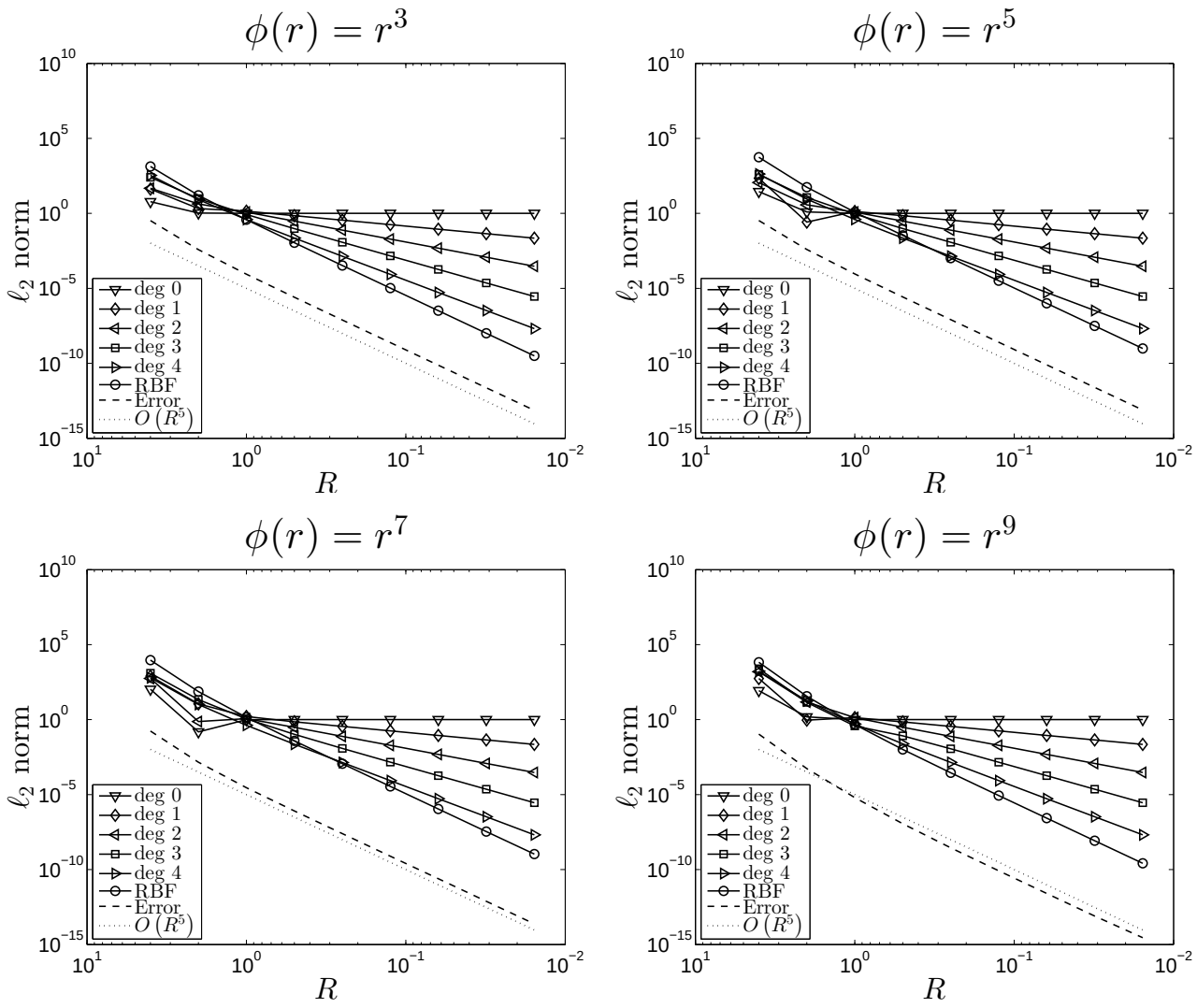

Figure 4.8: Behavior of the PHS and polynomial coefficients under refinement, where the latter have been separated according to polynomial degree. For the RBFs r^3, r^5, r^7, and r^9, the condition numbers of the collocation matrix are 3×10^4, 5×10^5, 5×10^6, and 3×10^7, respectively. In every case, the RBF coefficients converge to zero at the same rate as the interpolation error. While the error is slightly different in each subplot, the results confirm that adjusting the exponent of the PHS RBF might shift the error curve up or down, but does not change its slope.

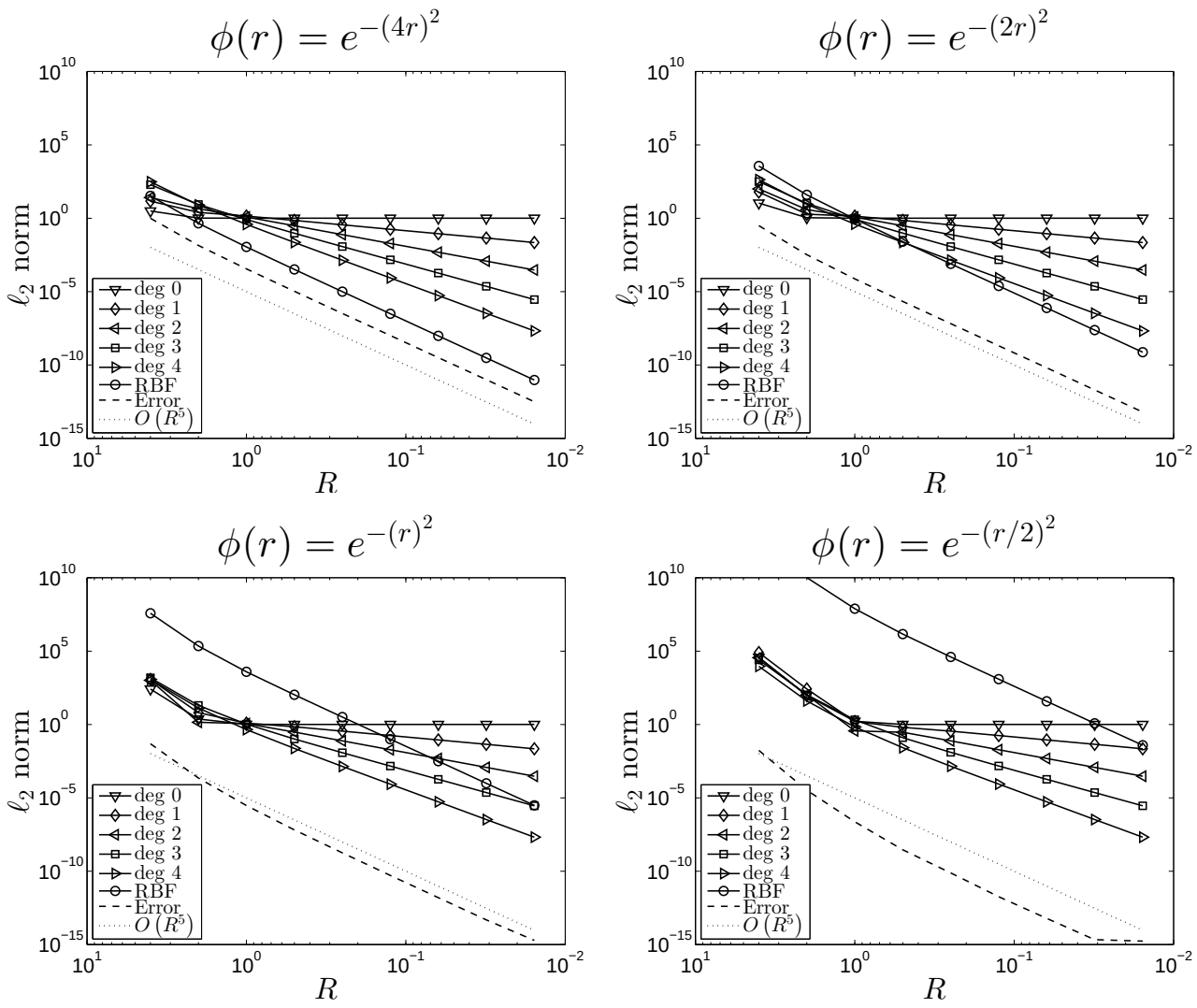

Figure 4.9: Behavior of the GA and polynomial coefficients under refinement, where the latter have been separated according to polynomial degree. For the shape parameters 4, 2, 1, and 1/2, the condition numbers of the collocation matrix are 7×10^2, 9×10^4, 2×10^{10}, and 1×10^{16}, respectively. Here, the differences between the subplots is much more extreme than in the PHS case. While it remains true that the RBF coefficients converge at the same rate as the error, small values of ε yield a combined interpolant that is still heavily influenced by the RBFs, even at high resolution.

4.5 RBF-FD vs. Polynomial Least Squares

Since the convergence rate appears to depend only on the degree of the polynomials, it is natural to wonder if the RBFs can be disregarded altogether. Instead of finding the differentiation weights based on the combined PHS and polynomial interpolant, one might achieve the same accuracy and stability (in conjunction with time-stepping) by finding weights based solely on the polynomial least-squares approximation. In other words, using the matrix P by itself still gives differentiation weights which are exact for polynomials up to some degree, and it is computationally less expensive to work with P rather than the augmented PHS and polynomial matrix $\begin{bmatrix} A & P \\ P^T & O \end{bmatrix}$.

The test problem is again the approximation of the function f from (4.1). However, in this case we will use three different stencil sizes, $n = 10$, 28, and 55 (see Figure 4.10).

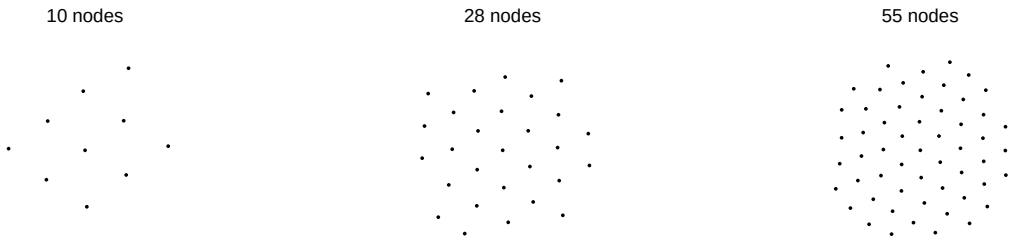

Figure 4.10: The three different stencils used for approximation of f in (4.1). In all cases, the error is calculated over evaluation points near the stencil center.

The function values near the stencil center will be approximated first using the PHS and polynomial method, and then with strictly polynomial least squares. The errors will be compared to discover (i) if the methods give the same order of convergence, and (ii) if there are any other similarities or differences between the two methods.

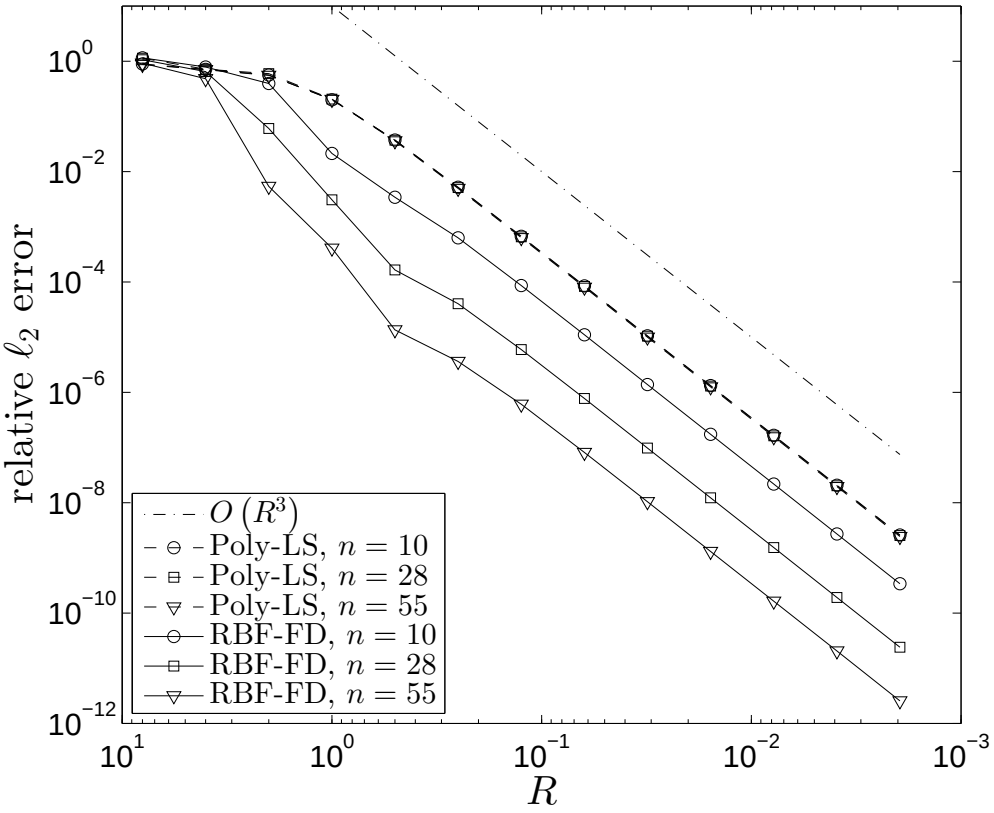

Figure 4.11: Convergence comparison for approximation of the test function in (4.1). In all cases, polynomials up to degree 2 are included, and in the RBF-FD cases we use $\phi(r) = r^5$. Including RBFs does not change the order of accuracy, but improves the error for each value of R. Note that increasing the stencil-size has virtually no effect on the polynomial least squares error (the error curves are indistinguishable).

4.5.1 Eigenvalues of a Differentiation Matrix

Since both RBF-FD and polynomial least squares are applicable for the approximation of a function, with RBF-FD achieving better error overall but converging at the same rate, we move on to test the suitability of the two methods for explicit time-stepping.

Let D be the unit disk in $\mathbb{R}^2$, and consider the following eigenvalue problem.

$$\begin{aligned} \Delta u =& \lambda u, \qquad (x, y) \in D, \\ u =& 0, \qquad (x, y) \in \partial D. \end{aligned} \tag{4.4}$$

It is well-known that the eigenvalues λ are all negative and real, and the ability of a differentiation matrix (DM) to closely match this result, at least to the extent that no eigenvalues end up in the right half-plane, is crucial for explicit time-stepping. Therefore, to compare the two methods in this context, we will apply each method to construct a DM for approximating the solution to (4.4).

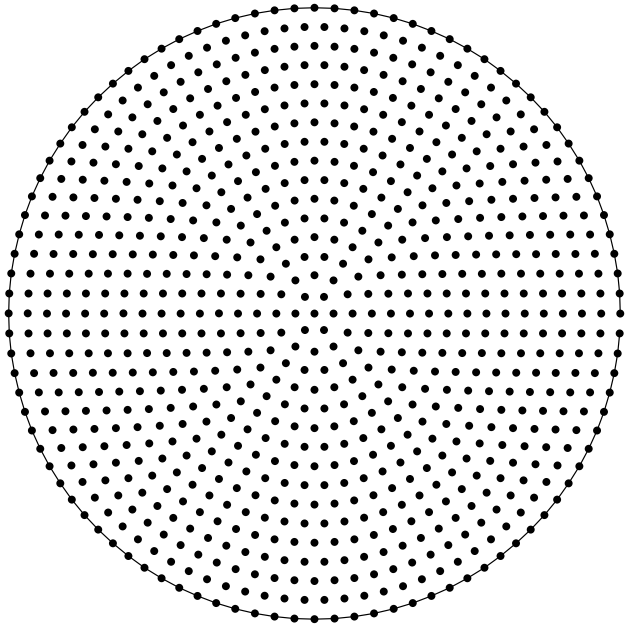

Figure 4.12: A set of 817 nodes on the unit disk. The polynomial least squares method and RBF-FD will be applied on these nodes using various parameter configurations to generate DMs for (4.4).

Looking at Figures 4.13 and 4.14, we conclude that RBF-FD is a superior method overall, achieving better accuracy for approximation and producing a DM with a much more promising distribution of eigenvalues. Even though the two methods converge at the same rate when applied to the interpolation problem, the polynomial least squares approach is not suitable for solving time-dependent PDEs using an explicit time-stepping method.

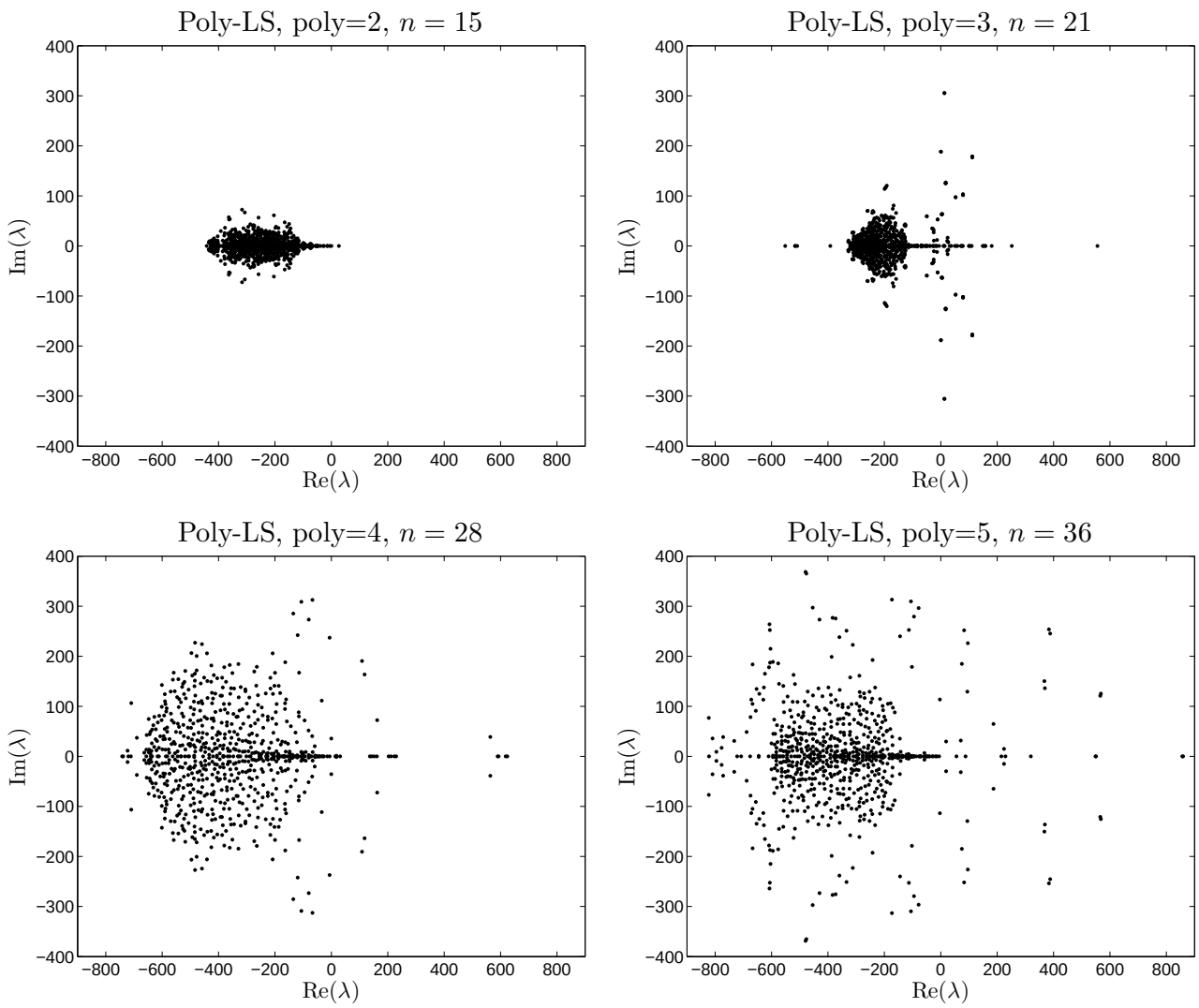

Figure 4.13: Eigenvalues of the Δ DM based on polynomial least squares approximation. In every case there are eigenvalues landing in the right half-plane, with the trend worsening as the degree of the polynomials and the stencil-size are increased. The eigenvalue distributions for the DMs indicate that the method would not be successful for solving a time-dependent PDE in conjunction with an explicit time-stepping algorithm, even with the help of hyperviscosity.

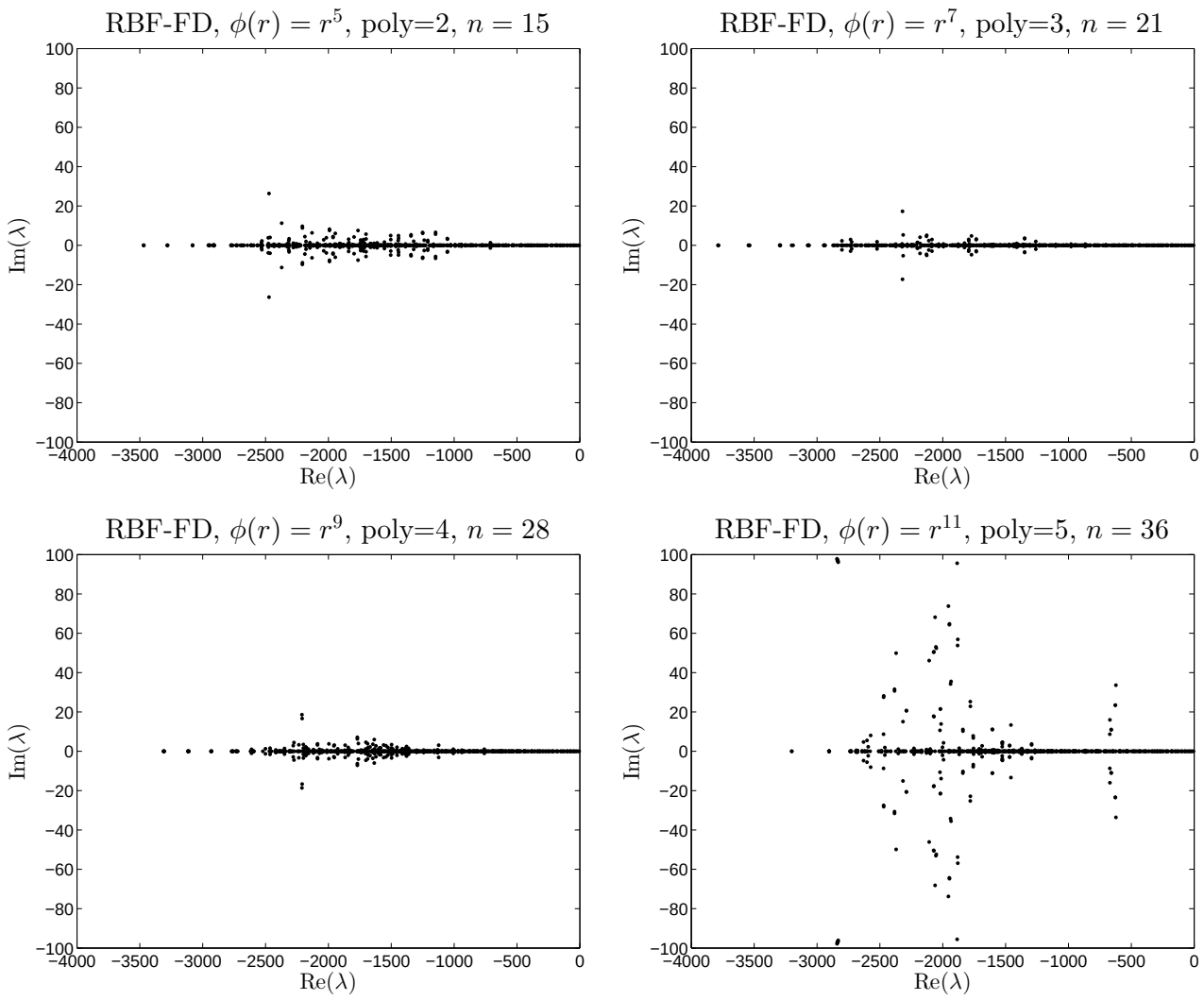

Figure 4.14: Eigenvalues of the Δ DM based on RBF-FD. In every case the eigenvalues are confined to the left half-plane, with many of the smaller ones being purely real and the larger ones straying only slightly from the real axis. The eigenvalues indicate that the combined PHS and polynomial RBF-FD method can be used to solve time-dependent PDEs in a MOL approach. The eigenvalues are not perfect, but they are good enough that stable time-stepping can be achieved by applying hyperviscosity.

Chapter 5

PHS+Polynomials in the RBF-FD Method II: Elliptic PDEs

Having tested PHS and polynomial RBF-FD for local interpolation and derivative approximation, we take the next step and apply the method to solve elliptic PDEs. In this thesis we will cover only the simplest example, namely Poisson's equation, but a more detailed exploration of the present method applied to elliptic PDEs is included in the manuscript in Appendix B. In particular, numerical solutions are found for a nonlinear elliptic PDE using semi-uniformly distributed nodes on an irregular domain. Also, to compare with results in the literature, the well-known lid-driven cavity flow problem is solved on nonuniformly distributed nodes at high Reynolds number.

5.1 Poisson's Equation

For a simple hyperbolic PDE test case, consider Poisson's equation

$$\begin{aligned} \Delta u &= f(x,y), \quad \text{in } \Omega, \\ u &= g(x,y), \quad \text{on } \partial\Omega, \end{aligned} \tag{5.1}$$

on the unit disc $\Omega = \{(x,y)\,,\, x^2+y^2 \leq 1\}$, where $f(x,y)$ and $g(x,y)$ are computed from the simple test function

$$u(x,y) = \sin\left[10\,(x+y)\right]. \tag{5.2}$$

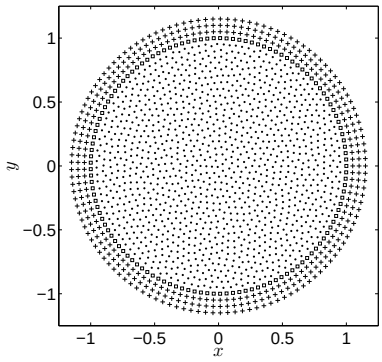

Figure 5.1: Scattered nodes on the unit disc using $h \approx 0.05$. Interior nodes (dot), boundary nodes (square) and exterior nodes (cross). The exterior nodes are treated in a different way depending on the boundary approach.

In order to determine boundary effects for numerical solutions of the Poisson problem, we treat the boundary in three different ways. **Ideal**: Set the function values on the exterior nodes according to the exact solution (5.2), removing boundary effects entirely. **Ghost nodes**: Enforce $\Delta u = f$, $\Delta^2 u = \Delta f$, and $\Delta^3 u = \Delta^2 f$ on the boundary nodes to make use of the exterior nodes. **Boundary**: Perform no special boundary treatment. Looking at the results in Figure 5.2, we see a trend similar to that which was seen for interpolation, with the error depending most prominently on the highest degree polynomial included. Also, as the resolution increases (as h gets smaller), the impact of the boundary treatment diminishes so that, especially for large stencil-sizes, it becomes unnecessary to do anything special at the boundary.

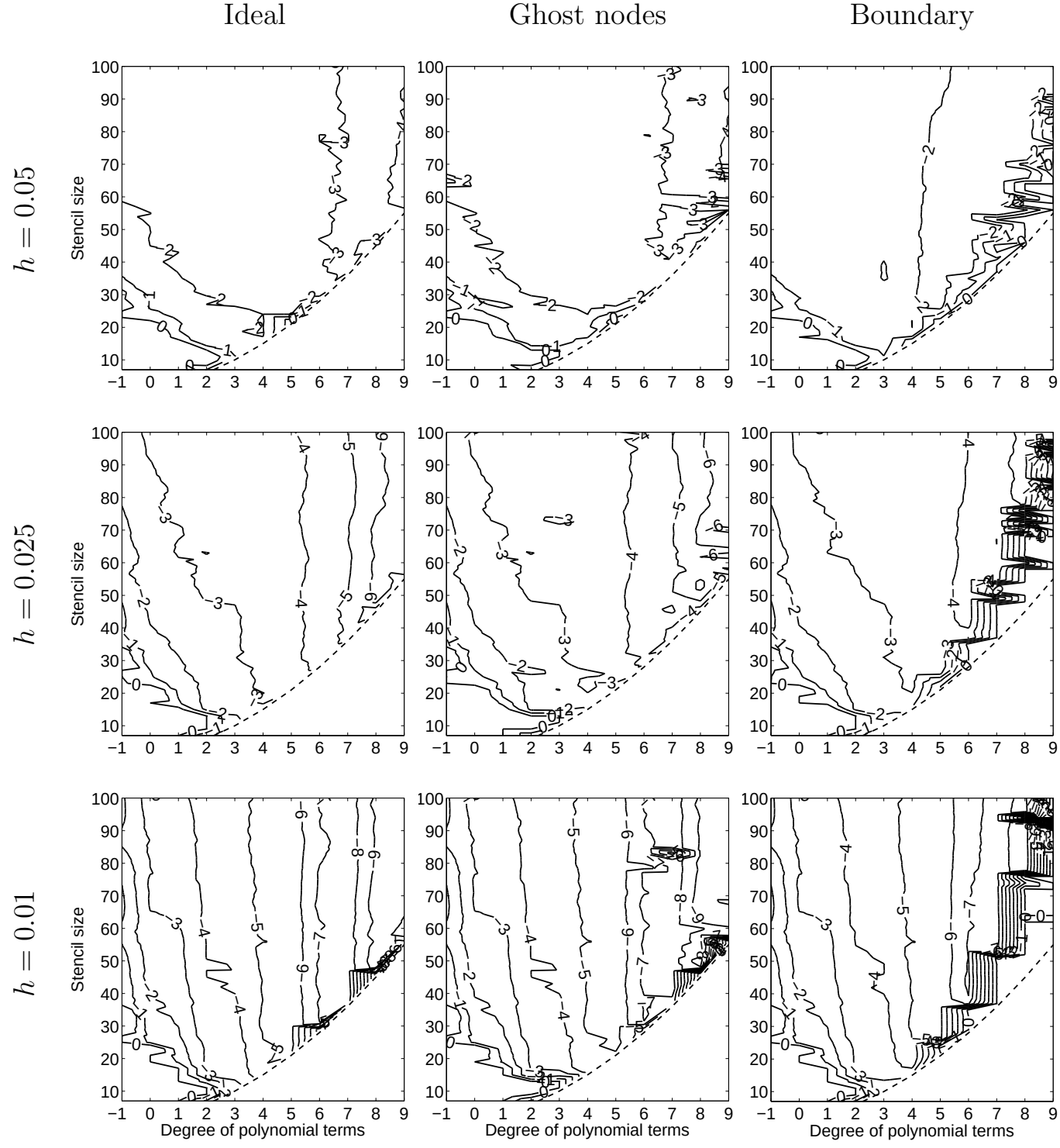

Figure 5.2: Accuracy ($\log_{10}$ error) when using the PHS RBF $\phi(r) = r^7$ with polynomial support to solve the Poisson problem (5.1). The dashed lines show the smallest possible stencil size n for each degree of polynomial term (equal to the number of polynomials in the basis).

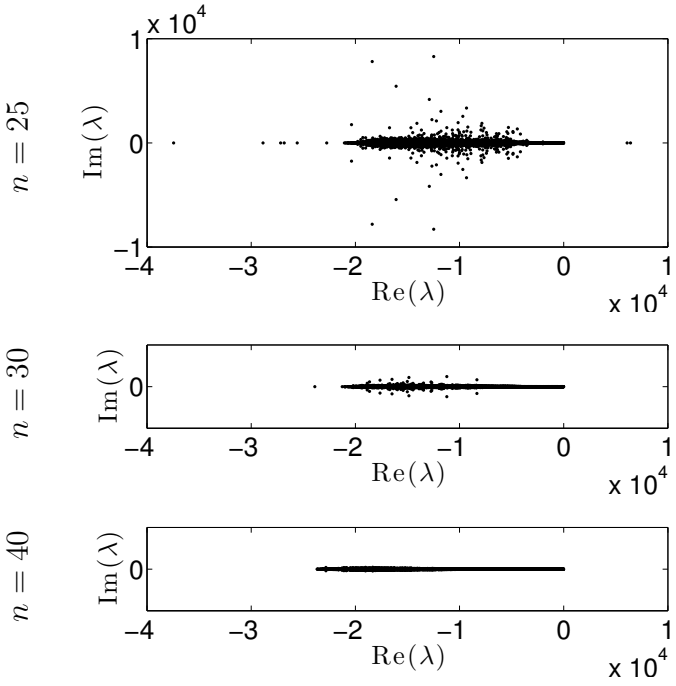

Figure 5.3: Eigenvalue distribution of the differentiation matrix that approximates (5.1) for the boundary case with $h \approx 0.025$, 5th order polynomial terms and $n = 25$, 30 and 40.

The eigenvalues of the DM for three different stencil-sizes are shown in Figure 5.3. Since the true eigenvalues are known to be negative and real, the plots show that as the stencil-size increases up to 40 (about twice as many RBFs as polynomials), the eigenvalues of the DM do a better job of approximating the true eigenvalues. Indeed, with too small of a stencil-size ($n = 25$ in this case), there are even some eigenvalues which land in the right half-plane. This result is reminiscent of the eigenvalue plots in Figure 4.13 from Section 4.5.1, where polynomials were used by themselves in a least squares approach to generate the DM.

Chapter 6

PHS+Polynomials in the RBF-FD Method III: Hyperbolic PDEs

In order to further demonstrate the effectiveness of the RBF-FD method with polynomial support, we solve three hyperbolic time-dependent PDEs using the method of lines. Spatial derivatives are approximated locally with weights derived from polyharmonic splines and polynomials, and the resulting system of ODEs is advanced through time using explicit Runge-Kutta. Hyperbolic example problems are chosen because they are unforgiving, with no natural dissipation to diminish spatial errors over time. Because of this, in order to achieve stable explicit time-stepping, hyperviscosity is included in all of the test cases below (see Section 6.1).

The first example is computationally the easiest, with a simple PDE and no boundary effects, since the domain is a doubly-periodic rectangle. The second case is slightly more difficult, with still a simple PDE (acoustic wave equation) but with a solid Dirichlet boundary condition on the unit disk, which introduces the Runge phenomenon as an additional source of error [9, 17]. Finally, the third hyperbolic PDE is the most challenging, with a governing equation describing scalar advection in a time-varying velocity field, and an initial condition that is only once continuously differentiable.

6.1 Hyperviscosity

Convective PDEs feature little or no natural dissipation, and since numerical methods do not treat every frequency correctly, some sort of filter needs to be included for stable explicit time-stepping. Left untreated, high-frequency errors in the spatial approximation will amplify as the

solution is marched forward in time, contaminating the physically important low-frequency portion of the solution.

Hyperviscosity for the stabilization of RBF-FD calculations was studied extensively in [13], and was shown to be a viable and powerful method for hyperbolic PDEs on a sphere. Although we use PHS RBFs rather than Gaussians to generate the weights, and are on a bounded domain rather than a sphere, the principle is the same: Damp high-frequency noise generated by the approximation, but avoid damaging the physically relevant frequencies which are being treated correctly by the numerical scheme.

Applying hyperviscosity amounts to adding a new term to the right-hand-side of the governing equation. For example, if the system of governing PDEs is

$$\begin{bmatrix} u_t \\ v_t \end{bmatrix} = L \begin{bmatrix} u \\ v \end{bmatrix},$$

where L is some operator which could be either linear or nonlinear, then the modified governing equations with hyperviscosity are

$$\begin{bmatrix} u_t \\ v_t \end{bmatrix} = L \begin{bmatrix} u \\ v \end{bmatrix} + \gamma h^{2K-1} \begin{bmatrix} \Delta^K u \\ \Delta^K v \end{bmatrix},$$

where Δ is the usual Laplacian operator, $K \in \mathbb{N}$ controls which frequencies are most affected, and $\gamma \in \mathbb{R}$ controls the strength of the hyperviscosity. h is the approximate node-spacing, and the factor h^{2K-1} helps minimize adjustments of γ as the resolution changes.

When we derive RBF-FD weights for a particular operator, we are required to apply the operator to the basis functions (see Chapter 2). In the case of PHS RBFs ($\phi(r) = r^m$, where m is odd), the hyperviscosity operator is particularly simple to apply. It turns out that applying the Laplacian to $\phi(r)$ in $\mathbb{R}^d$ yields another PHS RBF:

$$\begin{aligned} \Delta\left(r^m\right) =& \left(\frac{\partial^2}{\partial r^2} + \frac{d-1}{r}\frac{\partial}{\partial r}\right)\left(r^m\right) \\ =& m(m-1)\, r^{m-2} + \frac{d-1}{r} m\, r^{m-1} \\ =& m(m-2+d)\, r^{m-2}. \end{aligned}$$

Here, we apply the usual Laplacian in polar coordinates, but without the θ term (since $\phi(r)$ does not depend on θ). By using this simple formula repeatedly, $\Delta^K\left(r^m\right)$ is easily calculated, and the result is used to generate hyperviscosity weights.

6.2 Example 1: Doubly Periodic transport

The first hyperbolic problem is similar to the solid body rotation test case in [13], except that the domain is a doubly periodic rectangle instead of a sphere, and the initial condition is C^∞ rather than C^1. The governing equation expresses a simple translation of the initial condition in 2-D in a fixed direction. Consider the PDE

$$u_t = -(a,b)\cdot\nabla u = -a\,u_x - b\,u_y, \tag{6.1}$$

on $[-\pi,\pi]\times\left[-\frac{\pi}{2},\frac{\pi}{2}\right]$ with periodic boundary conditions in x and y. Whatever the initial function is, it just moves in the direction of the vector (a,b) with velocity $\sqrt{a^2+b^2}$, so the exact solution is known for all time. We solve the problem numerically with $(a,b)=\left(\frac{\pi}{2},0\right)$ on scattered nodes, using the initial condition (see Figure 6.1)

$$u|_{t=0} = e^{-5\left(x^2+y^2\right)}. \tag{6.2}$$

The time-stepping is carried out using Runge-Kutta-4 with a very small time-step, so that the errors are dominated by the approximation of the spatial derivatives. In order to ensure stable explicit time-stepping, hyperviscosity is added to shift spurious eigenvalues from the right half-plane to the left (see Figure 6.2). From the results in Chapters 4 and 5, we might expect the convergence rate to follow the polynomial degree as it did for interpolation and elliptic PDEs.

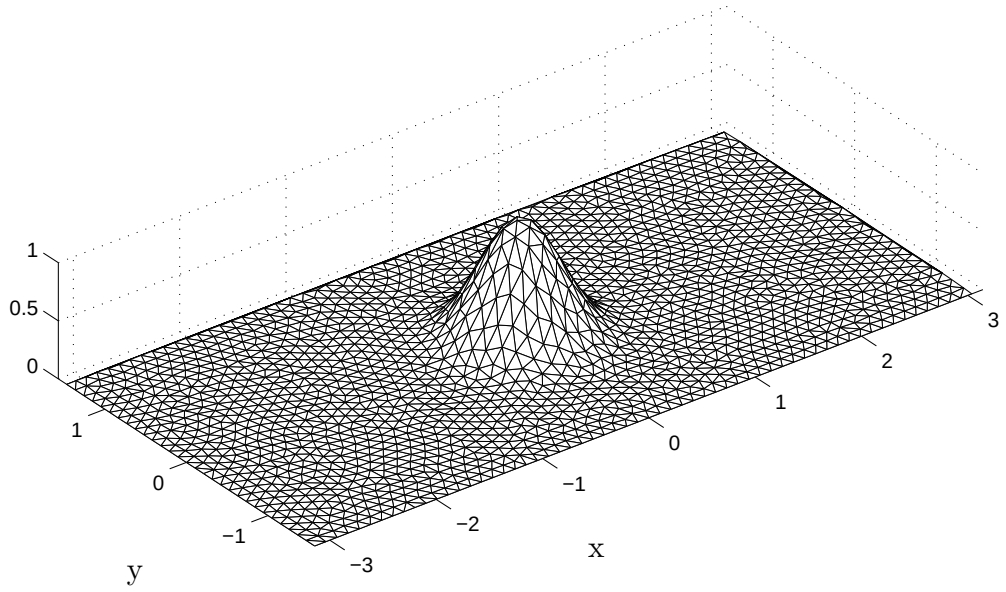

Figure 6.1: The function (6.2) used as the initial condition for the PDE (6.1).

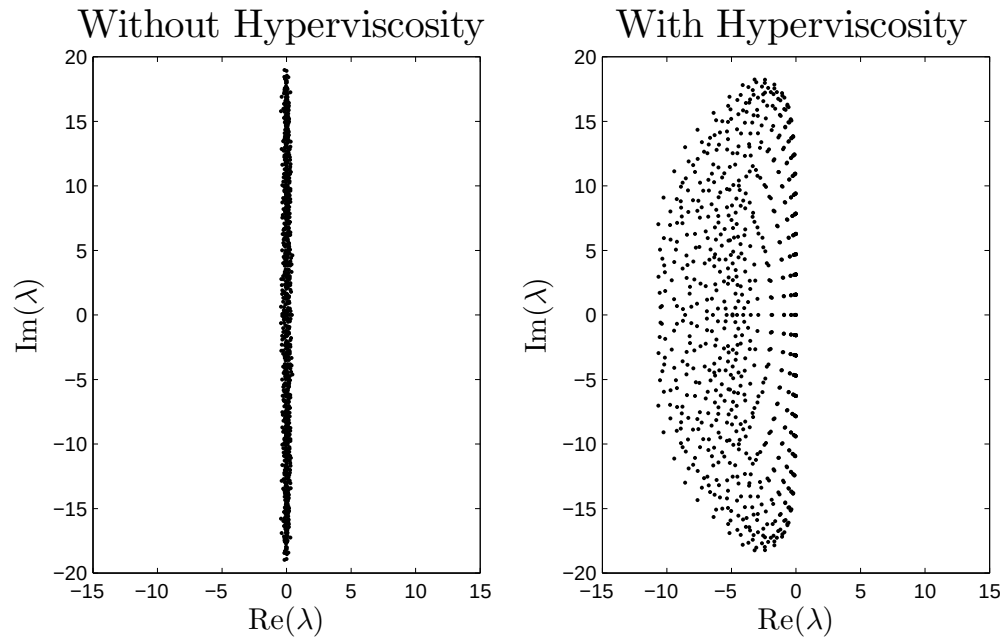

Figure 6.2: Eigenvalues of DM for approximating $\left(-\frac{\pi}{2}\right)\frac{\partial}{\partial x}$ before (left) and after (right) hyperviscosity is added. $\phi(r) = r^7$ with polynomials up to 5^{th} degree on a 28-node stencil using Δ^3-type hyperviscosity were used for approximating both $\partial/\partial x$ and Δ^3 in this case.

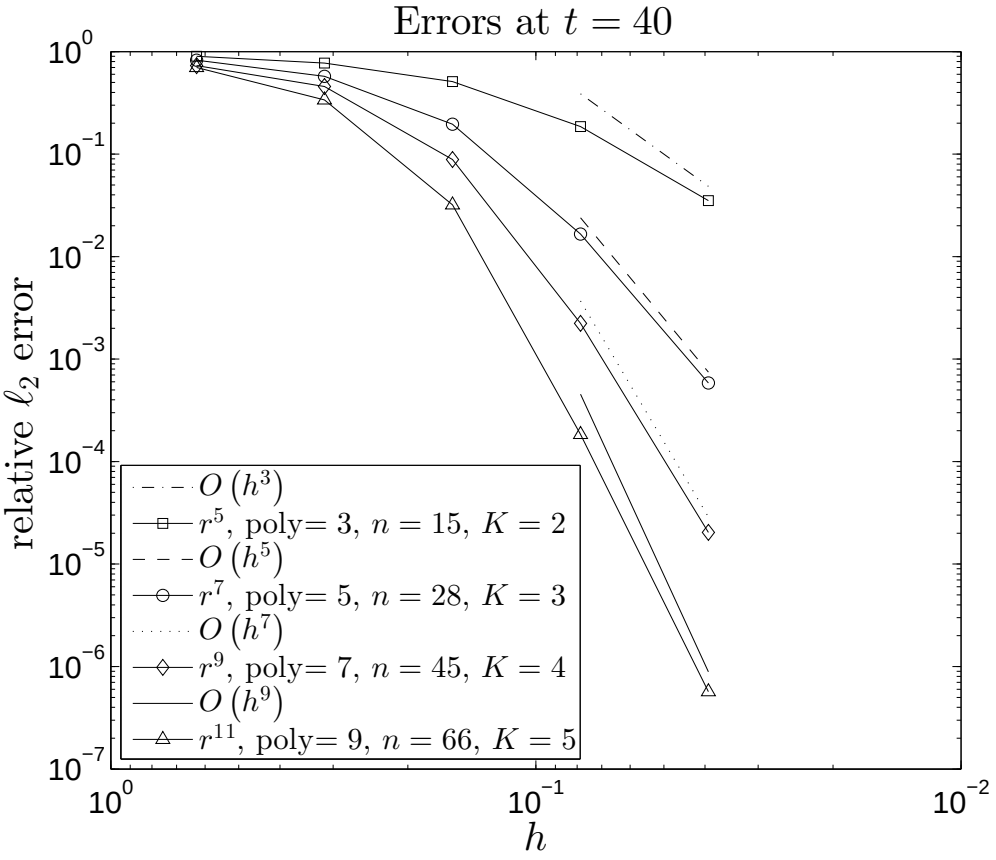

Figure 6.3: Convergence results for various RBF-FD configurations on scattered nodes with approximate node-spacings of $h = \frac{\pi}{5}$, $\frac{\pi}{10}$, $\frac{\pi}{20}$, $\frac{\pi}{40}$, and $\frac{\pi}{80}$. To time-step stably with explicit Runge-Kutta, Δ^K-type hyperviscosity is applied.

Looking at Figure 6.3, we see that the RBF-FD method indeed gives convergence rates which match the highest degree polynomial included in the local basis. Since only first derivatives in space appear in the governing equation (6.1), this result might have been expected. The test shows that in an ideal situation with no boundary effects and a smooth initial condition, the order of convergence to the true solution matches the polynomial degree in the spatial approximation, provided the time-step is small enough.

6.3 Example 2: Acoustic Wave Equation

The acoustic wave equation on the unit disk D is another simple example of a hyperbolic PDE. In this case we will assume a Dirichlet boundary condition, which introduces the Runge phenomenon as an additional source of error near the edge of the domain.

$$\begin{aligned} u_{tt} =& \Delta u, \\ u|_{t=0} =& f(x,y), \\ u_t|_{t=0} =& 0, \\ u|_{(x,y)\in\partial D} =& 0. \end{aligned} \tag{6.3}$$

Although (6.3) is a simple linear IBVP, it is used frequently to model wave motions and the vibrations of a circular membrane. For certain choices of f, the exact solution is known analytically by separation of variables. For example, if α is any positive root of the Bessel function J_ν, then the initial condition

$$f(r,\theta) = J_\nu(\alpha r)\left[\cos(\nu\theta) + \sin(\nu\theta)\right] \tag{6.4}$$

gives the exact solution

$$u(r,\theta,t) = \cos(\alpha t)\, J_\nu(\alpha r)\left[\cos(\nu\theta) + \sin(\nu\theta)\right],$$

in polar coordinates. We use an initial condition that is a linear combination of functions of the form (6.4), so that the exact solution is available for comparison.

The discrete version of the problem looks like

$$\begin{aligned} \begin{bmatrix} \underline{u}' \\ \underline{v}' \end{bmatrix} &= \begin{bmatrix} \gamma h^{2K-1}\Delta^K & I \\ \Delta & \gamma h^{2K-1}\Delta^K \end{bmatrix} \begin{bmatrix} \underline{u} \\ \underline{v} \end{bmatrix}, \\ \begin{bmatrix} \underline{u} \\ \underline{v} \end{bmatrix}_{t=0} &= \begin{bmatrix} \underline{f} \\ \underline{0} \end{bmatrix}, \end{aligned} \tag{6.5}$$

where $\underline{u}$ and $\underline{v}$ are the discrete representations of the functions u and $v = u_t$, respectively, and hyperviscosity is added and included in the block matrix on the RHS.

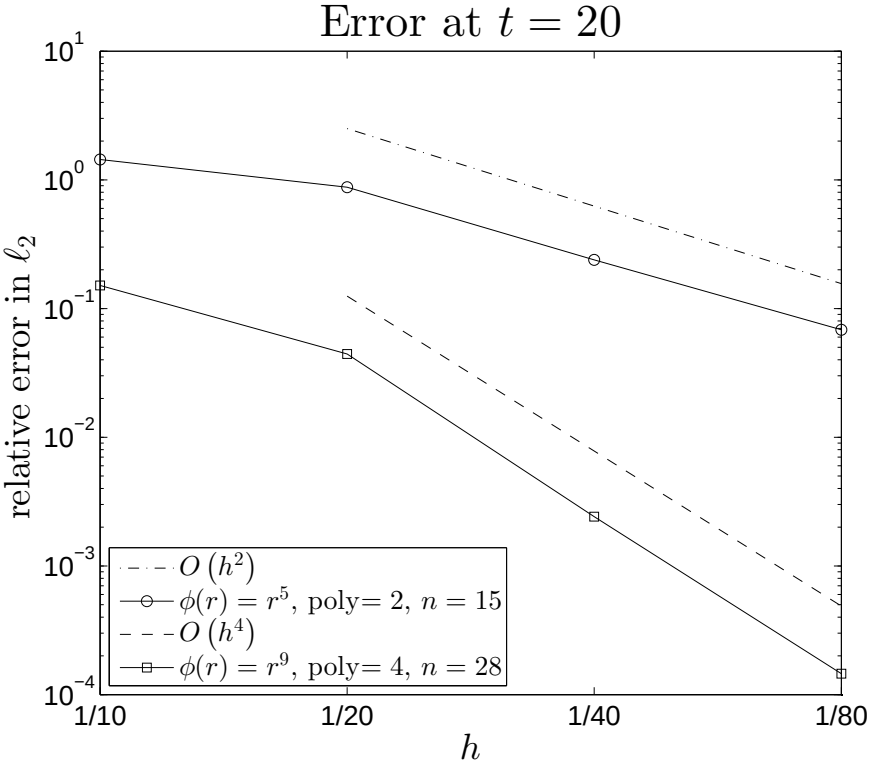

Figure 6.4: Convergence plot for the acoustic wave equation on the unit disk using two different RBF-FD parameter configurations, with the convergence rate again following the highest degree polynomial included. Δ^2-type hyperviscosity is used in the 15-node configuration, while Δ^3 was applied in the 28-node case.

For doubly periodic transport in Section 6.2, there are no boundary effects, and high orders of accuracy can be achieved in space. Unfortunately, getting better than 4^{th} order convergence in the presence of a solid boundary is difficult, because the Runge phenomenon limits the degree of polynomial you can get away with for stencils near the boundary. If the PDE is sufficiently simple, as is the case here, one or more layers of ghost nodes can be placed outside the domain to allow for higher-order spatial approximations. The use of ghost nodes will not be explored further here, but will be discussed in greater detail in Section 7.3.1.

6.4 Example 3: Scalar Advection

This time-dependent PDE test case, taken from [3], is designed to test the ability of a numerical scheme to handle advection. The governing equation describes the advection of the scalar quantity ψ under a time-varying, but given, velocity field:

$$\frac{\partial}{\partial t}(\rho\psi) = -\frac{\partial}{\partial x}(\rho u\psi) - \frac{\partial}{\partial y}(\rho v\psi).$$

The test prescribes a constant density of $\rho \equiv 1$, so

$$\begin{aligned}\frac{\partial \psi}{\partial t} &= -\frac{\partial}{\partial x}(u\psi) - \frac{\partial}{\partial y}(v\psi) \\ &= -u\frac{\partial \psi}{\partial x} - v\frac{\partial \psi}{\partial y} - \psi\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) \\ &= -u\frac{\partial \psi}{\partial x} - v\frac{\partial \psi}{\partial y},\end{aligned} \tag{6.6}$$

where the velocity (u, v) is known (see (6.7)) and satisfies $\nabla \cdot \mathbf{u} = 0$. Define the polar velocity (periodic in time with period T) by

$$u_\theta(r,t) = \frac{4\pi r}{T}\left[1 - \cos\left(\frac{2\pi t}{T}\right)\frac{1-(4r)^6}{1+(4r)^6}\right],$$

with

$$u(x,y,t) = u_\theta(r,t)\sin\theta, \qquad v(x,y,t) = -u_\theta(r,t)\cos\theta. \tag{6.7}$$

Next, define the initial condition for ψ with the C^1 function

$$\psi|_{t=0} = \begin{cases} \left(\frac{1+\cos\pi\tilde{r}}{2}\right)^2, & \tilde{r} \le 1, \\ 0, & \tilde{r} > 1, \end{cases}$$

where $\tilde{r} = 5\sqrt{(x-0.3)^2 + (y-0.5)^2}$. The initial "cosine bell" in ψ is deformed and transported in a circular pattern before returning to its original position. In other words, the exact solution for ψ is periodic in time with period T, so every T units of time we can compare the numerical solution to the exact to measure the error. In order that the problem be well-posed, we enforce boundary conditions of zero scalar gradient at inflow and linear scalar extrapolation at outflow [3]. In all cases below, we set the period $T = 1$ and step forward in time from $t = 0$ to $t = 1$ using explicit RK4.

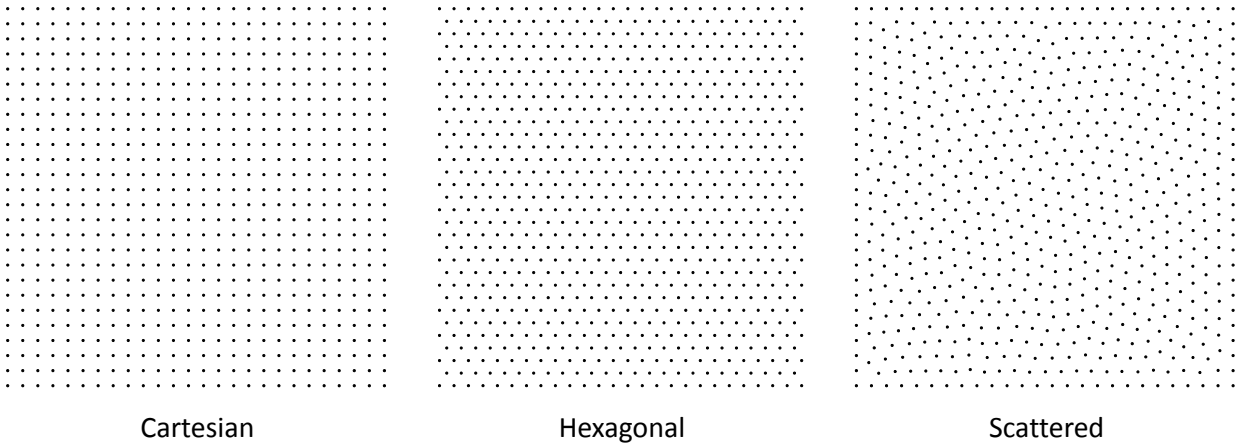

Figure 6.5: The three different types of node-set that RBF-FD will be applied on. Scattered nodes were generated using a local electrostatic repulsion algorithm.

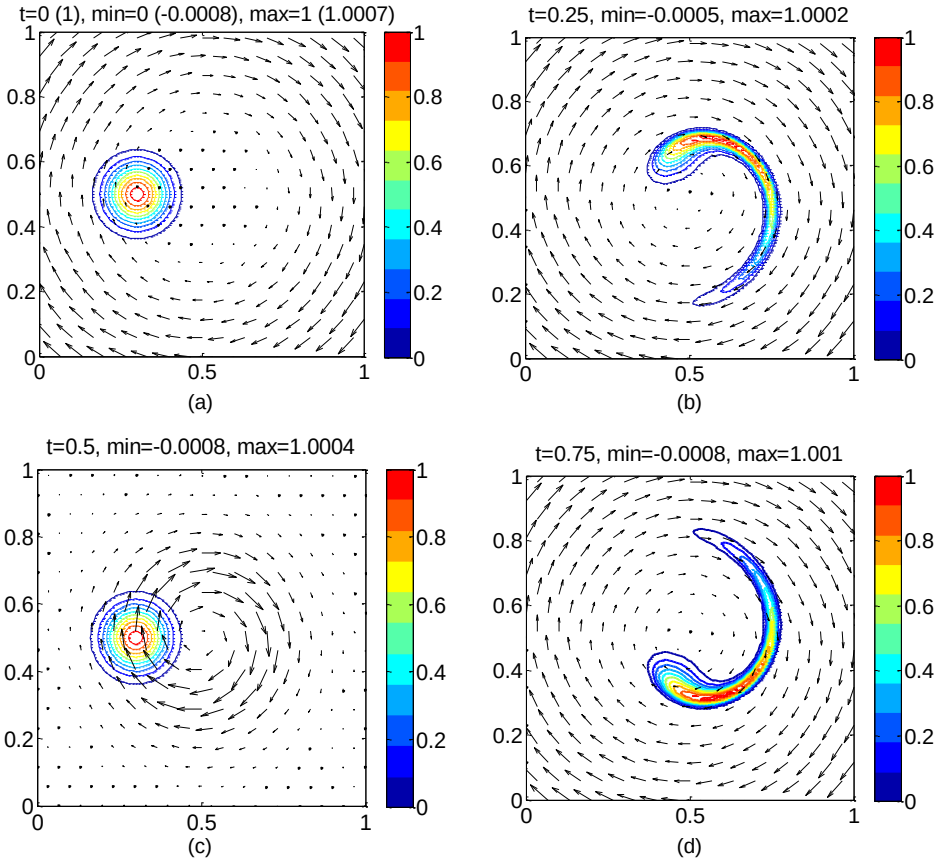

Figure 6.6: The behavior of ψ in time from $t = 0$ to $t = 1$ in a high-resolution RBF-FD solution.

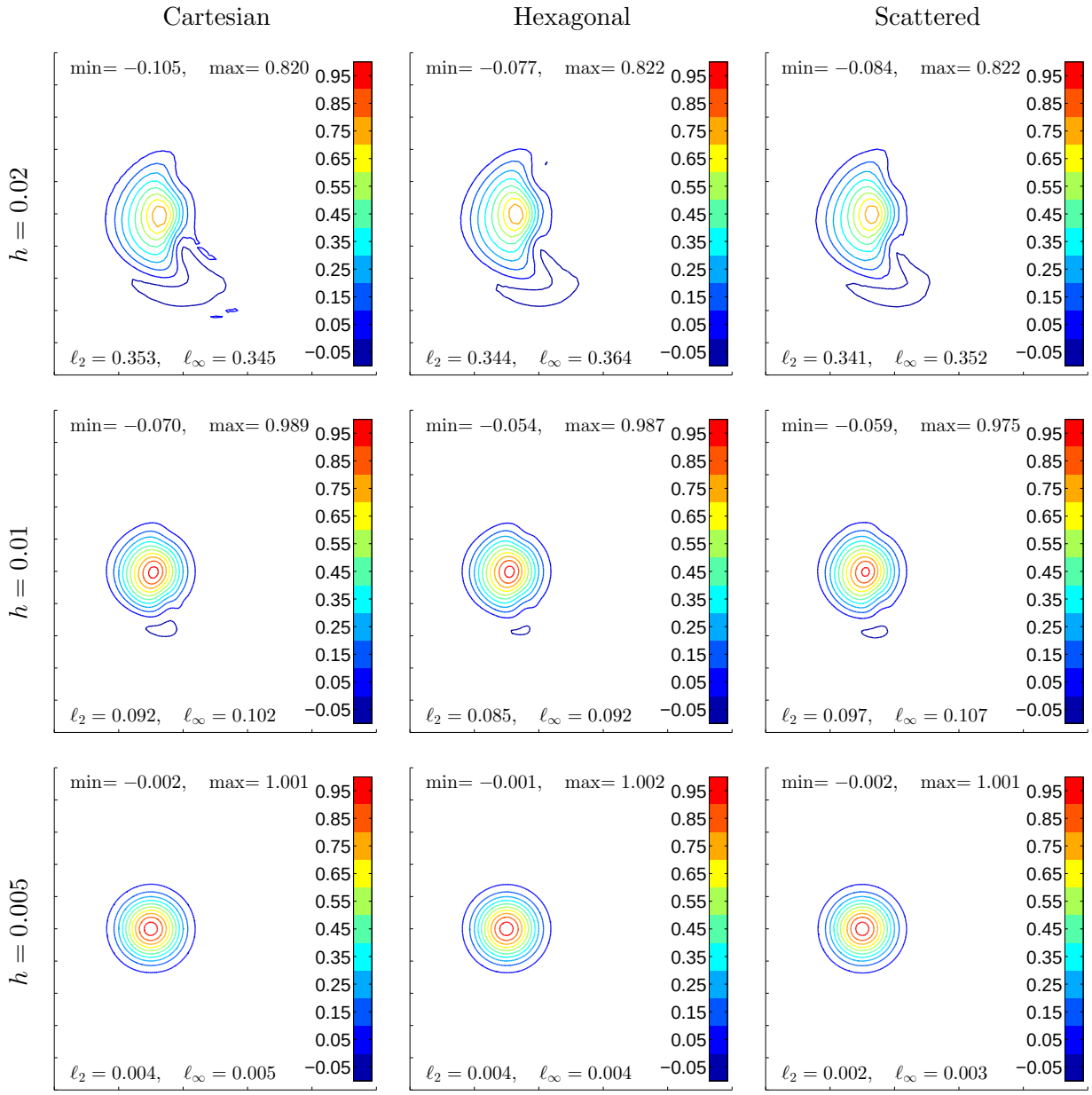

Figure 6.7: Numerical solutions for ψ at $t = T = 1$ using the RBF-FD method on the three different types of node-sets and at three different resolutions. In all cases, we use $\phi(r) = r^3$ with polynomials up to degree 5 on a 37-node stencil, and Δ^3-type hyperviscosity is applied. The configuration is robust enough that it can be applied on each node-set using the same amount of hyperviscosity.

The errors compare favorably with results reported in [3] for various alternative methods. In

order for a more direct comparison, the test problem was solved using fifth order upwind finite-differences with and without adjustments for handling sharp features (WENO), and the convergence results are shown in Figure 6.8.

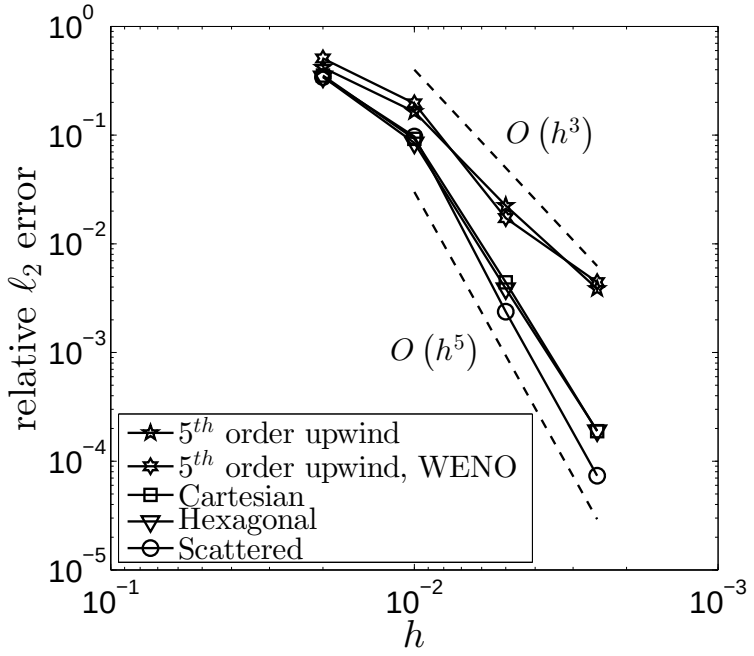

Figure 6.8: Convergence plot of relative ℓ_2 error at $t = 1$. RBF-FD vs. upwind finite differences for the advection test case from [3]. In the three RBF-FD cases, we use $\phi(r) = r^3$ with polynomials up to 5^{th} degree on a 37 node stencil and Δ^3-type hyperviscosity.

In this particular test case, decreasing the exponent in the RBF $\phi(r) = r^m$ leads to more stable solutions in time, especially on scattered nodes. Since changing this parameter has only minor effects on accuracy (see Chapter 4), the overall numerical scheme still converges according to the highest polynomial degree included, in this case 5.

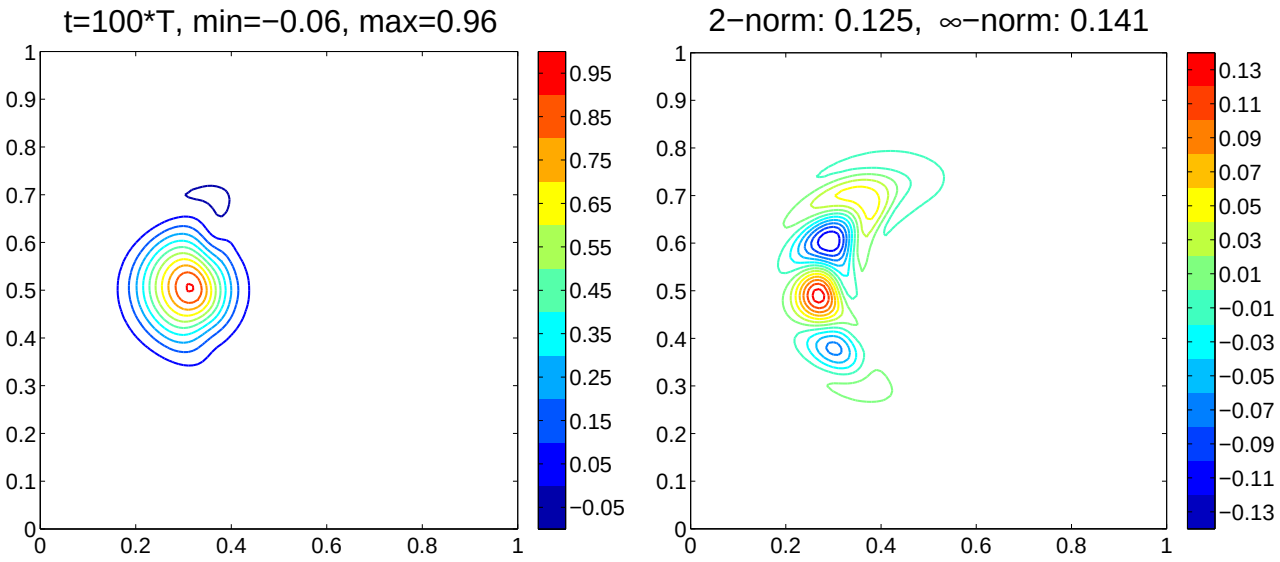

Figure 6.9: Long time ($t = 100$) numerical solution for ψ on hexagonal nodes with approximate node-spacing of $h = .005$. $\phi(r) = r^9$ with polynomials up to 4^{th} degree on a 37-node stencil were used in space.

To test the method for long time integration, we take the hexagonal nodes with a resolution of $h = 0.005$, and step forward in time from $t = 0$ up to $t = 100$. After 100 revolutions, the numerical solution retains its important features, and the min and max values are still close to 0 and 1, respectively. For a more complete discussion of the results of this test case, please see the manuscript in Appendix C, where this simple test is used as a preliminary case before moving on to more complicated problems in NWP.

Chapter 7

RBF-FD for Numerical Weather Prediction

This chapter summarizes the results from the manuscript in Appendix C, where the current RBF-FD method is applied to several test cases in numerical weather prediction (NWP). Please consult Appendix C for a complete discussion of our findings on this topic.

In state-of-the-art models for NWP, the compressible Navier-Stokes equations are coupled with additional equations of motion and parameterizations for precipitation, turbulence, aerosol transport, solar radiation, and more. Each portion of the model is important, but we focus exclusively on resolving the dynamics (compressible Navier-Stokes equations), by applying the method to some test cases popular in the NWP community. These test problems are more challenging than those in Chapter 6, because instead of one simple hyperbolic PDE, the governing equations are a set of four coupled nonlinear PDEs.

In each of the test cases below, the goal is to compare RBF-FD solutions on three different types of node sets, both qualitatively and quantitatively, to determine if the present RBF-FD method is accurate and robust enough for NWP. Some of the test cases are simple enough that convergence to the true solution is feasible. In these situations, the error and convergence behavior of the RBF-FD method are plotted and discussed. On the other hand, some of the test cases feature a more true-to-life atmosphere (very low viscosity), and in these cases convergence analysis is pointless. In the more physically relevant simulations, the tests are meant to determine how flexible the method is, and whether it can be applied in chaotic real-world situations to yield a physically reasonable result.

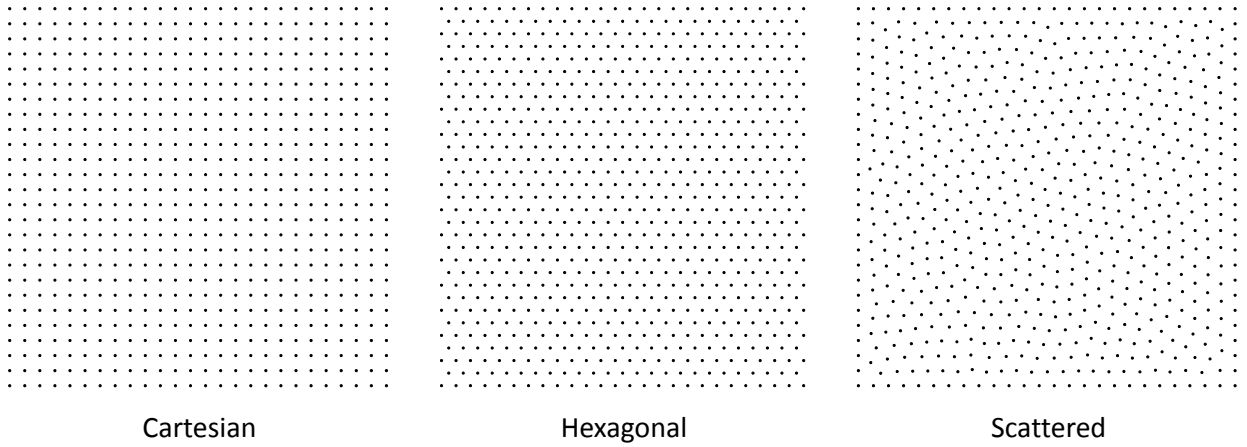

Figure 7.1: Example of what the three different types of node-sets look like on a square domain. In all cases below, in order to make a fair comparison between node-sets, the number of nodes in each set is kept as constant as possible for a given resolution. For example, in the density current case in Section 7.5 below, for 400 meter resolution there are $2,699$ hexagonal nodes, and $2,755$ Cartesian and scattered nodes.

To test if the RBF-FD method is viable for numerical weather prediction, we apply it to two test problems which are common in the literature, both of which require the numerical solution of the same set of coupled governing equations. The first test problem is a rising thermal bubble, which has solid boundaries, but for which the solution never reaches any of these boundaries, so edge effects are greatly reduced. The second test case, known as the density current [29], features a descending cool bubble which does interact with the bottom boundary, thus this test case is computationally more challenging.

7.1 Governing Equations

In the following test problems, the same set of governing equations is solved numerically. In all cases, spatial derivatives are approximated with RBF-FD (using PHS and polynomials) and the remaining system of ODEs is advanced through time using explicit Runge-Kutta. The governing

equations are written

$$\begin{aligned}
\frac{\partial u}{\partial t} =& - u\frac{\partial u}{\partial x} - w\frac{\partial u}{\partial z} - c_p\theta\frac{\partial \pi}{\partial x} + \mu\Delta u, \\
\frac{\partial w}{\partial t} =& - u\frac{\partial w}{\partial x} - w\frac{\partial w}{\partial z} - c_p\theta\frac{\partial \pi}{\partial z} - g + \mu\Delta w, \\
\frac{\partial \theta}{\partial t} =& - u\frac{\partial \theta}{\partial x} - w\frac{\partial \theta}{\partial z} + \mu\Delta\theta, \\
\frac{\partial \pi}{\partial t} =& - u\frac{\partial \pi}{\partial x} - w\frac{\partial \pi}{\partial z} - \frac{R_d}{c_v}\pi\left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z}\right),
\end{aligned} \tag{7.1}$$

where $c_p = 1004$ J/(kg K) and $c_v = 717$ J/(kg K) are the specific heat of air at constant pressure and volume, respectively, $R_d = c_p - c_v = 287$ J/(kg K) is the gas constant for dry air, $g = 9.81$ m/s^2 is the gravitational constant, u is the horizontal velocity, w is the vertical velocity, $\pi = (P/P_0)^{R_d/c_p}$ is the nondimensional Exner pressure $\left(P_0 = 1 \times 10^5 \text{ Pa is the pressure at the surface}\right)$, and $\theta = T/\pi$ is the potential temperature in kelvin. To simplify the description of the initial and boundary conditions, θ and π are split into a static background state plus a time-varying perturbation:

$$\begin{aligned}
\theta(x,z,t) =& \overline{\theta} + \theta'(x,z,t), \\
\pi(x,z,t) =& \overline{\pi}(z) + \pi'(x,z,t).
\end{aligned} \tag{7.2}$$

The four time-varying unknowns in the problem are u, w, θ', and π'. (7.1) is written in terms of these unknowns to arrive at the final form of the governing equations, which will be used for computation:

$$\begin{aligned}
\frac{\partial u}{\partial t} =& - u\frac{\partial u}{\partial x} - w\frac{\partial u}{\partial z} - c_p\left(\overline{\theta} + \theta'\right)\frac{\partial \pi'}{\partial x} + \mu\Delta u, \\
\frac{\partial w}{\partial t} =& - u\frac{\partial w}{\partial x} - w\frac{\partial w}{\partial z} - c_p\left(\overline{\theta} + \theta'\right)\frac{\partial \pi'}{\partial z} + \frac{g\theta'}{\overline{\theta}} + \mu\Delta w, \\
\frac{\partial \theta'}{\partial t} =& - u\frac{\partial \theta'}{\partial x} - w\frac{\partial \theta'}{\partial z} + \mu\Delta\theta', \\
\frac{\partial \pi'}{\partial t} =& - u\frac{\partial \pi'}{\partial x} - w\left(\frac{\partial \overline{\pi}}{\partial z} + \frac{\partial \pi'}{\partial z}\right) - \frac{R_d}{c_v}\left(\overline{\pi} + \pi'\right)\left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z}\right).
\end{aligned} \tag{7.3}$$

7.2 Boundary Conditions

The lateral boundaries are assumed periodic in every case, and the solid top and bottom boundaries ($w = 0$) are assumed to be free-slip and no-flux:

$$\frac{\partial u}{\partial z} = 0, \qquad \frac{\partial \theta'}{\partial z} = 0. \tag{7.4}$$

Enforcing the vertical momentum equation in (7.3) on the top and bottom boundaries yields two additional boundary conditions:

$$\frac{\partial^2 w}{\partial z^2} = 0, \qquad \frac{\partial \pi'}{\partial z} = \frac{g\theta'}{c_p \overline{\theta}\left(\overline{\theta} + \theta'\right)}. \tag{7.5}$$

Including these two extra boundary conditions allows for one layer of ghost nodes in each variable (see Section 7.3.1).

7.3 Numerical Method

Spatial derivatives in (7.3) are approximated using RBF-FD, with $\phi(r) = r^7$ and polynomials up to 4^{th} degree on a 37-node stencil. The remaining system of first order ODEs is advanced through time using the standard 4^{th} order 4-step Runge-Kutta method. While there certainly are more sophisticated and efficient time-stepping methods available for NWP, for example the time-splitting method derived in [30], the focus here is the validation of the RBF-FD approximations for the governing equations (7.3), so time-stepping alternatives are not considered.

7.3.1 Ghost Nodes

Outside of the top and bottom boundaries, one layer of ghost nodes is used for each variable to lessen the effects of nearly one-sided stencils (Runge phenomenon) [9, 17]. The method for placing the ghost nodes is simple. For each top and bottom boundary node, find the distance to the nearest interior neighbor and place one ghost node this same distance away from the boundary node, in the direction perpendicular to the boundary (see Figure 7.2).

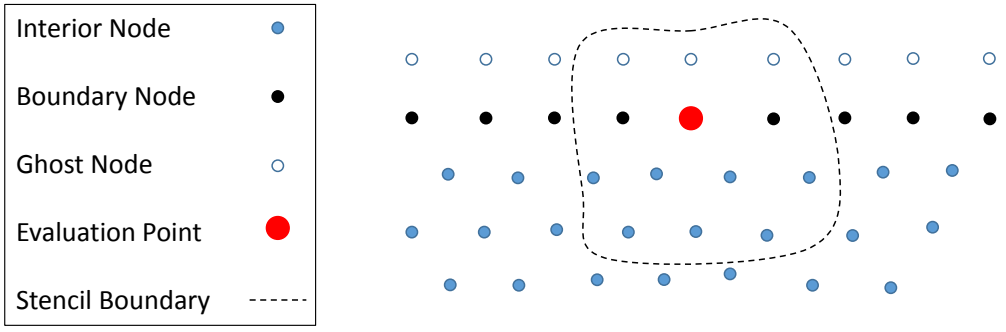

Figure 7.2: Example of a stencil that might be used to enforce a Dirichlet boundary condition on the top boundary.

Each of the four unknown variables (u, w, θ', and π') satisfies a non-Dirichlet boundary condition on the top and bottom boundaries, given in (7.4) and (7.5). The numerical solution for each variable is stepped forward in time on the boundary nodes, and the values at the ghost nodes are solved for at the beginning of each Runge-Kutta substep by enforcing the non-Dirichlet boundary conditions. For example, on each top boundary node, we have the condition $\partial u/\partial z = 0$, and this exact boundary condition is approximated discretely using the differentiation matrix W (generated by the PHS and polynomial RBF-FD method):

$$\begin{aligned} W\underline{u} &= \underline{0}, \\ W_N\underline{u}_N + W_G\underline{u}_G &= \underline{0}, \end{aligned} \tag{7.6}$$

where $\underline{u}_G$ are the ghost nodes and $\underline{u}_N$ are the non-ghost nodes. At each Runge-Kutta substep, the values for the function u at the ghost nodes are approximated by solving the linear system (7.6) for $\underline{u}_G$:

$$\underline{u}_G = -W_G^{-1}W_N\underline{u}_N.$$

Similarly, the values for w, θ', and π' at the ghost nodes are solved for by enforcing $\partial^2 w/\partial z^2 = 0$, $\partial\theta'/\partial z = 0$, and $\partial\pi'/\partial z = g\theta'/\left[c_p\overline{\theta}\left(\overline{\theta}+\theta'\right)\right]$, respectively, at each top and bottom boundary node. Once each variable is known at the ghost nodes, this information is used for approximating the spatial derivatives that appear in the right-hand-side of (7.3).

7.3.2 Hyperviscosity

In order for explicit time-stepping to be stable, hyperviscosity terms are added to the right-hand-side of the system of governing equations (7.3) (See Section 6.1 for a more detailed description and background information regarding hyperviscosity):

$$\begin{aligned}
\frac{\partial u}{\partial t} &= \cdots + \gamma h^{2K-1} \Delta^K u, \\
\frac{\partial w}{\partial t} &= \cdots + \gamma h^{2K-1} \Delta^K w, \\
\frac{\partial \theta'}{\partial t} &= \cdots + \gamma h^{2K-1} \Delta^K \theta', \\
\frac{\partial \pi'}{\partial t} &= \cdots + \gamma h^{2K-1} \Delta^K \pi'.
\end{aligned}$$

In all of the test cases below, $\gamma = 2^{-6}$ controls the strength of the hyperviscosity and $K = 3$ controls which frequencies are most affected. h^{2K-1} is included so that γ needs no adjustment when h changes. The parameter configuration is robust enough that, once the hyperviscosity coefficient γ is chosen, the method can be applied without adjustment on all of the different node-sets.

7.4 Test Case: Rising Bubble

The rising bubble test case has been used in many forms to test the capabilities of various numerical methods for weather prediction [18, 23]. Usually the test case prescribes no natural diffusion (dynamic viscosity $\mu = 0$), but we solve the problem using both high and low values for viscosity, to test the convergence rate and the qualitative behavior, respectively. The governing equations (7.3) are solved numerically, and the dynamics are initiated by a perturbation in the potential temperature. The spatial domain is $[0, 10] \times [0, 10]$ km^2, the time domain is $[0, 1100]$ s, and the hydrostatic background states are defined by

$$\overline{\theta} = 300, \qquad \overline{\pi}(z) = 1 - \frac{g}{c_p \overline{\theta}} z.$$

At $t = 0$, all variables are zero except for the potential temperature perturbation θ', which is initialized as a C^0 bubble with radius $R = 1.5$ km centered at $(x_c, z_c) = (5 \text{ km}, 3 \text{ km})$:

$$\theta'\big|_{t=0} = 2 \max \left\{ 0, 1 - \frac{r(x, z)}{R} \right\},$$

where $r(x,z) = \sqrt{(x-x_c)^2 + (z-z_c)^2}$. We first solve the problem with a nonphysical viscosity of $\mu = 10$ m^2/s, to test the convergence rate of the method. Then, we use a more realistic viscosity of $\mu = 2 \times 10^{-5}$ m^2/s to see if RBF-FD is stable in this regime, and if it gives a qualitatively reasonable result.

7.4.1 Rising Bubble, $\mu = 10$ m^2/s

To get an idea of what the exact solution looks like, Figure 7.3 shows five snapshots in time of a high-resolution RBF-FD solution on Cartesian nodes.

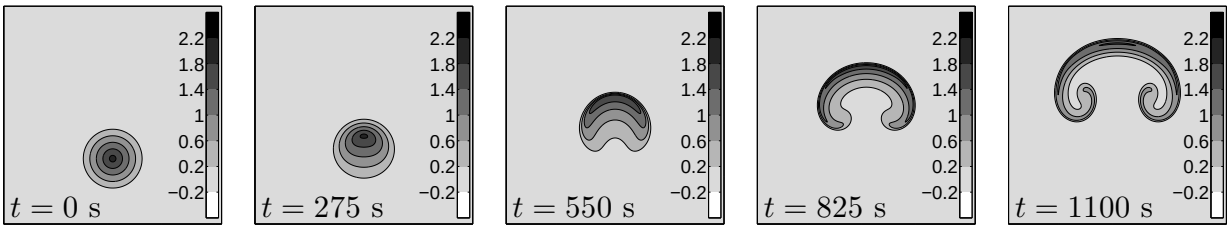

Figure 7.3: Snapshots of the numerical solution for θ' at various times in the high-viscosity rising bubble test case. The viewing window is the full physical domain, $[0,10] \times [0,10]$ km^2, and the contours are from -0.2 to 2.2 by 0.4 K.

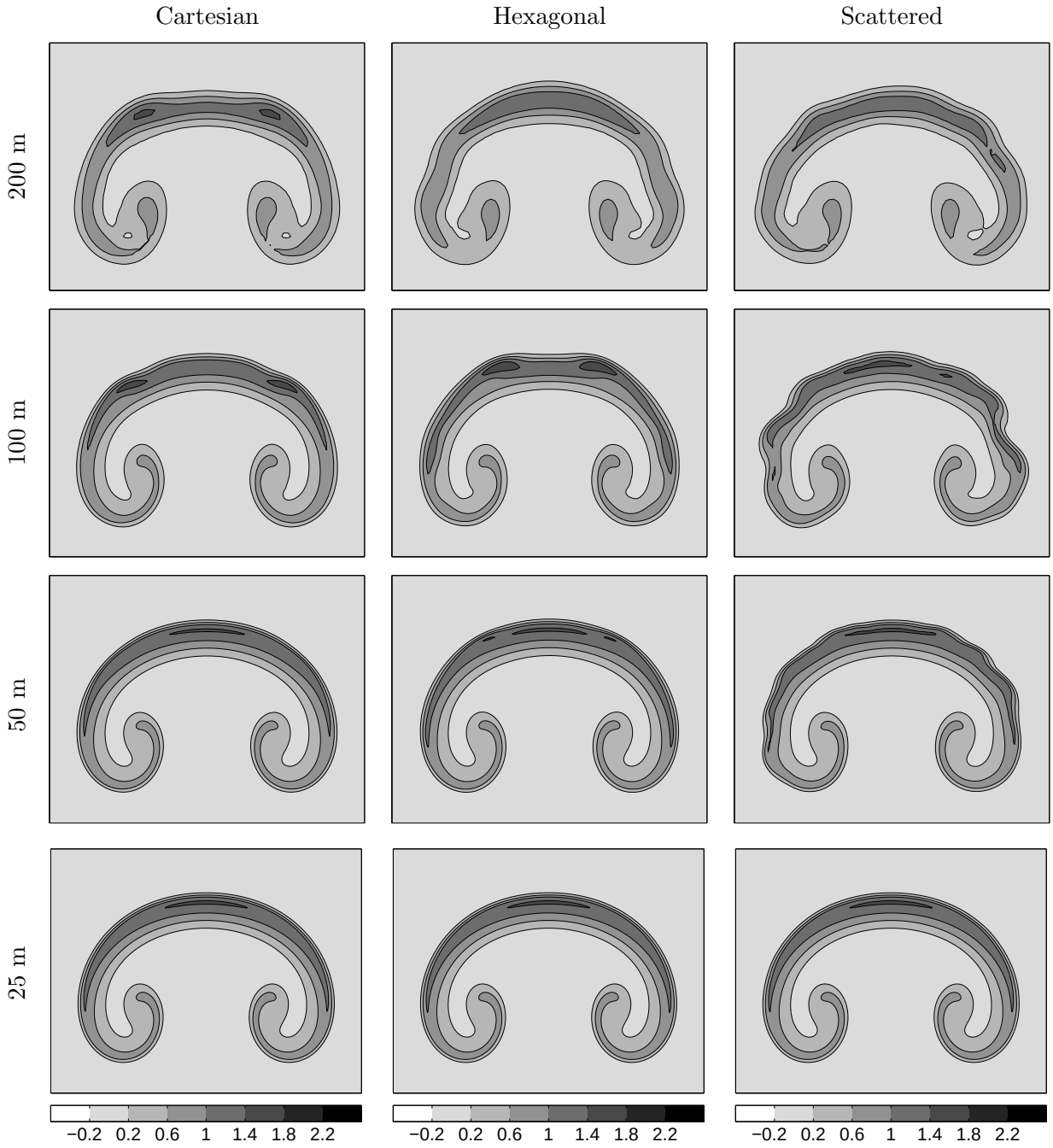

Figure 7.4: Numerical solutions for θ' at $t = 1100$ seconds in the rising bubble case $\left(\mu = 10 \text{ m}^2/\text{s}\right)$ at different resolutions on the three different types of node sets. In all cases, $\phi(r) = r^7$ with polynomials up to fourth degree on a 37-node stencil, and Δ^3-type hyperviscosity is applied. The viewing window is $[1.5, 8.5] \times [4, 9.5]$ km^2, and contours are from -0.2 to 2.2 by 0.4 K.

h [m]	400	200	100	50	25
Δt [s]	1	1/2	1/4	1/8	1/16
Error, Cartesian	0.5429	0.3253	0.1526	0.0279	0.0018
Error, Hexagonal	0.6285	0.2972	0.2385	0.0868	0.0041
Error, Scattered	0.5487	0.3066	0.2640	0.1192	0.0058

Table 7.1: Relative ℓ_2 errors for rising bubble test case. The reference is a 12.5 meter RBF-FD solution on hexagonal nodes.

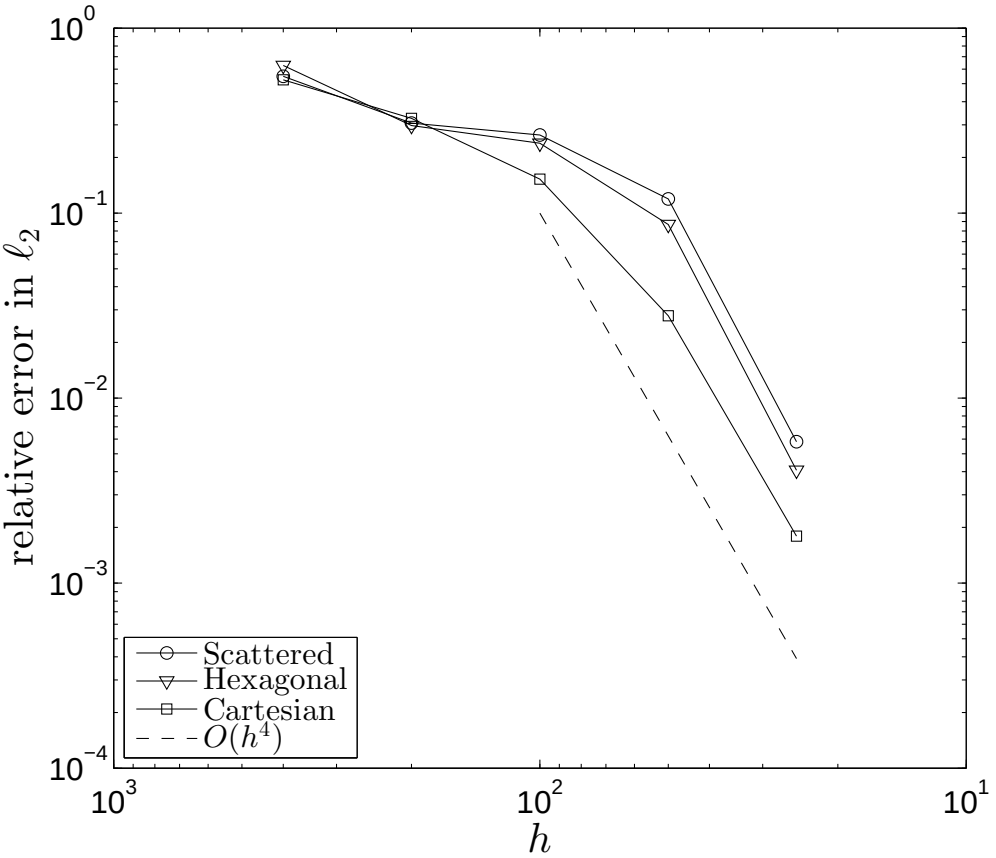

Figure 7.5: Convergence plot for θ' in RBF-FD solutions at 400, 200, 100, 50, and 25 meter resolutions. The reference is a 12.5 meter RBF-FD solution on hexagonal nodes.

The numerical solutions for θ' converge $O\left(h^4\right)$ to the reference on all three node-sets. Qual-

itatively, the results on the three different types of nodes look reasonable at low resolution, and at high resolution they look identical. In practical weather applications, the low-resolution runs are of the most interest, because even limited-area weather simulations typically use a resolution of 1 km or more. In other words, real-world weather approximations never enjoy numerical convergence, being perpetually confined to low-resolution and never capturing all scales correctly.

7.4.2 Rising Bubble, $\mu = 2 \times 10^{-5}$ $\mathbf{m^2/s}$

This case is the same as 7.4.1, except that we lower the viscosity from $\mu = 10$ m^2/s to $\mu = 2 \times 10^{-5}$ m^2/s. This adjustment makes the governing equations a more realistic model (the viscosity of dry air in Earth's atmosphere is about 2×10^{-5} m^2/s), but the small amount of natural diffusion makes the problem much more difficult to solve numerically. In fact, using the viscosity of air, convergence to the true solution on any reasonable set of nodes is impossible. With no exact solution to compare to, a convergence analysis would be pointless in this case, so instead we analyze the qualitative properties of the solutions on the three types of node-set.

Looking at Figure 7.6, we make some observations regarding the RBF-FD results on the three types of node-sets. The numerical solutions on cartesian and hexagonal nodes maintain symmetry, but they are significantly deformed from the smooth bubble solution that we might expect after solving the problem with $\mu = 10$ m^2/s. On the other hand, while the scattered-node solution has many small vortices around the periphery of the bubble, it maintains the same basic shape that we saw in the high-viscosity case. Also, since there is virtually no diffusion in the problem, we expect that the true solution maintains a minimum value of 0 and a maximum value of 2, and the scattered-node approximation, while still off, does a better job of maintaining these values.

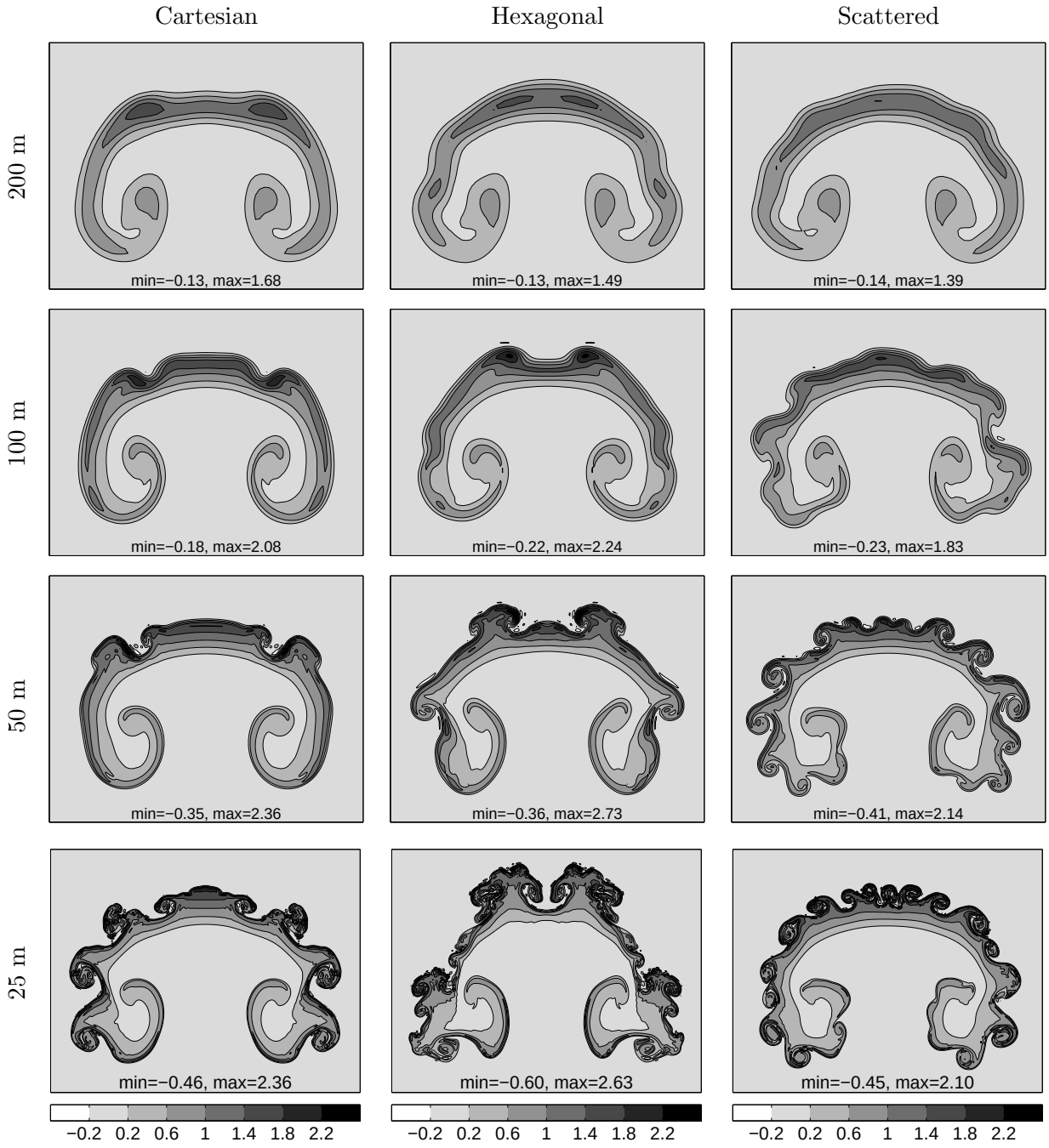

Figure 7.6: Numerical solutions for θ' at $t = 1100$ seconds in the rising bubble case $\left(\mu = 2 \times 10^{-5}\ \mathrm{m}^2/\mathrm{s}\right)$ at different resolutions on the three different types of node sets. In all cases, $\phi(r) = r^7$ with polynomials up to fourth degree on a 37-node stencil, and Δ^3-type hyperviscosity is applied. The viewing window is $[1.5, 8.5] \times [4, 9.5]\ \mathrm{km}^2$, with contours from -0.2 to 2.2 by 0.4 K.

7.5 Test Case: Density Current

In this popular test case [29, 18, 23, 24], a mass of cool air starts in the middle of the domain and descends to the surface, spreading out and producing Kelvin-Helmholz rotors as it moves along the bottom boundary (See Figure 7.7). In order that numerical solutions begin to converge to the exact solution under refinement, a non-physical viscosity of $\mu = 75$ m^2/s is used in the governing equations (7.3).

The spatial domain is $[-25.6, 25.6] \times [0, 6.4]$ km^2, the time domain is $[0, 900]$ s, and the hydrostatic background states are defined by

$$\overline{\theta} = 300, \qquad \overline{\pi}(z) = 1 - \frac{g}{c_p \overline{\theta}} z.$$

All of the unknowns are initially set to zero, except for the potential temperature perturbation θ', which is defined via a C^1 perturbation in the temperature T. Let

$$T'\big|_{t=0} = \begin{cases} -\frac{15}{2}\left[1 + \cos\left(\pi_c \tilde{r}\right)\right], & \tilde{r} \leq 1, \\ 0, & \tilde{r} > 1, \end{cases}$$

where $\pi_c = 3.14159\ldots$ is the usual trigonometric constant, $\tilde{r}(x,z) = \sqrt{\left(\frac{x-x_c}{x_r}\right)^2 + \left(\frac{z-z_c}{z_r}\right)^2}$, $(x_c, z_c) = (0 \text{ km}, 3 \text{ km})$, and $(x_r, z_r) = (4 \text{ km}, 2 \text{ km})$. The initial condition for θ' is derived from the initial condition for T':

$$\theta'\big|_{t=0} = \left(\theta - \overline{\theta}\right)_{t=0} = \frac{T|_{t=0}}{\pi|_{t=0}} - \overline{\theta} = \frac{\overline{\theta}\overline{\pi} + T'|_{t=0} - \overline{\theta}\overline{\pi}}{\overline{\pi}} = \frac{T'|_{t=0}}{\overline{\pi}}.$$

In summary, the initial conditions for the four unknowns are:

$$u|_{t=0} = 0, \qquad w|_{t=0} = 0, \qquad \theta'\big|_{t=0} = \frac{T'|_{t=0}}{\overline{\pi}}, \qquad \pi'\big|_{t=0} = 0.$$

7.5.1 Density Current, $\mu = 75$ $\mathbf{m^2/s}$

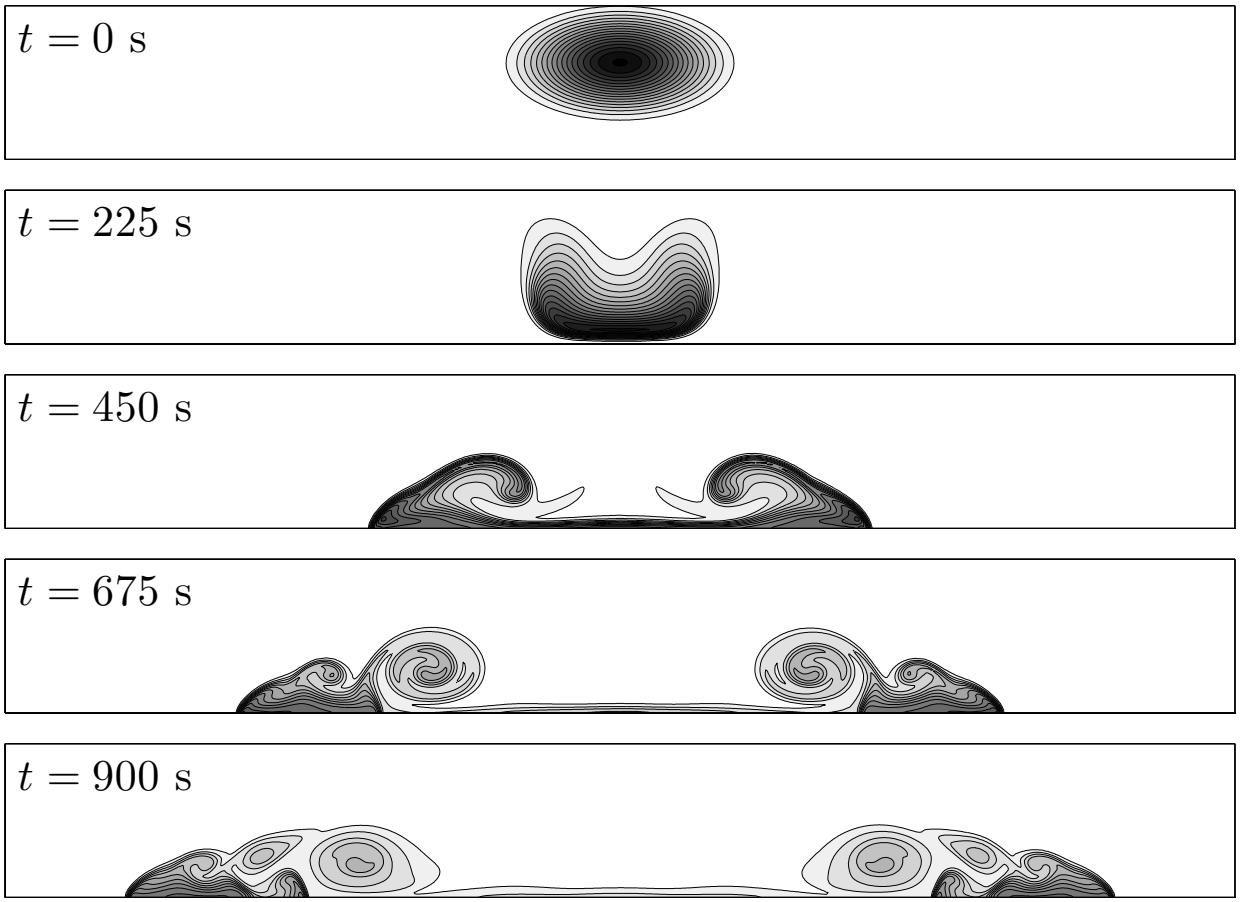

Figure 7.7: Snapshots of θ' on scattered nodes using an approximate node-spacing of 100 meters. The viewing window is $[-19.2, 19.2] \times [0, 6.4]$ km^2, and contours are from -16.5 to -0.5 by 1K.

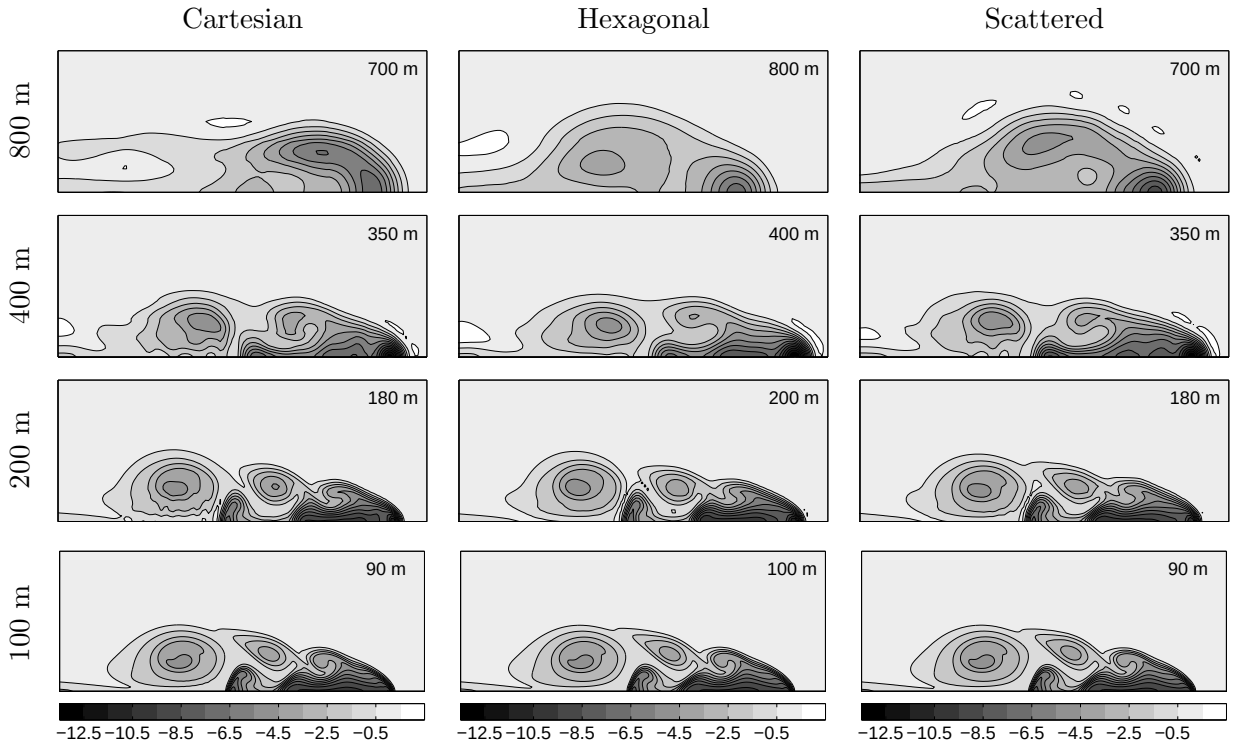

Figure 7.8: Numerical solutions for θ' at various resolutions on the three different types of node sets. In all cases $\phi(r) = r^7$ with polynomials up to fourth degree on a 37-node stencil using Δ^3-type hyperviscosity. The viewing window is $[4, 16.5] \times [0, 4.8]$ km^2 and contours are from -12.5 to 0.5 by 1 K.

h [m]	800	400	200	100	50
Δt [s]	5/3	5/6	5/12	5/24	5/48
Error, Cartesian	0.7898	0.5004	0.3662	0.0697	0.0071
Error, Hexagonal	0.6312	0.5605	0.3453	0.0409	0.0102
Error, Scattered	0.6848	0.4700	0.1854	0.0316	0.0063

Table 7.2: Relative ℓ_2 errors for the density current test case. The reference is a 25 meter RBF-FD solution on scattered nodes.

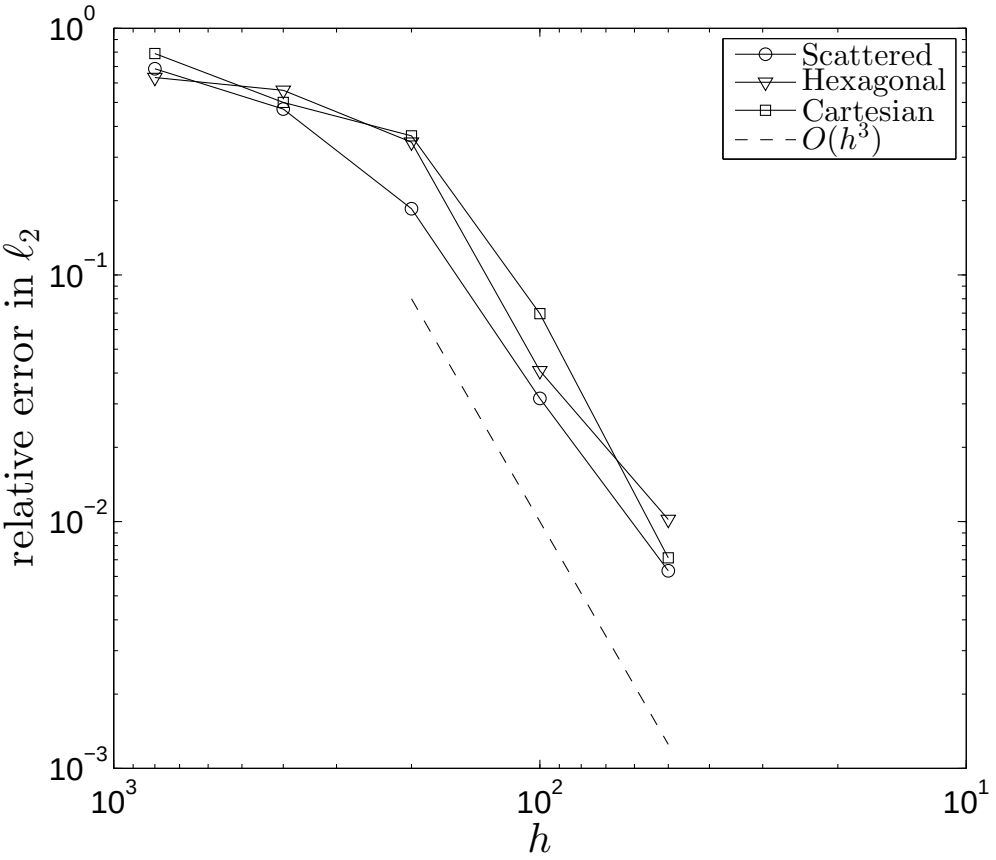

Figure 7.9: Convergence plot for $h = 800$, 400, 200, 100, and 50 meters in the Straka density current case 7.5.1. The reference is a 25 meter scattered-node solution. In all cases $\phi(r) = r^7$ with polynomials up to fourth degree on a 37-node stencil using Δ^3-type hyperviscosity.

7.5.2 Density Current, $\mu = 2 \times 10^{-5}$ $\mathbf{m}^2/\mathbf{s}$

In the low-viscosity case, since the numerical solutions should not be expected to converge, the purpose is to confirm that the method is robust enough that it can still be applied, and that it produces qualitatively reasonable numerical solutions. This case is the most realistic, and also the most computationally difficult, because we use essentially zero viscosity and the solution interacts strongly with the bottom boundary.

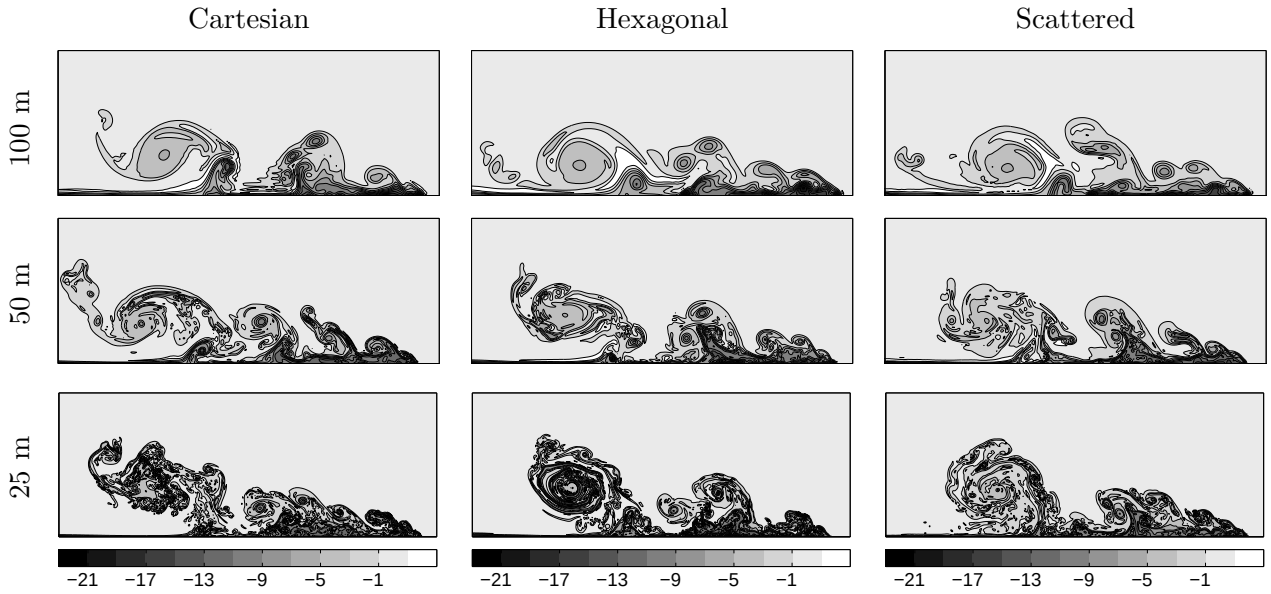

Figure 7.10: Numerical solution for θ' at 100, 50, and 25 meter resolutions using the three different types of node-sets. The viewing window is $[3900, 16500] \times [0, 4800]$ m^2. In all cases $\phi(r) = r^7$, polynomials up to 4^{th} degree are included on a 37-node stencil, and Δ^3-type hyperviscosity is used. Contours are from -21 to 1 by 2 K.

In each of the above test cases, RBF-FD based on PHS and polynomials produces qualitatively reasonable results, and explicit time-stepping is stable. In situations where convergence is feasible, the numerical solution converges to the true solution according to the highest degree polynomial included. Although we do not include a comparison with other methods here, a much more detailed discussion of the method can be found in Appendix C, where RBF-FD is shown to compete well with alternative methods in the literature.

Chapter 8

Conclusions

A new local PHS and polynomial based method for RBF-FD was introduced and numerically validated in several settings. The method is simple, accurate, and robust, bypassing the technical difficulties of previous RBF-FD implementations such as ill-conditioning and stagnation errors. The new RBF-FD scheme was applied first for interpolation and derivative approximation, and then for the numerical solution of elliptic and hyperbolic PDEs. Finally, the local method was combined with explicit time-stepping to solve benchmark problems from numerical weather prediction.

For interpolation and approximation of differential operators, it was shown that the order of convergence is controlled by the degree of polynomial support, not by the stencil-size or the type of RBF. In order to avoid having to choose a shape parameter, one can use a finitely-differentiable PHS RBF rather than the typical Gaussian or Multiquadric, and this change does not affect the order of convergence of the scheme.

A similar convergence trend, where the level of polynomial support determines the order of convergence, was also observed for the numerical solution of elliptic PDEs. In this setting, the RBF-FD method is robust enough to be applied on domains with boundaries using scattered nodes, even without special boundary treatment (ghost nodes), provided the stencil-size is large enough.

In the NWP test cases, the set of compressible Navier-Stokes equations in 2-D were numerically solved in a MOL approach, using hyperviscosity to achieve stability in time. The RBF-FD results are in good agreement with others published in the literature, and RBF-FD has the advantage of being geometrically flexible, and equally applicable on any reasonable set of nodes.

Bibliography

[1] G. A. Barnett, N. Flyer, and L. J. Wicker. An RBF-FD polynomial method based on polyharmonic splines for the Navier-Stokes equations: Comparisons on different node layouts. Journal of Computational Physics, submitted.

[2] V. Bayona, N. Flyer, B. Fornberg, and G. Barnett. On the role of polynomials in RBF-FD approximations: II. Numerical solution of Elliptic PDEs.

[3] P. N. Blossey and D. R. Durran. Selective monotonicity preservation in scalar advection. Journal of Computational Physics, 227(10):5160 – 5183, 2008.

[4] S. Bochner. Monotone funktionen, stieltjessche integrale und harmonische analyse. Mathematische Annalen, 108(1):378–410, 1933.

[5] J. Duchon. Splines minimizing rotation-invariant semi-norms in sobolev spaces. In Walter Schempp and Karl Zeller, editors, Constructive Theory of Functions of Several Variables, volume 571 of Lecture Notes in Mathematics, pages 85–100. Springer Berlin Heidelberg, 1977.

[6] G. F. Fasshauer. Meshfree Approximation Methods with MATLAB. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2007.

[7] N. Flyer, B. Fornberg, G. Barnett, and V. Bayona. On the role of polynomials in RBF-FD approximations: I. Interpolation and Accuracy.

[8] N. Flyer, E. Lehto, S. Blaise, G. Wright, and A. St-Cyr. A guide to RBF-generated finite differences for nonlinear transport: Shallow water simulations on a sphere. Journal of Computational Physics, 231(11):4078 – 4095, 2012.

[9] B. Fornberg. A practical guide to pseudospectral methods, volume 1. Cambridge university press, 1998.

[10] B. Fornberg, T. A. Driscoll, G. Wright, and R. Charles. Observations on the behavior of radial basis function approximations near boundaries. Computers and Mathematics with Applications, 43(35):473 – 490, 2002.

[11] B. Fornberg and N. Flyer. A Primer on Radial Basis Functions with Applications to the Geosciences. SIAM, Philadelphia, 2015.

[12] B. Fornberg and N. Flyer. Solving PDEs with Radial Basis Functions. Acta Numerica, pages 215–258, 2015.

[13] B. Fornberg and E. Lehto. Stabilization of RBF-generated finite difference methods for convective PDEs. Journal of Computational Physics, 230(6):2270 – 2285, 2011.

[14] B. Fornberg, E. Lehto, and C. Powell. Stable calculation of Gaussian-based RBF-FD stencils. Computers and Mathematics with Applications, 65(4):627 – 637, 2013.

[15] B. Fornberg and C. Piret. A stable algorithm for flat radial basis functions on a sphere. SIAM Journal on Scientific Computing, 30(1):60–80, 2008.

[16] B. Fornberg and G. Wright. Stable computation of multiquadric interpolants for all values of the shape parameter. Computers and Mathematics with Applications, 48(56):853 – 867, 2004.

[17] B. Fornberg and J. Zuev. The runge phenomenon and spatially variable shape parameters in {RBF} interpolation. Computers and Mathematics with Applications, 54(3):379 – 398, 2007.

[18] F. X. Giraldo and M. Restelli. A study of spectral element and discontinuous galerkin methods for the Navier-Stokes equations in nonhydrostatic mesoscale atmospheric modeling: Equation sets and test cases. Journal of Computational Physics, 227(8):3849 – 3877, 2008.

[19] R. L. Hardy. Multiquadric equations of topography and other irregular surfaces. Journal of geophysical research, 76(8):1905–1915, 1971.

[20] W. Haussmann. Modern developments in multivariate approximation: 5th international conference, Witten-Bommerholtz (Germany), September 2002. International Series of Numerical Mathematics. Birkhauser Verlag, 2003.

[21] E. J. Kansa. Multiquadrics - A scattered data approximation scheme with applications to computational fluid-dynamics - I surface approximations and partial derivative estimates. Computers and Mathematics with Applications, 19(89):127 – 145, 1990.

[22] 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. Computers and Mathematics with Applications, 19(89):147 – 161, 1990.

[23] M. R. Norman, R. D. Nair, and F. HM Semazzi. A low communication and large time step explicit finite-volume solver for non-hydrostatic atmospheric dynamics. Journal of Computational Physics, 230(4):1567–1584, 2011.

[24] K. V. Ooyama. A dynamic and thermodynamic foundation for modeling the moist atmosphere with parameterized microphysics. J. Atmos. Sci., 58:2073–2102, 2000.

[25] A. Robert. Bubble convection experiments with a semi-implicit formulation of the euler equations. Journal of the Atmospheric Sciences, 50(13):1865–1873, 1993.

[26] Y. V. S. S. Sanyasiraju and G. Chandhini. Local radial basis function based gridfree scheme for unsteady incompressible viscous flows. Journal of Computational Physics, 227(20):8922 – 8948, 2008.

[27] I. J. Schoenberg. Metric spaces and completely monotone functions. Annals of Mathematics, 39(4):pp. 811–841, 1938.

[28] C Shu, H Ding, and K. S Yeo. Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible NavierStokes equations. Computer Methods in Applied Mechanics and Engineering, 192(78):941 – 954, 2003.

[29] J. M. Straka, R. B. Wilhelmson, L. J. Wicker, J. R. Anderson, and K. K. Droegemeier. Numerical solutions of a non-linear density current: A benchmark solution and comparisons. International Journal for Numerical Methods in Fluids, 17(1):1–22, 1993.

[30] L. J. Wicker and W. C. Skamarock. Time-splitting methods for elastic models using forward time schemes. Monthly Weather Review, 130:2088–2097, 2002.

[31] G. B. Wright, N. Flyer, and D. A. Yuen. A hybrid radial basis function–pseudospectral method for thermal convection in a 3-D spherical shell. Geochemistry, Geophysics, Geosystems, 11(7), 2010.

Appendix A

Manuscript 1. On the role of polynomials in RBF-FD approximations: I. Interpolation and Accuracy

On the role of polynomials in RBF-FD approximations: I. Interpolation and accuracy

Natasha Flyer *
Institute for Mathematics Applied to Geosciences
National Center for Atmospheric Research
Boulder, CO 80305, USA

Bengt Fornberg †and Gregory A. Barnett ‡
Department of Applied Mathematics
University of Colorado
Boulder, CO 80309, USA

Victor Bayona §
Institute for Mathematics Applied to Geosciences
National Center for Atmospheric Research
Boulder, CO 80305, USA

July 6, 2015

Abstract

{To be written...}

1 Introduction

{Recall super-quickly FD⇒PS⇒RBF⇒RBF-FD. With RBFs increasingly being used for large-scale applications, global RBFs are often being replaced by local RBF-FD approximations. This is to be expected, for several reasons: (i) global RBFs lead to full matrices, with can cause high operation counts, (ii) derivatives are local properties of a function, so local approximations are natural in the context, and (iii) multi-core and distributed memory computing benefit from spatial locality.}

We first very briefly review global RBFs in Section 2 and discuss for these polynomial reproduction and stagnation errors in Section 3. The main novelties of this study are contained in Section 4, where we discuss the use of supplementary polynomials when creating RBF-FD approximations.

Email: flyer@ucar.edu
†*Email*: fornberg@colorado.edu
‡*Email*: gregory.barnett@colorado.edu
§*Email*: vbayona@ucar.edu

Type of basis function	**Radial function $\phi(\mathbf{r})$**	**Reproduced degree**
BE (Bessel; $p = 1, 2, \ldots, d$)	$J_{p/2-1}(\varepsilon r)/(\varepsilon r)^{p/2-1}$	all
Cubic	r^3	$d+2$
TPS (Thin Plate Spline)	$r^2 \log r$	$d+1$
Linear	r	d
MQ (Multiquadric)	$\sqrt{1+(\varepsilon r)^2}$	d
IMQ (Inverse MQ)	$1/\sqrt{1+(\varepsilon r)^2}$	$d-2$
IQ (Inverse quadratic)	$1/(1+(\varepsilon r)^2)$	$d-3$
GA (Gaussian)	$e^{-(\varepsilon r)^2}$	none
RBFs with compact support	various formulas	none

Table 1: Common RBF types, and the degrees of polynomials global RBFs based on these can reproduce exactly on general infinite d-D node layouts.

Key issues concern the effect on accuracy and conditioning. RBF-FD approximations based on piecewise smooth radial functions of polyharmonic spline (PHS) type turn out to be of particular interest. Finally, the main conclusions are summarized in Section 5.

2 Some basic RBF relations

A RBF interpolant $s(\underline{x})$ to scattered data in d-D takes the form

$$s(\underline{x}) = \sum_{k=1}^{n} \lambda_k \phi(||\underline{x} - \underline{x}_k||) \,, \tag{1}$$

where $||\cdot||$ denotes the standard Euclidean norm. Table 1 lists a number of common choices for the *radial function* $\phi(r)$. The significance of the last column 'Reproduced degree' will be described in Section 3.1.

With data f_k at node $\underline{x}_k$, $k = 1, 2, \ldots, n$, the coefficients λ_k in (1) can be found by solving the linear system

$$\begin{bmatrix} \phi(||\underline{x}_1 - \underline{x}_1||) & \phi(||\underline{x}_1 - \underline{x}_2||) & \cdots & \phi(||\underline{x}_1 - \underline{x}_n||) \\ \phi(||\underline{x}_2 - \underline{x}_1||) & \phi(||\underline{x}_2 - \underline{x}_2||) & \cdots & \phi(||\underline{x}_2 - \underline{x}_n||) \\ \vdots & \vdots & & \vdots \\ \phi(||\underline{x}_n - \underline{x}_1||) & \phi(||\underline{x}_n - \underline{x}_2||) & \cdots & \phi(||\underline{x}_n - \underline{x}_n||) \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} . \tag{2}$$

Remarkably, the symmetric coefficient matrix A is guaranteed to be non-singular for many choices of $\phi(r)$, no matter how the n nodes (assumed distinct) are scattered in any number d of dimensions.

2.1 Global RBFs with polynomials

Equation (1) is often modified to also include polynomial terms, together with matching constraints on the expansion coefficients. For example, including up to linear terms in 2-D would amount to

using

$$s(\underline{x}) = \sum_{k=1}^{n} \lambda_k \phi(||\underline{x} - \underline{x}_k||) \; + \; \gamma_1 + (\gamma_2 x + \gamma_3 y) \; , \qquad \text{with the constraints} \quad \sum_{k=1}^{n} \lambda_k = \sum_{k=1}^{n} \lambda_k x_k = \sum_{k=1}^{n} \lambda_k y_k = 0. \tag{3}$$

Extensive theory on these approximation types can be found in several recent monographs [2, 6, 27]. {In particular, ... summarize some key non-singularity theorems}

3 Polynomial reproduction and stagnation errors for global RBFs

In the global RBF case, the phenomenon of *polynomial reproduction* adds insights to the issue of *stagnation errors*, meaning errors that fail to decrease to zero under continuing node refinement.

3.1 Polynomial reproduction

The first case (BE) in Table 1 has some unique features [7, 12], which are advantageous in certain applications [19, 26]. However, BE RBFs are quite different from the other listed cases (primarily due to its Fourier transform having compact support), and we will not discuss it further here.

The topic of polynomial reproduction has a rich history [2, 5, 24]. We will here at first discuss the special case of reproducing a constant on an equispaced lattice. It suffices in this case to only consider the growth/decay rates of the radial functions $\phi(r)$ for $r \to \infty$. Initially focusing on 1-D, let the node points be $x_k = k$, $k = -N, \ldots, -1, 0, 1, \ldots, N$, where we later will let $N \to \infty$.

3.1.1 GA interpolants in 1-D

For each N, and with all data values $f_k = 1$, it will hold for $i = -N, \ldots, +N$ that $1 = \sum_{k=-N}^{N} \lambda_k e^{-(\varepsilon\,(k-i))^2}$. In the limit of $N \to \infty$, the unique solution will have all the λ_k the same; $\lambda_k = \lambda$ with

$$\lambda = 1 / \sum_{k=-\infty}^{\infty} e^{-(\varepsilon\,k)^2}. \tag{4}$$

The GA interpolant then becomes $s(x) = \lambda \sum_{k=-\infty}^{\infty} e^{-(\varepsilon\,(k-x))^2}$. This function $s(x)$ has period 1, but is not identical to 1. Constant data is not identically reproduced.

3.1.2 Linear RBF ($\phi(r) = r$) interpolants in 1-D

When using $\phi(r) = r$, the counterpart to (4) becomes $\lambda = 1 / \sum_{k=-\infty}^{\infty} |k|$. Given that the sum diverges, this relation cannot be satisfied by any non-zero constant value for λ. However, inspection shows that

$$\lambda_k = \begin{cases} \frac{1}{2N}\,, & k = \pm N \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

achieves $s(x) = \sum_{k=-\infty}^{\infty} \lambda_k \phi(|\,x - x_k|) \equiv 1$ for $-N \le x \le N$. Letting $N \to \infty$, the constant 1 has been identically reproduced. We can likewise identically reproduce x, but will fail to similarly reproduce x^2.

3.1.3 Some additional comments on polynomial reproduction

The remarks above indicate that two entirely different coefficient patterns can arise when trying to reproduce constant data. For GA, the counterpart to (4) in any number of dimensions will always converge, and identical reproduction of a constant will therefore fail. Generalizing from Linear to MQ basis functions, we can readily find a small group of expansion coefficients to use near $x = \pm N$ such that their combined effect is virtually the same as using (5). Constant and linear functions can then again be reproduced exactly in the limit of $N \to \infty$. To understand why the degree that can be reproduced in these cases increases by exactly one each time one dimension is added (cf. the terms 'd' in the last column of Table 1) requires the more advanced analysis cited above. However, if the goal is limited to assessing whether a constant is reproduced, or not (which suffices for our following discussion of stagnation errors for global RBFs), checking the convergence/divergence of the (multi-dimensional) counterparts to (4) gives the right answer.

The ability of many smooth RBF types to exactly reproduce polynomials up to some finite degree *in the infinite domain limit* has given rise to confusing statements about RBF approximations having only algebraic $O(h^p)$-type accuracy (with h being a 'typical' node distance), when convergence generally is spectral, of type $O(e^{-c/h})$. For classical FD methods, convergence rates can be found by inspecting the highest degree polynomial for which an approximation is *exact*. In the RBF case, there is no equally clear connection between polynomial reproduction and actual convergence rate.

3.2 Stagnation errors for global RBF approximations

Moving all nodes closer to each other by reducing h and, at the same time, increasing ε to keep εh constant leaves the matrix in (2) unchanged and therefore its condition number unaffected. Following this somewhat common practice can however give rise to *stagnation errors* (sometimes denoted saturation errors) - failure of convergence in the $h \to 0$ limit. In this limit, any smooth function will locally (on the scale of the node spacing) appear increasingly like a constant. As an example, on an equi-spaced 2-D infinite lattice, a GA interpolant to data identically one will develop a 2-D periodic structure, as seen over one 'period box' in Figure 1 (in the case of $\varepsilon = 1/h$). The deviation from being identically one may seem small (here about $4 \cdot 10^{-4}$), but remains unchanged at this level as $h \to 0$. For a p^{th} derivative, this implies divergence at the rate $O(1/h^p)$. With, say, MQ in place of GA (and still in 2-D), polynomial reproduction holds up through quadratic polynomials, and errors would instead become of size $O(1/h^{p-3})$. This still falls far short of the spectral convergence that should be expected for smooth global RBFs if ε was kept fixed rather than increased as $\varepsilon = O(1/h)$.

One possible way to avoid these stagnation errors is to increase ε less rapidly than as $O(1/h)$ (or not increase it at all), and then address the ill conditioning of RBF-Direct (a direct implementation of (2), (1)) by the use of a *stable algorithm*, e.g. RBF-QR [11, 15, 20] or RBF-GA [14].

4 RBF-FD approximations

{The RBF-FD idea ... describe/illustrate super-briefly}

We limit in the following our attention to Gaussians (GA) $\phi(r) = e^{-(\varepsilon r)^2}$ and polyharmonic splines (PHS) $\phi(r) = \begin{cases} r^m & , \quad m = 1, 3, 5, \ldots \\ r^m \log r & , \quad m = 2, 4, 6, \ldots \end{cases}$ (which contain the Linear, Cubic, and TPS RBF types in Table 1 as special cases).

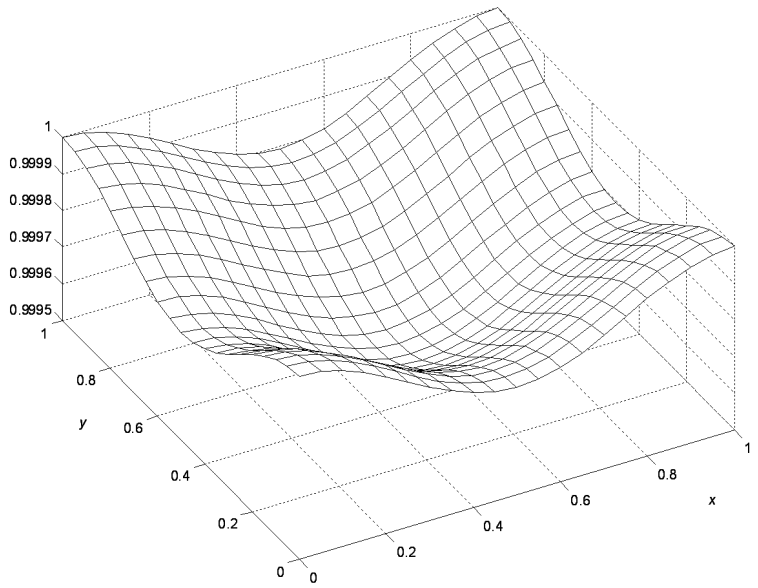

Figure 1: The GA interpolant to node data that is identically one on an infinite 2-D unit spaced lattice, shown in the case of $\varepsilon = 1/h$, with $h = 1$, over its period $[0, 1] \times [0, 1]$.

4.1 RBF-FD weights

{An RBF-FD formula takes the form ..... explain - maybe show some picture of the RBF-FD concept.}

4.1.1 Calculation of RBF-FD weights

In 2-D, using RBFs + linear terms, the RBF-FD weights w_i at locations $\underline{x}_i$, $i = 1, \ldots, n$ for approximating a linear operator L at a location $\underline{x}_c$, are obtained by solving

$$\left[\begin{array}{ccc|cccc} & & & 1 & x_1 & y_1 \\ & A & & \vdots & \vdots & \vdots \\ & & & 1 & x_n & y_n \\ \hline - & - & - & + & - & - & - \\ 1 & \cdots & 1 & & & \\ x_1 & \cdots & x_n & & 0 & \\ y_1 & \cdots & y_n & & & \end{array}\right] \left[\begin{array}{c} w_1 \\ \vdots \\ w_n \\ \hline - \\ w_{n+1} \\ w_{n+1} \\ w_{n+3} \end{array}\right] = \left[\begin{array}{c} L\phi(||\underline{x} - \underline{x}_1||)|_{\underline{x}=\underline{x}_c} \\ \vdots \\ L\phi(||\underline{x} - \underline{x}_n||)|_{\underline{x}=\underline{x}_c} \\ \hline - \\ L\ 1\ |_{\underline{x}=\underline{x}_c} \\ L\ x\ |_{\underline{x}=\underline{x}_c} \\ L\ y\ |_{\underline{x}=\underline{x}_c} \end{array}\right], \tag{6}$$

where A is the same matrix as in (2). In the solution vector, the entries w_{n+1}, w_{n+2}, w_{n+3} should be discarded. The pattern of entries in (6) generalizes immediately to more space dimensions and to higher degree polynomials. Derivation are given in [10], Section 5.1.4, and in [20]. {For non-singularity theorems ... see ...}

4.1.2 Least squares interpretation of RBF-FD weights with polynomial augmentation

RBF-FD with polynomial augmentation can be actually interpreted the equality-constrained quadratic programming problem

$$\min_{\underline{w}} \; J\left(\underline{w}\right) = \frac{1}{2}\underline{w}^T A\underline{w} - \underline{w}^T L\underline{\phi} \quad \text{subject to} \quad P^T\underline{w} = L\underline{p}. \tag{7}$$

where A is the same matrix as in (2), $\vec{w}$ and $L\vec{\phi}$ are column vectors formed by w_i and $L\phi(||\underline{x} - \underline{x}_i||)|_{\underline{x}=\underline{x}_c}$, $i = 1, \ldots, n$, respectively, P^T is a rectangular matrix formed by the augmented polynomials, and $L\vec{p}$ is a column vector formed by the polynomial terms $L\ 1\ |_{\underline{x}=\underline{x}_c}$, $L\ x\ |_{\underline{x}=\underline{x}_c}$, $L\ y\ |_{\underline{x}=\underline{x}_c}$, $\ldots$.

This follows from using Lagrange multipliers to find the minimum with respect to $\underline{w}$ and $\underline{\lambda}$ of the Lagrangian

$$\mathcal{L}(\underline{w}, \underline{\lambda}) = \frac{1}{2}\underline{w}^T A\underline{w} - \underline{w}^T L\underline{\phi} + \underline{\lambda}^T \left(P^T\underline{w} - L\underline{p}\right). \tag{8}$$

For positive definite RBFs, the A matrix is positive definite. For conditionally positive definite RBFs, such as PHS (which is the one used in this work), it is satisfied by definition that

$$\underline{c}^T A\underline{c} \geq 0 \qquad \text{for} \qquad P^T\underline{c} = 0, \qquad \forall\, \underline{c} \in \mathbb{R}^n, \tag{9}$$

meaning that A is positive definite on the kernel of P^T [6]. Therefore, the Lagrangian (8) is convex and has a unique minimum in both cases. As a result, $\nabla_{\underline{w},\underline{\lambda}}\mathcal{L}(\underline{w}, \underline{\lambda}) = 0$ yields the system of equations (6).

From the result above follows that RBFs with polynomial augmentation, among all solutions of the under-determined problem $P^T\underline{w} = L\underline{p}$, selects the one which minimizes the functional in (7). This provides an intuitive understanding for why RBFs with polynomial augmentation are better than using only RBFs (low order algebraic accuracy for typical PHS) or only polynomials (prone to singularities for scattered nodes). Moreover, it indicates that the convergence rate depends mostly on the degree of the included polynomial terms, and less on the stencil size.

4.2 Test problem for exploring RBF-FD accuracies

We consider for the present set of tests a stencil with $n = 56$ ME (minimum energy)-like nodes, as shown within the unit circle ($R = 1$) in Figure 2 (a). In most RBF-FD cases, one is only interested in the accuracy close to the stencil center, although nearly one-sided approximations may become needed in connection with boundaries. Hence, we will here consider two sets of test locations, illustrated in parts (b) and (c) of the figure. We choose as a test function

$$f(x, y) = 1 + \sin(4x) + \cos(3x) + \sin(2y), \tag{10}$$

shown over $R = 1$ in Figure 3 (a). In two steps of refinement, we then reduce R from 1 to 0.3 and to 0.1. The displays in Figure 2 remain unchanged apart from being shrunk in proportion to R. Subplots (b) and (c) of Figure 3 show how the test function, when zoomed in, appears gradually more like a tilted plane. If the vertical axis scale is fixed, the surfaces would instead increasingly look like a flat plane (a constant).

All the error results given in Sections 4.2.1 and 4.2.2 represent the worst case encountered at any of the evaluation nodes indicated in Figure 2 (b) - the case most representative of typical RBF-FD

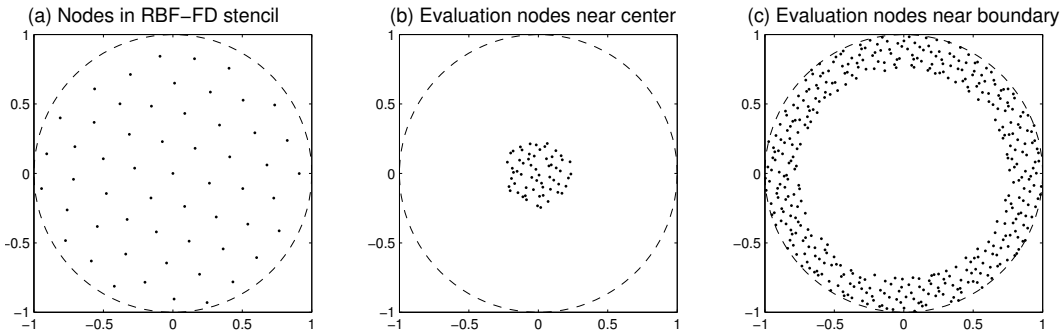

Figure 2: (a) RBF-FD stencil ($n = 56$ nodes) in the case of $R = 1$. Parts (b) and (c): Two sets of evaluation points at which the approximation's interpolation accuracy is evaluated.

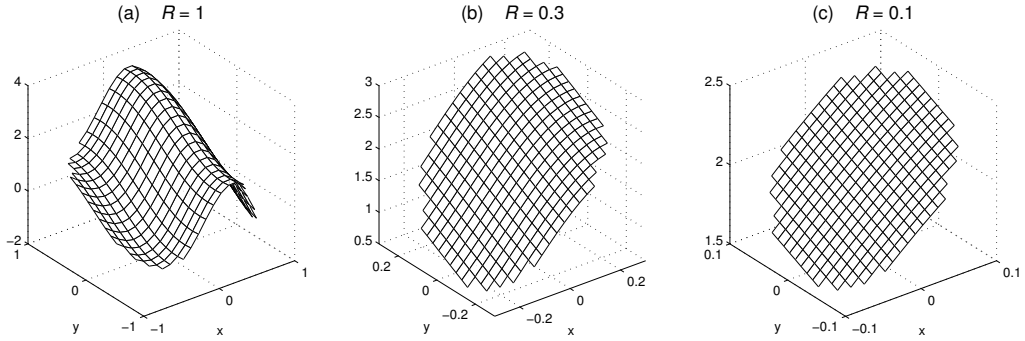

Figure 3: Test function (10), when displayed over $x^2 + y^2 \leq R^2$, with $R = 1$, $R = 0.3$, and $R = 0.1$, respectively.

usage. For evaluation nodes as indicated in Figure 2 (c), errors were generally about 100 times larger, but otherwise following equivalent patterns. The degree of polynomial terms included is varied from -1 (meaning no such terms) up through 9. In the present 2-D case with $n = 56$ nodes, the matrix in (6) becomes singular for degrees 10 and above.

4.2.1 RBF-FD using GA with polynomials

This section describes the test results shown in Figure 4. The accuracy plots in the top row are calculated in sufficiently high precision (80 decimal digits) that the conditioning of the matrix in (6) does not affect the results. When decreasing ε from very large values, the error stays for some time constant, but then decreases to a minimum that is essentially independent of the degree of supplementary polynomials. The location in ε of this minimum remains roughly unchanged as the stencil size is made smaller (corresponding to R decreasing from 1 to 0.3 and 0.1), and occurs in the present case around $\varepsilon = 0.5$. This invariance of the optimal ε value under refinement was (in the absence of supporting polynomials {check this!}) noted previously in [3]. When using a stable algorithm (such as RBF-QR [11, 20] or RBF-GA [14]), giving full numerical access to small ε-values already when using double precision, there is little (if any) reason to include extra polynomials.

The middle row of subplots differs from the top one in that it was calculated using double precision (16 digit) instead of extended precision (80 digit). The loss of accuracy for low ε (below the dotted

horizontal lines) correspond closely to where the bottom row of subplots shows the condition number of the coefficient matrix in (6) to exceed about 10^{15} (indicated with matching dotted lines).

If one uses GA + polynomials, in double precision, and without employing a stable algorithm, the last two rows of subplots show the best strategy to be to lower ε until reaching a condition number close to 10^{15}. Including only low degree polynomials has little effect, but including close to the maximal permissible degree of polynomials increases the accuracy significantly.

When including close to the maximal number of polynomials, the system in (6) has nearly doubled in size, increasing the cost of the weight calculation by around a factor of 8. For comparison, the RBF-GA algorithm (applied to GA RBFs, without needing polynomial terms) increases the cost by about a factor of 10. These two approaches for reaching high accuracy thus break roughly equal in cost. When operating in the small ε regime, it is not surprising that added polynomials offer no benefit, since the GA themselves then span virtually the same function space (and exactly the same in the $\varepsilon \to 0$ limit) [4, 16, 25].

4.2.2 RBF-FD using PHS with polynomials

This section describes the test results shown in Figure 5. Instead of GA RBFs, we now use the PHS RBFs $\phi(r) = r^m$ with $m = 1, 3, 5, \ldots, 21$. Tests with $\phi(r) = r^m \log r$ for $m = 2, 4, 6, \ldots$ showed no significant differences, and could have been used equally well. While some theory suggests using $\phi(r) = r^{\,m}$, $m = 1, 3, 5, \ldots$ in odd dimensions, and $\phi(r) = r^{\,m} \log r$, $m = 2, 4, 6, \ldots$ in even dimensions, that distinction may not be necessary.

Our first observation is that the top two rows of subplots (corresponding to quadruple vs. double precision) are virtually indistinguishable from each other. This is in spite of the last row showing the condition number for the matrix in (6) reaching high values in different parts of the parameter domain. In contrast to the previous GA case, the high condition number is this time harmless in actual computing, as it here only reflects a limitation in its definition (being sensitive to scaling issues for the rows and columns of a matrix even when these have no adverse effect on the computed interpolant), as observed earlier in [18].

Increasing the polynomial degree is seen to always be beneficial, whereas the choice of RBF degree matters less and less when node sets are refined (with stencil sizes fixed). Heuristically, smooth functions look under refinement increasingly like low degree polynomials, and the polynomial part of the RBF-FD approximations then 'take over' from the non-smooth RBF part. The latter remains however important to ensure non-singularity.

4.2.3 Some concluding observations of GA vs. PHS when used with polynomials

In the present test case, three choices appear quite similar both in terms of the accuracy that can be reached, and cost of generating the RBF-FD stencils: (i) GA without additional polynomials, implemented for small ε for example by the RBF-GA algorithm, (ii) GA with polynomials close to maximal permissible order, using ε values that give condition numbers around 10^{15}, and (iii) PHS, again with polynomials close to maximal permissible order.

Using a smooth RBF type (such as GA) for low ε together with weight calculations in extended precision would also reach the same high accuracy levels, but the cost will be far higher than with the choices above.

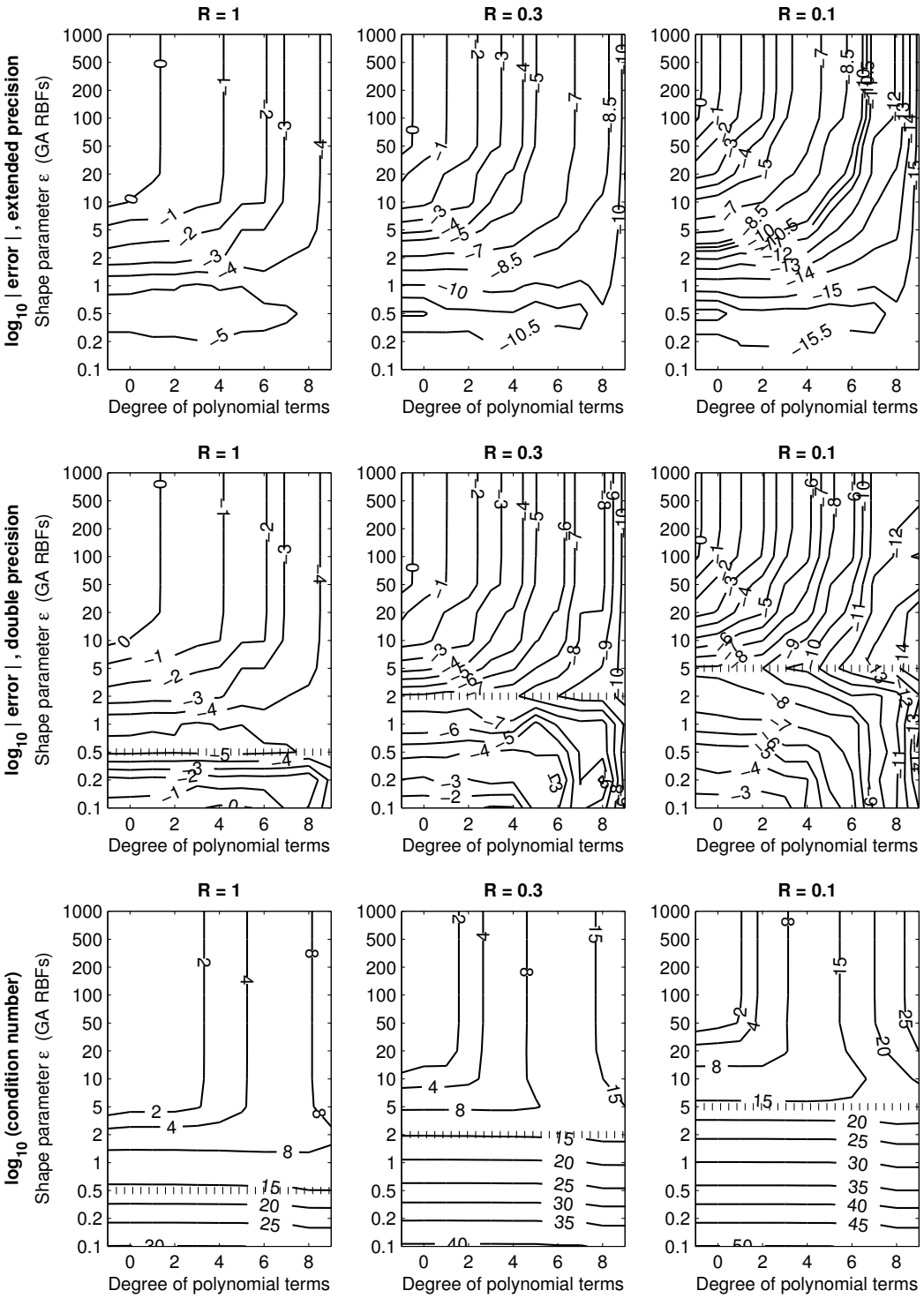

Figure 4: Top two rows: Accuracy ($\log_{10}$ error)in GA-based RBF-FD approximations, as functions of shape parameter ε and degree of included polynomial terms, calculated using 80 decimal digits extended precision vs. in standard 16 digit double precision, Bottom row: $\log_{10}$ of the condition number for the system (6) (calculated using 80 decimal digits extended precision).

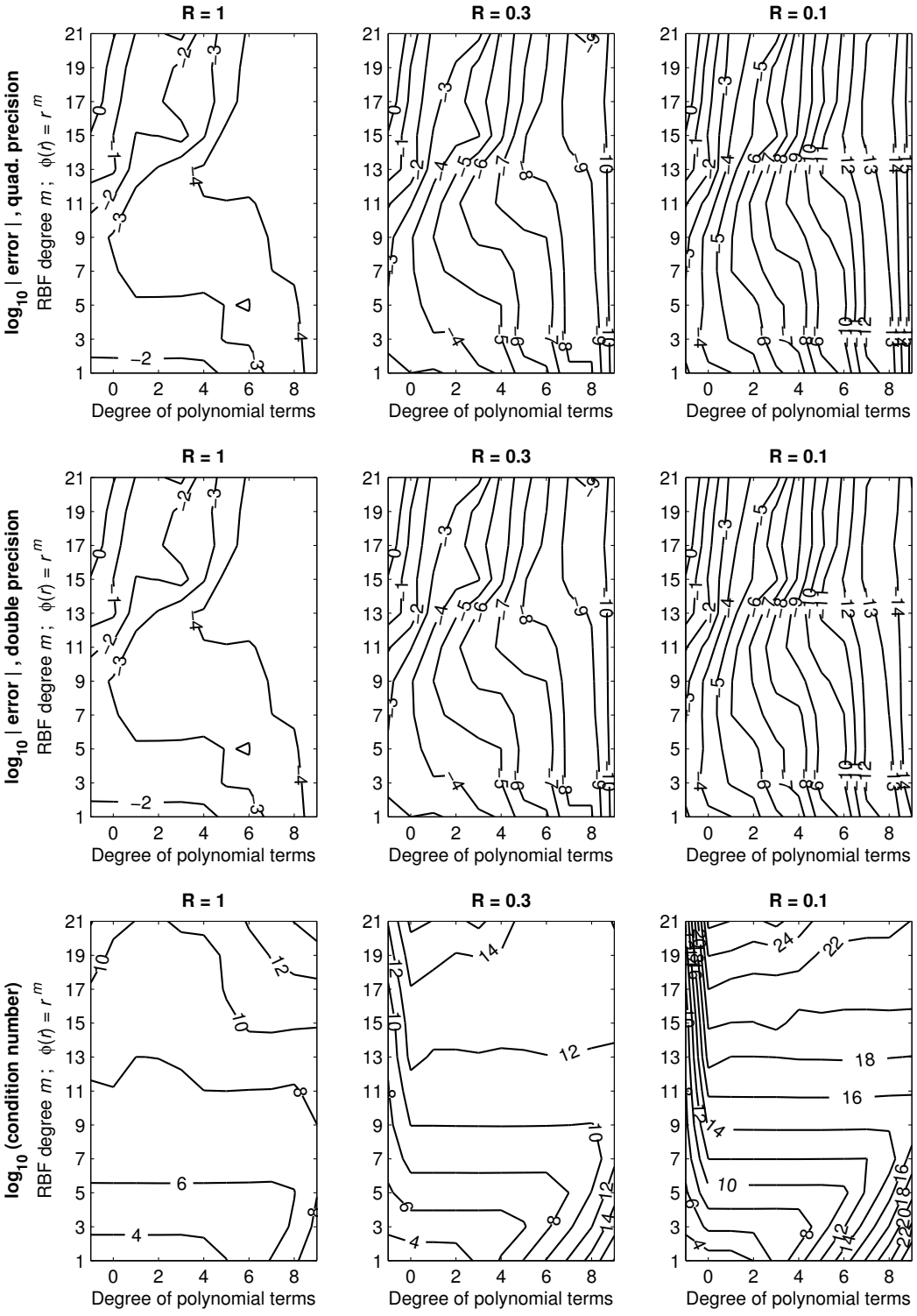

Figure 5: Top two rows: Accuracy ($\log_{10}$ error)in PHS-based RBF-FD approximations, as functions of the RBF degree and the degree of included polynomial terms, calculated using 34 decimal digits quad precision vs. in standard 16 digit double precision, Bottom row: $\log_{10}$ of the condition number for the system (6) (calculated using quad precision).

None one of the three options (i) - (iii) above will encounter stagnation errors under node refinement. The option (iii) is the easiest to implement, and will be the focus of our following discussion and demonstration example.

4.3 Stagnation errors when using PHS with polynomials

We have seen above why GA without polynomials encounters stagnation errors under node refinement if ε is increased to keep the condition number constant. In the case of PHS without polynomials, stagnation errors arise again, but for a different reason. Inspecting the left edge of the three subplots in either the top or the middle row of subplots in Figure 5 show again failure of convergence under refinement, i.e. a stagnation error.

We will in this section provide an intuitive qualitative explanation in 1-D for the cause of these stagnation errors (for interpolation, when using $\phi(r) = r^3$, r^5, r^7, etc.) The case of $\phi(r) = r^3$ was analyzed in [8]. With nodes $-1 = x_1 < x_2 < \ldots < x_n = 1$, the interpolant $s(x) = \sum_{i=1}^{n} \lambda_i |x - x_i|^3$ becomes a cubic spline with the non-intuitive end conditions

$$\begin{array}{ll} s''(-1) & = s'(1) - 2s'(-1) - \frac{3}{2}(s(1) + s(-1)) \\ s''(1) & = 2s'(1) - s'(-1) - \frac{3}{2}(s(1) + s(-1)) \end{array} , \tag{11}$$

which should be contrasted to the end conditions

$$s''(-1) = s''(1) = 0 \tag{12}$$

for the *natural* cubic spline. As a result, constant data $f_i = 1$ will produce a non-constant interpolant, as shown in Figure 6 (a) together with its counterpart for $\phi(r) = r^7$. Large errors oscillate and decay as we move from the stencil edge towards its interior. The geometric decay rates for these oscillations are known from spline theory: $O(\alpha^{|k|})$ with $\alpha = 2 - \sqrt{3} \approx 0.2679$ for k steps in the case of cubic splines, generalizing to splines of odd degrees as follows:

$$\begin{array}{lccccccc} \text{spline degree} & 3 & 5 & 7 & 9 & \cdots & \infty \\ \alpha & 0.2679 & 0.4306 & 0.5353 & 0.6080 & \cdots & 1.0000 \end{array} . \tag{13}$$

Figure 6 (b) shows that the amplitudes of the observed oscillations indeed decay at the predicted rates. The interpolation error near the stencil center is thus the edge error reduced by a certain factor. Shrinking the distance h between the stencil node points will therefore make no difference with regard to the interpolation error near the stencil center.

It was shown in [8] that supplementing $\phi(r) = r^3$ with a polynomial of degree 1 and imposing the matching constraints changes the resulting end conditions from (11) to (12), i.e. the approximation becomes exact for all linear functions. The pattern continues: when including higher order polynomial terms, these 'take over' and provide the corresponding levels of accuracy under RBF-FD refinement. The RBF terms remain important by providing other benefits, such as safety against singularities due to particular node layouts.

Independent of the dimension, we would from the argument above expect errors under refinement to be of the form $O(h^{\ell-k+1})$ when approximating a k^{th} derivative, using $\phi(r) = r^p$, p odd, supplemented with polynomials of degree ℓ. The two parts of Figure 7 show how the error when approximating the Laplacian varies with h when using $\phi(r) = r^3$ and $\phi(r) = r^7$, respectively, in both cases together with polynomials of different orders. For each choice of h, the worst case was

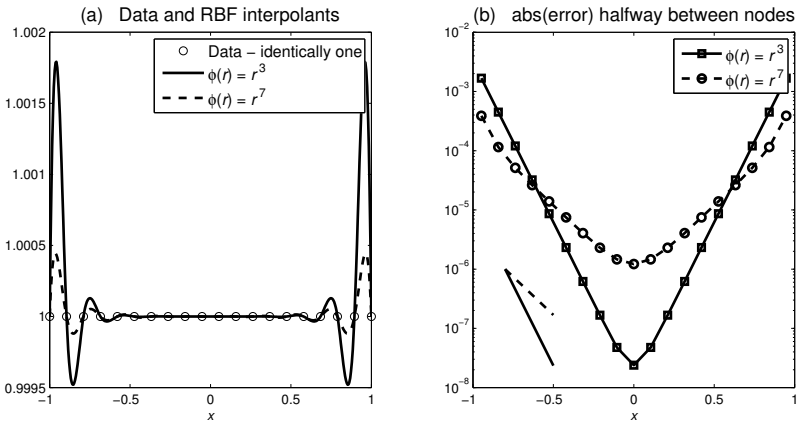

Figure 6: (a) The $\phi(r) = r^3$ and $\phi(r) = r^7$ RBF interpolants for $n = 18$ node constant equispaced data. (b) The errors at the half-way points between the nodes, corresponding roughly to the extrema of the oscillations in the interpolants. The errors decrease towards the interval center closely following the theoretical rates (13) for the two corresponding spline cases, shown by the short straight line segments at the bottom left.

recorded over 1,000 Halton distributed locations of stencil centers within the unit circle. The log-log plots are in excellent agreement with all the observations above - in particular with the errors being of size $O(h^{\ell-k+1})$ with here $k = 2$. The thick dashed lines mark $Error = 10^{-15}/h^2$, which is the accuracy barrier that arises if weights with magnitudes adding up to $10/h^2$ are combined with function values that are uncertain to 10^{-16}. For example, the weights for the standard second order FD approximation

$$\triangle = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \approx \begin{bmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{bmatrix} /h^2$$

add in magnitude up to $8/h^2$, with higher multiples of $1/h^2$ inevitable for more accurate approximations. The curves in Figure 7 reach right down to this barrier, showing that no significant digits have been lost due to any RBF-FD related conditioning issues. This 'barrier' appears in just the same position for regular lattice-based FD approximations, for which the weights are analytically known.

Moreover, the error under refinement is independent of the stencil size n, as it is pointed out in Section 4, where RBF-FD with polynomial augmentation is interpreted as a minimization problem. This is shown in Figure 8, where the error approximating the Laplacian using $n = 45$, 66 and 91 is computed using different polynomial oders. Notice that the convergence order is $O\left(h^{l-k+1}\right)$, independently of the stencil size n.

5 Conclusions

{To be written ...}

Acknowledgments:

The presented research was supported by the NSF grants DMS-0934317, OCI-0904599 and by Shell International Exploration and Production, Inc.

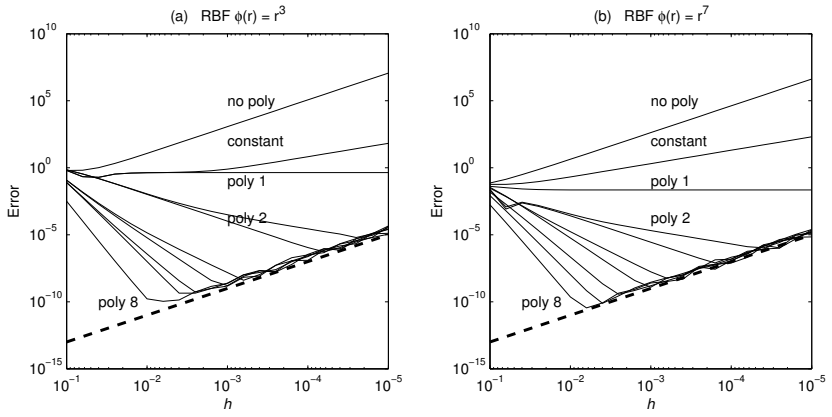

Figure 7: The worst error encountered when approximating the Laplacian using the $n = 54$ node PHS-based RBF-FD stencils, as described in the text. The curve legends ('no poly', 'constant', etc.) are in exactly the same locations in the two subplots, illustrating that the change when going from $\phi(r) = r^3$ to $\phi(r) = r^7$ indeed is minimal. The slopes of the different curves perfectly match the predictions (including the thick dashed lines, which mark $Error = 10^{-15}/h^2$).

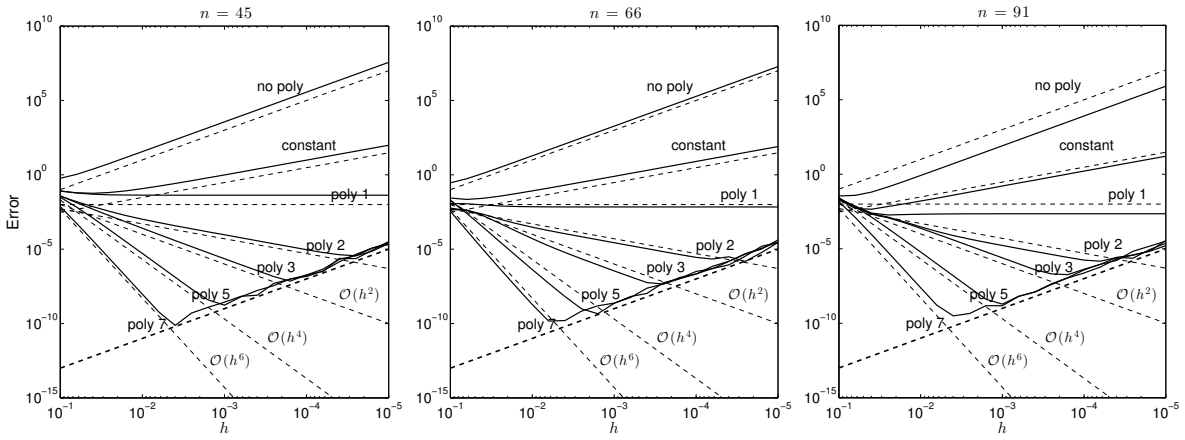

Figure 8: The worst error encountered when approximating the Laplacian using RBF-FD stencils based on r^7 + polynomials as described in the text. The thin dashed lines represent the convergence order $O\left(h^{l-k+1}\right)$ for each polynomail order l. Notice that the convergence ratio is independent of the stencil size and only depends on the degree of polynomial terms included. As in the previous figure, the thick dashed line marks the round-off limit $10^{-15}/h^2$.

References

[1] G. A. Barnett, N. Flyer, and L. J. Wicker, *An RBF-FD polynomial method for nonhydrostatic atmospheric modeling: Comparisons on different node layouts*, in preparation (2015).

[2] M. D. Buhmann, *Radial basis functions: Theory and implementations*, 12. Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, 2003.

[3] O. Davydov and D. T. Oanh, *Adaptive meshless centres and RBF stencils for Poisson equation*, J. Comput. Phys. **230** (2011), 287–304.

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

[5] N. Dyn and A. Ron, *Radial basis function approximation - From gridded centers to scattered centers*, Proc. London Math. Soc. **71** (1995), 76–108.

[6] G. E. Fasshauer, *Meshfree Approximation Methods with MATLAB*, Interdisciplinary Mathematical Sciences - Vol. 6, World Scientific Publishers, Singapore, 2007.

[7] N. Flyer, *Exact polynomial reproduction for oscillatory radial basis functions on infinite lattices*, Comput. Math. Appl. **51** (2006), 1199–1208.

[8] B. Fornberg, T. A. Driscoll, G. Wright, and R. Charles, *Observations on the behavior of radial basis functions near boundaries*, Comput. Math. Appl. **43** (2002), 473–490.

[9] B. Fornberg and N. Flyer, *Fast generation of 2-D node distributions for mesh-free PDE discretizations*, submitted (2015).

[10] ———, *A Primer on Radial Basis Functions with Applications to the Geosciences*, SIAM, Philadelphia, 2015.

[11] B. Fornberg, E. Larsson, and N. Flyer, *Stable computations with Gaussian radial basis functions*, SIAM J. Sci. Comput. **33(2)** (2011), 869–892.

[12] B. Fornberg, E. Larsson, and G. B. Wright, *A new class of oscillatory radial basis functions*, Comput. Math. Appl. **51** (2006), 1209–1222.

[13] B. Fornberg and E. Lehto, *Stabilization of RBF-generated finite difference methods for convective PDEs*, J. Comput. Phys. **230** (2011), 2270–2285.

[14] B. Fornberg, E. Lehto, and C. Powell, *Stable calculation of Gaussian-based RBF-FD stencils*, Comp. Math. Applic. **65** (2013), 627–637.

[15] B. Fornberg and C. Piret, *A stable algorithm for flat radial basis functions on a sphere*, SIAM J. Sci. Comput. **30** (2007), 60–80.

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

[17] S. C. Hagness, A. Taflove, and S. D. Gedney, *Finite-difference time-domain methods*, Handbook of Numerical Analysis - Special Volume Numerical Methods in Electromagnetics (W. H. A. Schilders and E. J. W. ter Maten, eds.), vol. XIII, Elsevier, Amsterdam, 2005, pp. 199–315.

[18] A. Iske, *On the approximation order and numerical stability of local Lagrange interpolation by polyharmonic splines*, Modern Developments in Multivariate Approximation (W. Haussmann, K. Jetter, M. Reimer, and J. Stöckler, eds.), International Series of Numerical Mathematics, vol. 145, Birkhäuser Verlag, Basel, 2003, pp. 153–165.

[19] S. H. Javaran, N. Khaji, and A. Noorzad, *First kind Bessel function (J-Bessel) as radial basis function for plane dynamic analysis using dual reciprocity boundary element method*, Acta Mech. **218** (2011), 247–258.

[20] E. Larsson, E. Lehto, A. Heryudono, and B. Fornberg, *Stable computation of differentiation matrices and scattered node stencils based on Gaussian radial basis functions*, SIAM J. Sci. Comput. **35** (2013), A2096–A2119.

[21] Y. Liu, *Fourier analysis of numerical algorithms for the Maxwell's equations*, J. Comput. Phys. **124** (1996), 396–416.

[22] B. Martin, B. Fornberg, and A. St-Cyr, *Seismic modeling with radial basis function generated finite differences (RBF-FD)*, Geophysics, to appear (2015).

[23] P.-O. Persson and G. Strang, *A simple mesh generator in Matlab*, SIAM Rev. **46** (2004), 329–345.

[24] M. J. D. Powell, *The theory of radial basis function approximation in 1990*, Advances in Numerical Analysis, Vol. II: Wavelets, Subdivision Algorithms and Radial Functions (W. Light, ed.), Oxford University Press, Oxford, UK, 1992, pp. 105–210.

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

[26] Y. Shi and C. H. Chan, *Improved 3D full-wave Green's function interpolation method*, Elec. Lett. **47(3)** (2011), 174–175.

[27] H. Wendland, *Scattered Data Approximation*, Cambridge Monographs on Applied and Computational Mathematics, vol. 17, Cambridge University Press, Cambridge, 2005.

Appendix B

Manuscript 2. RBF-FD approximations based on PHS with polynomial augmentation: II. Numerical solution of elliptic partial differential equations

RBF-FD approximations based on PHS with polynomial augmentation: II. Numerical solution of elliptic partial differential equations

Victor Bayona [*], Natasha Flyer [†]
Institute for Mathematics Applied to Geosciences
National Center for Atmospheric Research
Boulder, CO 80305, USA

Bengt Fornberg [‡] and Gregory A. Barnett [§]
Department of Applied Mathematics
University of Colorado
Boulder, CO 80309, USA

June 4, 2015

Abstract

RBF-generated finite differences (RBF-FD) have in the last decade emerged as a very powerful and flexible numerical approach for solving a wide range of PDEs. We find in the present study that combining polyharmonic splines (PHS) with multivariate polynomials offers an outstanding combination of simplicity, accuracy, and geometric flexibility when solving elliptic equations in irregular (or regular) regions. Test problems include the classical 2-D driven cavity, and also a 3-D global electric circuit problem with the earth's irregular topography as its bottom boundary. The results we find are fully consistent with previous results for data interpolation.

1 Introduction

In this follow-up paper to our previous work in [9], we address the use of RBF-FD approximations based on polyharmonic splines (PHS) with polynomial augmentation for the numerical solution of elliptic PDEs. This approach benefits from the properties of RBF-FD methods, such as sparsity, high-order convergence, local node refinement, geometric flexibility, easy to program, etc. It bypasses complications such as the selection of a shape parameter, ill-conditioning of some RBF

[*] *Email*: vbayona@ucar.edu
[†] *Email*: flyer@ucar.edu
[‡] *Email*: fornberg@colorado.edu
[§] *Email*: gregory.barnett@colorado.edu

collocation matrices and potential convergence failure under refinement (also known as stagnation errors) [10, 12, 11].

RBF-FD approximations based on PHS with polynomial augmentation are introduced in Section 2. Section 3 explains a ghost node approach for handling boundaries. Sections 4 and 5 describe the algorithm used to discretize the domains and how the linear system of equations are solved, respectively. The main results of this study are contained in Sections 6, where we discuss the use RBF-FD approximations based on PHS with polynomial augmentation for the solution of elliptic PDE problems. This is followed by Sections 7-9, where the method is applied to solve elliptic PDE problems of increasing complexity. These include the 2-D lid driven cavity and the 3-D Global Electric Circuit. Finally, the conclusions are given in Section 10.

2 RBF-FD formulation

RBF-FD methods can be viewed as a natural generalization of FD methods. As in FD, RBF-FD approximates a linear operator L at a location $\underline{x}_c$ as a linear combination of the values of the function at the n closest nodes which form the stencil,

$$Lu|_{\underline{x}_c} \approx \sum_{i=1}^{n} w_i u_i. \tag{1}$$

In classical FD, the weights w_i are determined by enforcing the approximation (1) to be exact for polynomials of as high degrees as possible. However, this approach is grid based and does not generalize well to scattered nodes in more than 1-D, with the Mairhuber-Curtis theorem being just one reason. Instead of only relying on multivariate polynomials, we include also RBFs that are centered at the nodes of the stencil of size n. This approach for finding weights is guaranteed to be non-singular no matter how the n nodes (assumed distinct) are scattered in any number of dimensions.

2.1 RBF-FD weights

When not augmenting the RBFs with polynomials, the RBF-FD weights w_i in (1) based on stencil nodes $\underline{x}_i$, $i = 1, 2, \ldots, n$ are obtained by solving

$$\begin{bmatrix} \phi(\|\underline{x}_1 - \underline{x}_1\|) & \cdots & \phi(\|\underline{x}_1 - \underline{x}_n\|) \\ \vdots & \ddots & \vdots \\ \phi(\|\underline{x}_n - \underline{x}_1\|) & \cdots & \phi(\|\underline{x}_n - \underline{x}_n\|) \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} L\phi(\|\underline{x} - \underline{x}_1\|)|_{\underline{x}_c} \\ \vdots \\ L\phi(\|\underline{x} - \underline{x}_n\|)|_{\underline{x}_c} \end{bmatrix}. \tag{2}$$

There are many choices of RBFs $\phi(r)$ that can be used [11, 7]. In this work, we limit our attention to the use of polyharmonic splines (PHS, here $\phi(r) = r^{2m-1}$, $m \in \mathbb{N}$) with polynomial augmentation [14, 9]. For example, using RBF-FD + linear terms, the RBF-FD weights w_i in 2-D are obtained

by solving

$$\left[\begin{array}{ccc|ccc} & & & 1 & x_1 & y_1 \\ & A & & \vdots & \vdots & \vdots \\ & & & 1 & x_n & y_n \\ \hline 1 & \dots & 1 & & & \\ x_1 & \dots & x_n & & 0 & \\ y_1 & \dots & y_n & & & \end{array}\right] \left[\begin{array}{c} w_1 \\ \vdots \\ w_n \\ \hline \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{array}\right] = \left[\begin{array}{c} L\phi\left(\|\underline{x}-\underline{x}_1\|\right)|_{\underline{x}_c} \\ \vdots \\ L\phi\left(\|\underline{x}-\underline{x}_n\|\right)|_{\underline{x}_c} \\ \hline L1|_{\underline{x}_c} \\ Lx|_{\underline{x}_c} \\ Ly|_{\underline{x}_c} \end{array}\right], \tag{3}$$

where A is the same matrix as in (2). In the solution vector, the entries λ_1, λ_2 and λ_3 should be discarded. For a proof see [11], Section 5.1.4. The pattern of entries in (3) can be easily generalized, resulting in the linear system of equations

$$\left[\begin{array}{cc} A & P \\ P^T & 0 \end{array}\right] \left[\begin{array}{c} \underline{w} \\ \underline{\lambda} \end{array}\right] = \left[\begin{array}{c} L\underline{\phi} \\ L\underline{p} \end{array}\right], \tag{4}$$

where P^T is a $m \times n$ matrix formed by the $m = \binom{k+d}{k}$ elements of the basis of polynomials of total degree less than or equal to k in $\mathbb{R}^d$, $\underline{\lambda}$ is a vector which contains the discarded weights and $L\underline{p}$ is a vector formed by the action of the operator L on the basis of polynomials evaluated on $\underline{x}_c$.

2.2 Least squares interpretation of RBF-FD with polynomial augmentation

The linear system of equations (4) is the solution of the equality-constrained quadratic programming problem

$$\min_{\underline{w}} \; J\left(\underline{w}\right) = \frac{1}{2}\underline{w}^T A \underline{w} - \underline{w}^T L\underline{\phi} \quad \text{subject to} \quad P^T\underline{w} = L\underline{p}. \tag{5}$$

This follows from using Lagrange multipliers to find the minimum with respect to $\underline{w}$ and $\underline{\lambda}$ of the Lagrangian

$$\mathcal{L}(\underline{w}, \underline{\lambda}) = \frac{1}{2}\underline{w}^T A \underline{w} - \underline{w}^T L\underline{\phi} + \underline{\lambda}^T\left(P^T\underline{w} - L\underline{p}\right). \tag{6}$$

For positive definite RBFs, the A matrix is positive definite. For conditionally positive definite RBFs, such as PHS (which is the one used in this work), it is satisfied by definition that

$$\underline{c}^T A \underline{c} \geq 0 \qquad \text{for} \qquad P^T\underline{c} = 0, \qquad \forall\, \underline{c} \in \mathbb{R}^n, \tag{7}$$

meaning that A is positive definite on the kernel of P^T [7]. Therefore, the Lagrangian (6) is convex and has a unique minimum in both cases. As a result, $\nabla_{\underline{w},\underline{\lambda}}\mathcal{L}(\underline{w}, \underline{\lambda}) = 0$ yields the system of equations (4). Notice that the discarded entries in (3) and (4) are the Lagrange multipliers $\underline{\lambda}$.

From the result above follows that RBFs with polynomial augmentation, among all solutions of the under-determined problem $P^T\underline{w} = L\underline{p}$, selects the one which minimizes the functional in (5). This provides an intuitive understanding for why RBFs with polynomial augmentation are better than using only RBFs (low order algebraic accuracy for typical PHS) or only polynomials (prone to singularities for scattered nodes). Moreover, it indicates that the convergence rate depends mostly on the degree of the included polynomial terms, and less on the stencil size.

3 Ghost nodes

The ghost (or fictitious) node approach has been used frequently in the past to implement boundary conditions both for finite difference [18] and for pseudospectral methods [5]. Avoiding fully one-sided stencils improves both accuracy and stability. It was described in [8] for solving elliptic PDEs with global RBFs. We use here the ghost node approach with RBF-FD approximations (rather than with global RBFs).

Consider an elliptic PDE problem defined in a domain Ω. To avoid one-sided stencils near the boundary, a layer of ghost nodes with as many nodes as boundary nodes is introduced outside the domain all around the boundary. Using these nodes to discretize the PDE avoid fully one-sided stencils. However, the resulting $N \times (N + N_g)$ differentiation matrix is not longer square, where N is the number of interior and boundary nodes and N_g is the number of ghost nodes. In this case, the differentiation matrix can be written as

$$\left[\begin{array}{cc} D & D_g \end{array} \right] \left[\begin{array}{c} \underline{u} \\ \underline{u_g} \end{array} \right] = \underline{f}, \tag{8}$$

where the block matrices are $(D)_{N\times N}$ and $(D_g)_{N\times N_g}$, $\underline{u}$ is the value of the unknowns on the interior and boundary nodes, $\underline{u_g}$ is the value of the unknowns on the ghost nodes and $\underline{f}$ is the right-hand side evaluated on the interior nodes.

In order to have a linear system of equations with as many unknowns as conditions, the PDE can also be enforced on the boundary, which leads to the square differentiation matrix,

$$\left[\begin{array}{cc} D & D_g \\ B & B_g \end{array} \right] \left[\begin{array}{c} \underline{u} \\ \underline{u_g} \end{array} \right] = \left[\begin{array}{c} \underline{f} \\ \underline{f_g} \end{array} \right], \tag{9}$$

where $(B)_{N_g\times N}$ and $(B_g)_{N_g\times N_g}$ are the matrices that contain the weigths dicretizing the PDE on the boundary and $\underline{f_g}$ is the right-hand side evaluated on the boundary nodes. By simple elimination, the differentiation matrix can be written as a function of the interior and boundary values as

$$\underbrace{\left[D \, - \, D_g \, B_g^{-1} \, B\right]}_{\hat{D}} \, \underline{u} \, = \, \underbrace{\underline{f} \, - \, D_g \, B_g^{-1} \, \underline{f_g}}_{\underline{\hat{f}}} \, . \tag{10}$$

For large stencil sizes, when a single layer of ghost nodes might not be enough to relieve the boundary effects, it is possible to increase the number of layers. For each layer, a new condition must be enforced on the boundary. A simple way to obtain these extra conditions for linear PDEs is by applying powers of the PDE. For example, $Lu = f$ for the first layer, $L^2u = Lf$ for the second layer, $L^3u = L^2f$ for the third layer, and so on (assuming that f is of a form such that we can evaluate Lf, L^2f, etc)

4 Domain discretization

The domains Ω considered in this work are discretized using scattered nodes based on the following repel algorithm. Other options for creating 2-D node distribution are described in [12].

1. Discretize the boundary $\partial\Omega$ with equispaced nodes using root-finding algorithms if needed.

2. Embed domain Ω into a regular domain, e.g. a rectangle, which is easy to discretize using structured nodes. The size of this regular domain must allow at least 4 layers of nodes all around the embedded domain. The 2-3 most exterior layers of nodes are called frame nodes.

3. Discard the structured nodes lying within a certain distance to the boundary nodes.

4. Boundary and frame nodes are fixed to their positions. A slighly random displacement is applied to the rest of the nodes, followed by the repulsion force

$$\vec{F} = \sum_{i=1}^{n} \frac{q_i}{\|\vec{r}_i\|_2^3} \vec{r}_i,$$

where $\vec{r}_i = (x - x_i, y - y_i)$, n is the n-th closest nodes to the i-th node and $q_i \sim 10^{-6}$, during a predefined number of iterations.

5. All nodes outside Ω are discarded. The domain Ω is finally discretized using the interior and boundary nodes.

5 Elliptic solver and preconditioner

Unless specified otherwise in the text, the linear system of equations is solved using the BICGSTAB iterative method with incomplete LU factorization as preconditioner and Cuthill-McKee ordering. The residual tolerance is set to 10^{-11} and the maximum number of iterations is equal to 1000.

6 Test problem for exploring accuracy of PHS with polyomial augmentation

In the following test problem, we explore the performance of PHS with polynomial augmentation when solving the Poisson equation

$$\begin{cases} \Delta u &= f(x,y), \quad \text{in } \Omega \\ u &= g(x,y), \quad \text{on } \partial\Omega \end{cases} \tag{11}$$

on the unit disc $\Omega = \{(x,y)\,,\, x^2 + y^2 \leq 1\}$, where $f(x,y)$ and $g(x,y)$ are computed from the test function

$$u(x,y) = \sin\left[10\,(x+y)\right]. \tag{12}$$

Scattered nodes based the repel algorithm from Section 4 are used to discretize the domain. Figure 1 shows the present node distribution for $h \approx 0.05$, where the dots, squares and crosses represent interior, boundary and exterior nodes, respectively.

As it is well-known, boundaries can affect the accuracy and stability of the solver due to the one-sided nature of the resulting stencils. To explore the effects of boundaries on PHS augmented with polynomials, we solve (11) using three different approaches. This is shown in Figure 2, where the error in the solution is plotted as a function of the stencil size and polynomial degree for different resolutions (row of subplots) and different methods for handling the boundary (column of subplots). The dash curved $n = \binom{m+2}{m}$ plotted in each figure represents the case where the stencil size is equal to the number polynomial terms. For higher values of n, the collocation matrix (4) is non-singular: below it, the columns of P will be linearly dependent.

In the first column, we neglect the existance of the boundary by considering the exterior nodes as boundary nodes and enforce the analytical solution (12) on them. This is an ideal case that can only

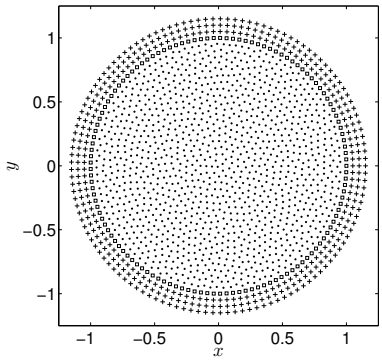

Figure 1: Unit disc discretized using $h \approx 0.05$: interior nodes (dot), boundary nodes (square) and exterior nodes (cross). Exterior nodes are treated differently depending on the approach used to handle the boundary.

be done because we know the analytical solution. As a result, one-sided stencils are avoided and an optimal reference solution is obtained. Notice that the error decreases as the node distribution is refined and also as the polynomial order increases. In contrast, an increase of the stencil size does not significantly improve the accuracy, specially for fine node distributions. Note also that only for $h = 0.01$ the accuracy deteriorates when the stencil size n is very close to the smallest stencil size that can be used for a given polynomial degree.

The second column shows the result of using the ghost node approach explained in Section 3. This is an effective method to avoid boundary effects in PDE problems. The exterior nodes, known as ghost nodes, are used to discretize the equation and the PDE is also enforced on the boundary to have an unisolvent linear system of equations with as many unknowns as equations. In this case, fully one-sided stencils are avoided and the accuracy and stability of the solver are not compromised. Notice that the results obtained are very similar to the ideal ones from the first column.

The third column shows the result of solving the problem directly without doing anything special to handle the boundary. In this case, exterior nodes are discarded and only interior and boundary nodes are considered. As a result, one-sided stencils are used near the boundary. When the node distribution is refined, the boundary effects are significant only for stencil sizes smaller than twice the number of polynomial terms included. Remarkably, it reproduces basically the same results than the ideal case outside this region. This is consistent with tests in [15] (using smooth global RBFs), which showed only insignificant differences between using ghost nodes and not doing so.

In this case, the eigenvalue distribution of the differentiation matrix also exhibits an interesting behaviour as a function of the stencil size. Figure 3 shows the eigenvalue distribution for $h \approx 0.025$, with 5th order polynomial terms and $n = 25$, 30 and 40. Notice that for $n = 25$, the eigenvalues are widely scattered over the complex plane, some of them with positive real part. As the stencil size increases, the eigenvalue distribution improves, becoming closer to the negative real axis. In the particular case $n = 40$, which corresponds to twice as many nodes as polynomials, the eigenvalues recover the distribution expected for a differentiation matrix approximating the Laplacian, with all of them very close to the negative real axis.

These results suggest that boundary effects can be bypassed by selecting a large enough stencil size. This is specially interesting for domains with irregular boundaries, where the ghost nodes approach

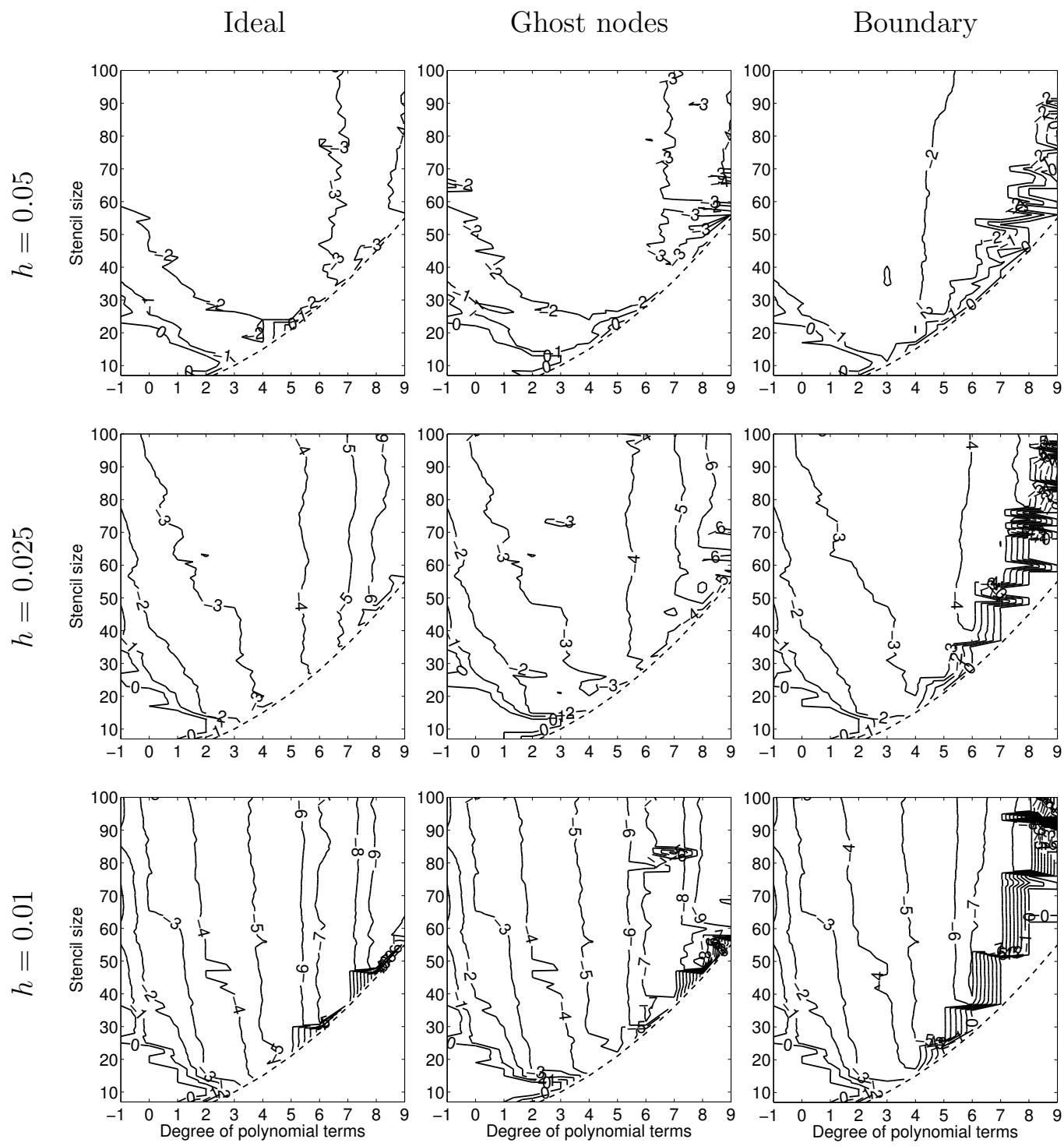

Figure 2: Accuracy ($\log_{10}$ error) when using PHS r^7 with polynomial augmentation to solve (11) as a function of the stencil size and degree of included polynomial terms. The rows show different resolutions and the columns different approaches to handle the boundaries. The dashed line show the smallest stencil size n that can be used for each degree of polynomial term.

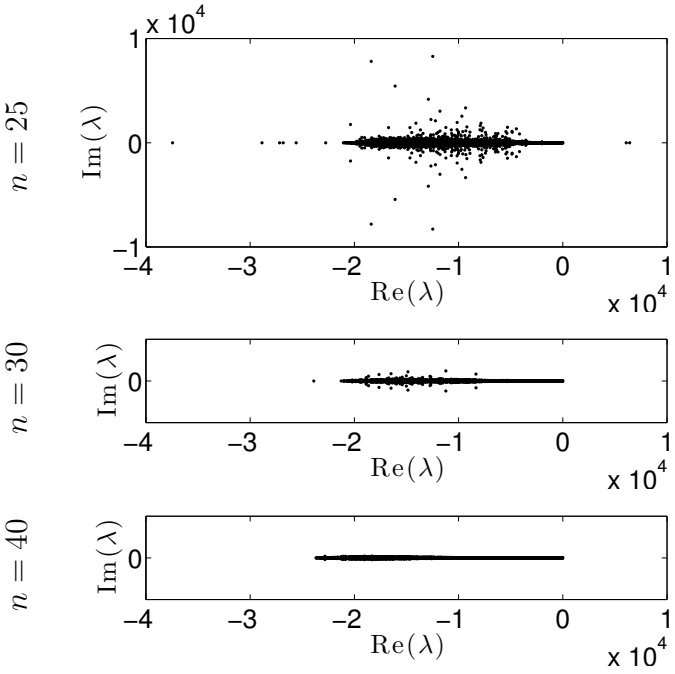

Figure 3: Eigenvalue distribution of the differentiation matrix that approximates (11) for the boundary case with $h \approx 0.025$, 5th order polynomial terms and $n = 25$, 30 and 40.

from Section 3 is non-trivial.

7 Elliptic PDE with variable coefficients

In this section, we explore further the convergence of PHS with polynomial augmentation as a function of the degree of the polynomial terms. For this purpose, we consider the general elliptic PDE with variable coefficients,

$$\Delta u + \eta(x,y)\frac{\partial^2 u}{\partial x \partial y} + \alpha(x,y)\frac{\partial u}{\partial x} + \beta(x,y)\frac{\partial u}{\partial y} + \gamma(x,y)u = f(x,y), \qquad \text{in } \Omega \tag{13}$$

and Dirichlet boundary conditions

$$u = g(x,y), \qquad \text{on } \partial\Omega \tag{14}$$

defined on the domain Ω shown in Figure 4 and with variable coefficients given by

$$\begin{aligned} \eta(x,y) &= \cos(10\pi y), \\ \alpha(x,y) &= \exp\left[-y^2 + \cos(4\pi x)\sin(3\pi y)\right], \\ \beta(x,y) &= -y\sin(4\pi x), \\ \gamma(x,y) &= x^2 y. \end{aligned}$$

The functions $f(x,y)$ and $g(x,y)$ are obtained assuming the analytical solution

$$u(x,y) = \sin(2\pi y^2 + 3\pi x) - \cos(\pi y - 2\pi x^2). \tag{15}$$

Notice that the variable coefficients are highly oscillatory and that a mixed derivative term $\frac{\partial^2 u}{\partial x \partial y}$ (which adversely affects diagonal dominance in FD contexts) is also included in (13).

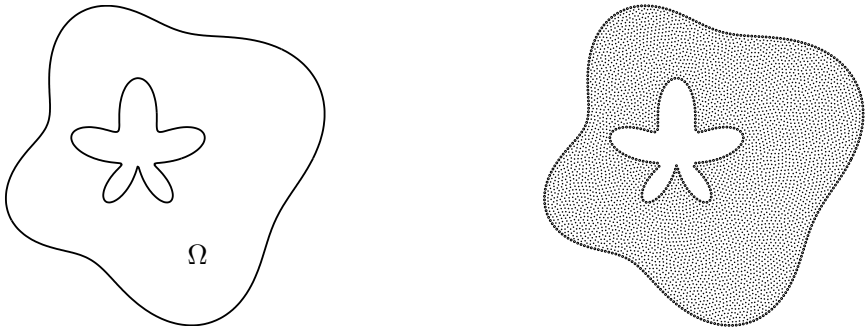

Figure 4: The left side of the figure shows the domain Ω and the right side a scattered discretization based on a node repel algorithm for $h \approx 0.025$.

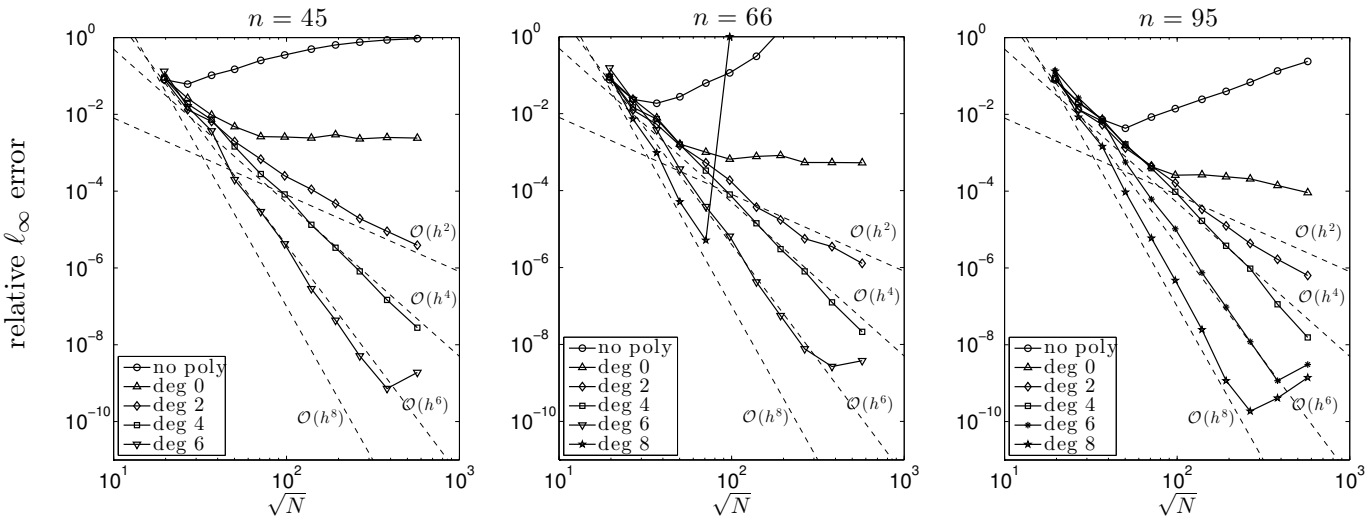

Figure 5: Relative ℓ_∞ error as a function of the number of nodes $\sqrt{N}$ when solving (13) using the stencil sizes $n = 45$, 66 and 95.

The domain is discretized based on the repel algorithm from Section 4. The right side of Figure 4 shows the discretized domain for $h \approx 0.025$. Following the results of Section 6, ghost nodes are not used in this problem.

Figure 5 shows the relative ℓ_∞ error as a function of $\sqrt{N}$ for different polynomial orders and stencil sizes $n = 45$, 66, and 95. Observe that the convergence order coincides with the degree of the polynomials included and is independent of the stencil size n. The stencil size only shifts the curves horizontally. Figure 5 also illustrates the effect of increasing the stencil size when boundaries are present. For example, the case $n = 45$ and 8th degree polynomial does not work. This is expected because the stencil size is equal to the corresponding number of polynomial terms. For $n = 66$, numerical solutions can be computed for coarse resolutions, while the method becomes ill-conditioned for finer distributions. By increasing the stencil size to $n = 95$, where the stencil size is about twice the number of polynomial terms, the performance of the method highly improves and the boundary effects disappear. This is the same effect observed previously in Section 6.

8 Non-linear elliptic PDE: Lid driven cavity flow

To test the performance of PHS with polynomial augmentation, we consider in this section the lid-driven cavity flow [13, 16, 1, 3], which is a well-known benchmark problem usually tested for code validation. In the context of RBF-FD methods, it was first considered in [17, 4]. The governing steady-state incomprensible Navier-Stokes equations, expressed in the stream function formulation, take the form of the non-linear biharmonic equation

$$\Delta^2\psi + \mathrm{Re}\left[\frac{\partial\psi}{\partial x}\frac{\partial\Delta\psi}{\partial y} - \frac{\partial\psi}{\partial y}\frac{\partial\Delta\psi}{\partial x}\right] = 0, \tag{16}$$

defined in the unit square $\Omega = [0,1]\times[0,1]$. The boundary conditions are

$$\psi = 0, \qquad \vec{x} \text{ on } \partial\Omega, \tag{17}$$

and

$$\left.\frac{\partial\psi}{\partial x}\right|_{(0,y)} = 0, \quad \left.\frac{\partial\psi}{\partial x}\right|_{(1,y)} = 0, \quad \left.\frac{\partial\psi}{\partial y}\right|_{(x,0)} = 0, \quad \left.\frac{\partial\psi}{\partial y}\right|_{(x,1)} = 1. \tag{18}$$

To solve this non-linear biharmonic equation, we have used the Newton's method. Starting with an initial solution $\psi^{(0)}$, a sequence of approximate solutions $\psi^{(i)}$ are computed using the formula $\psi^{(i)} = \psi^{(i-1)} + \xi$, where the correction ξ is obtained at each iteration by solving the linearized PDE

$$\Delta^2\xi + \mathrm{Re}\left[\frac{\partial\psi^{(i-1)}}{\partial x}\frac{\partial\Delta\xi}{\partial y} - \frac{\partial\psi^{(i-1)}}{\partial y}\frac{\partial\Delta\xi}{\partial x} + \frac{\partial\xi}{\partial x}\frac{\partial\Delta\psi^{(i-1)}}{\partial y} - \frac{\partial\xi}{\partial y}\frac{\partial\Delta\psi^{(i-1)}}{\partial x}\right] = R\left(\psi^{(i-1)}\right), \tag{19}$$

with boundary conditions

$$B\xi = g(x,y) - B\psi^{(i-1)}. \tag{20}$$

In this equation, $R\left(\psi^{(i-1)}\right)$ is the residual at iteration i, given by

$$R\left(\psi^{(i-1)}\right) = \Delta^2\psi^{(i-1)} + \mathrm{Re}\left[\frac{\partial\psi^{(i-1)}}{\partial x}\frac{\partial\Delta\psi^{(i-1)}}{\partial y} - \frac{\partial\psi^{(i-1)}}{\partial y}\frac{\partial\Delta\psi^{(i-1)}}{\partial x}\right], \tag{21}$$

and B is the operator obtained after writing (17) and (18) as $B\psi = g(x,y)$. The Newton iterations continue until the tolerance $\left\|R\left(\psi^{(i-1)}\right)\right\| \leq \epsilon$ is reached.

The value of Re is increased at steps of 100. Starting at $\mathrm{Re} = 0$, the procedure above is repeated for each value of Re using as $\psi^{(0)}$ the solution from the previous Reynolds number. The algorithm stops when the selected Reynolds number has been reached.

It is well-known that this problem benefits of the use of a node distribution with a higher density of nodes close to the boundary [1, 3]. In this test problem, we have used the node distribution illustrated and described in Figure 6 (with an enlargement of the right bottom corner in the right subplot).

The derivatives in (19) are approximated using PHS r^7 augmented with all polynomial terms up to 8th order, which represents $\binom{8+2}{8} = 45$ elements. The selected stencil size is $n = 90$, which is twice as many nodes as polynomial terms. To enforce the boundary conditions (17) and (18), one layer of ghost node is included around the domain. If N_b is the number of boundary nodes, the number of ghost nodes is equal to $N_b + 8$ nodes, yielding to $2N_b + 8$ unkowns. Enforcing (17) and (18)

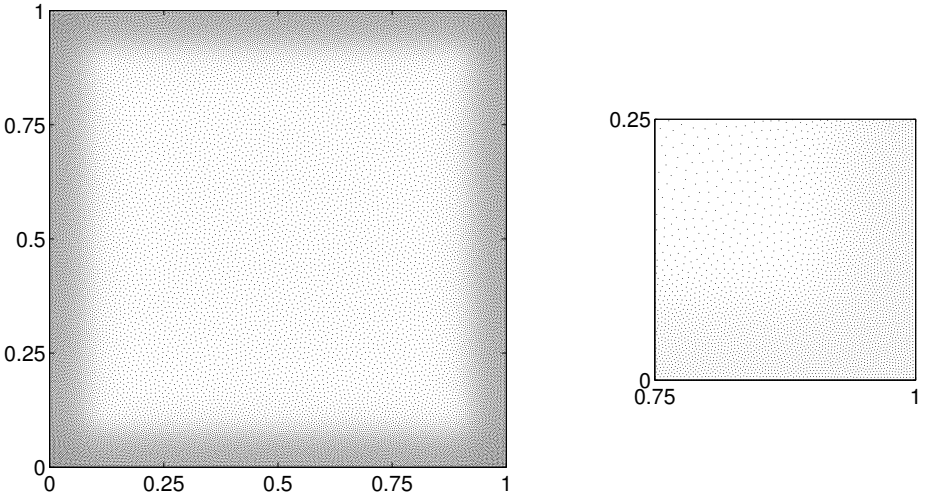

Figure 6: Discretized domain based on a node repel algorithm using $N = 39{,}389$ nodes: 36,181 interior nodes, 1,600 boundary nodes and 1,604 ghost nodes. The internodal distance increases from $h \approx 0.0025$ near boundaries to $h \approx 0.01$ in the center of the cavity.

on the boundary nodes yields to N_b and $N_b + 4$ conditions, respectively. The 4 extra conditions needed are obtained enforcing also the PDE (19) on the corner nodes.

Figure 7 shows the solution of (16) for Re = 0 (left) and Re = 1000 (right). Notice there are three vortices formed for these values: one at the center of the cavity (Primary vortex), another at the bottom-right corner (BR1) and the last one at the bottom-left corner (BL1). The stream function and vorticity values, as well as locations (x, y), are listed in Table 1. The values reported for these low Reynolds numbers are in very good agreement with those found in [1, 3].

Using the node distribution from Figure 6, accurate solutions have been computed up to Re = 10,000. The correspoding stream function and vorticity values, as well as locations (x, y), for the Primary, BR1 and BL1 vortices are listed in Table 1. The stream function for different Reynolds number are plotted in Figure 8 for the same contour values as in Figure 7. For qualitative illustration, solutions for Re = 15,000 and 20,000 are also plotted.

The present node distribution formed by $N = 39{,}389$ nodes, obtains very competitive results with respect to those found in the literature for $257 \times 257 = 66{,}049$ nodes [1, 6]. The locations of the vortices are captured quite accurately, while the strengths deteriorate as the Reynolds number increases. To accurately resolve all the features, the node distribution should be refine as the Reynolds number increases. For example, in [1] accurate results were computed up to Re = 12,500. For Re = 16,000 only qualitative results were shown, concluding that their solution at this Reynolds number was under-resolved and a finner node distribution was needed. In [6] they used a $601 \times 601 = 361{,}201$ node grid for Re = 21,000. The present numerical method is able to resolve the small vortices at the top-right corner and bottom corners shown in [6] for Re = 20,000. However, to improve the accuracy of the vortex strengths, a finner node distribution is required.

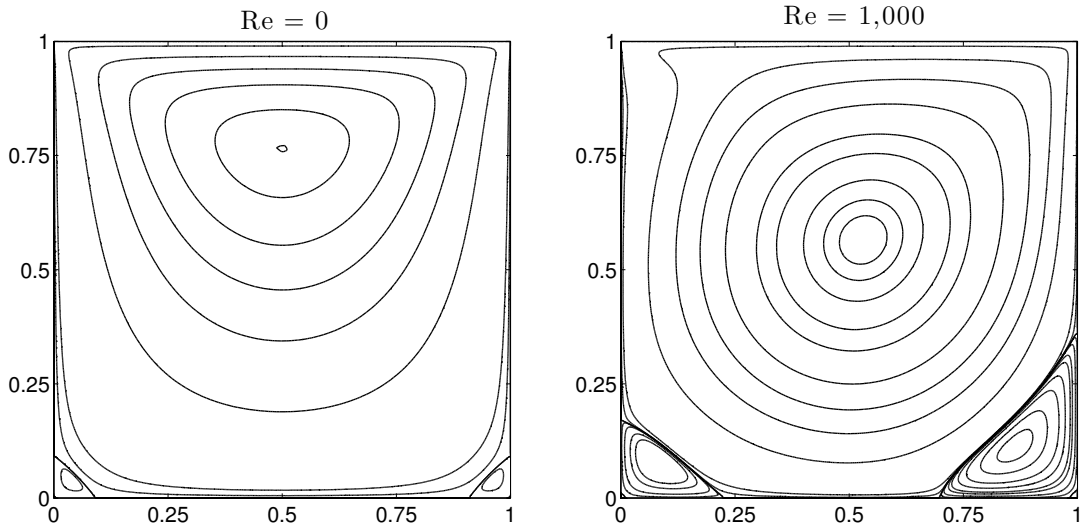

Figure 7: Stream function contours for Re = 0 (left) and Re = 1,000 (right). The contour values are set to $-0.1175, -0.115, -0.11, -0.1, -0.09, -0.07, -0.05, -0.03, -0.01, -1e-4, -1e-5, -1e-10, 0, 1e-6, 1e-5, 5e-5, 1e-4, 2.5e-4, 5e-4, 1e-3$ and $1.5e-3$.

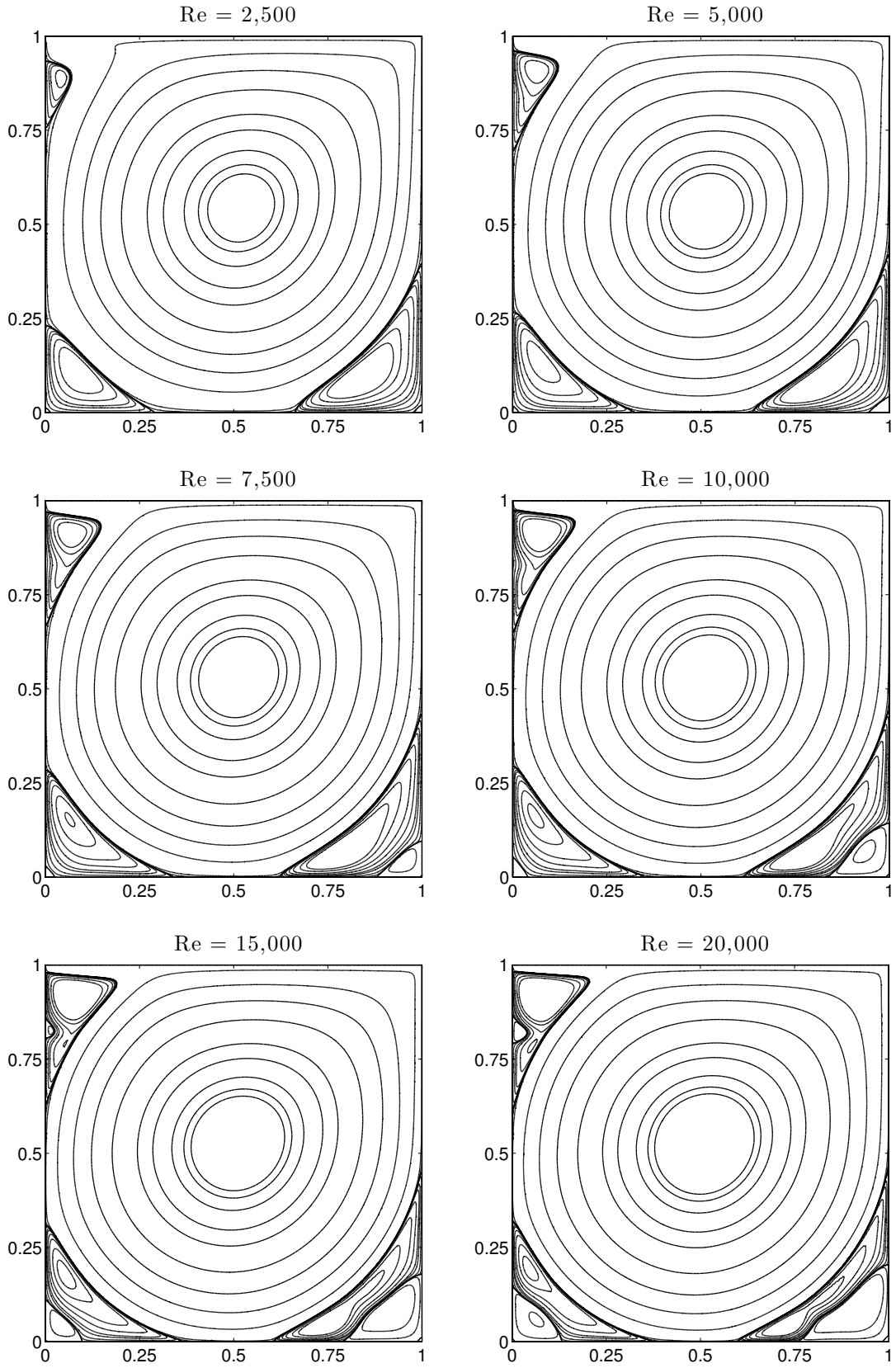

Figure 8: Stream function contours for Re = 2,500, 5,000, 7,500, 10,000, 15,000 and 20,000.

Table 1: Property of vortices: stream function and vorticity values at the vortex centers; their (x, y) locations.

	Re	0	1,000	2,500	5,000	7,500	10,000
Primary vortex	ψ	-0.1000756	-0.1189307	-0.1214544	-0.1223237	-0.1230018	-0.1237093
	ω	-3.2123303	-2.0676832	-1.9758657	-1.9419612	-1.9360192	-1.9385587
	(x, y)	(0.5000,0.7650)	(0.5308,0.5652)	(0.5198, 0.5439)	(0.5151,0.5352)	(0.5132, 0.5317)	(0.5121,0.5296)
BR1	ψ	0.2227561E-05	0.1729705E-02	0.2661659E-02	0.3077562E-02	0.3240731E-02	0.3216516E-02
	ω	0.0152408	1.1089003	1.9387076	2.7574341	3.2653721	3.8385719
	(x, y)	(0.9623,0.0381)	(0.8641, 0.1118)	(0.8344,0.0907)	(0.8046,0.0727)	(0.7908,0.0650)	(0.7770 ,0.0591)
BL1	ψ	0.2227400E-05	0.2334222E-03	0.9310293E-03	0.1379408E-02	0.1553299E-02	0.1658958E-02
	ω	0.0152618	0.3521649	0.9788872	1.5218878	1.9127837	2.2099272
	(x, y)	(0.0377,0.0381)	(0.0832, 0.0781)	(0.0843, 0.1109)	(0.0728,0.1371)	(0.0642,0.1535)	(0.0586,0.1633)

9 A 3-D spherical problem: the Global Electric Circuit

In this section, we test the performance of PHS with polynomial augmentation when solving a fully three-dimensional test problem with irregular boundaries. The selected test problem is known as the Global Electric Circuit (GEC), which is a problem from the field of atmospheric electricity solved in [2] using MQ interpolants augmented with constant terms in the limit $\varepsilon \to 0$. In this problem, the electric potential within the atmosphere $u(r, \theta, \lambda)$ is modeled by the 3-D elliptic PDE

$$-\nabla \cdot (\sigma \nabla u) = S, \qquad \text{in } \Omega \tag{22}$$

where the variable coefficient $\sigma(r, \theta, \lambda)$ represents the conductivity of the atmosphere and $S(r, \theta, \lambda)$ the source distribution. This equation is obtained by applying Ohm's law to the steady-state current continuity equation. The 3-D domain Ω is defined as $-90° \leq \theta \leq 90°, -180° \leq \lambda < 180°, k(\theta, \lambda) \leq r \leq r_b$, where the irregular boundary $k(\theta, \lambda)$ is the Earth's topography and r_b is the altitude from sea level on which the top boundary is enforced.

In our present simplified version of the problem, the variable coefficient is assumed to increase exponentially with altitude r as

$$\sigma(r) = \sigma_0 e^{r/c}, \tag{23}$$

where $c = 6$ km and $\sigma_0 = 5 \cdot 10^{-11}$ S/km. To overcome the possible ill-conditioning of the equation due to the highly variable and exponential dependance of (23), we rewrite (22) as in [2],

$$\Delta u + (\nabla \log \sigma) \cdot \nabla u = f. \tag{24}$$

Dirichlet boundary conditions are enforced on the boundary,

$$u = g(\theta, \lambda), \qquad \text{on } \partial\Omega \tag{25}$$

and the functions f and g are computed by assuming the exact solution

$$u(r, \theta, \lambda) = \left[3\, Y_9^{-7}(\theta, \lambda) + 2\, Y_6^0(\theta, \lambda)\right] \sin\left(4\pi \frac{r - r_0}{r_b - r_0}\right). \tag{26}$$

Following the same procedure than in [2], equation (24) is solved in the domain which results after the change of variable

$$r\,(\xi) = A e^{\beta(\xi - \xi_0)} + B, \tag{27}$$

where A and B are constants determined by the conditions

$$r(\xi_0) = r_0 \qquad \text{and} \qquad r(\xi_b) = r_b. \tag{28}$$

Here, r_0 is the mean radius of the Earth set to 6400 km and $r_b = 6460$ km. Notice that under this change of variable, the Earth is mapped over a sphere of radius ξ_0 and the radial coordinate is exponentially stretched. In [2], the domain is discretized using two different node layouts: scattered nodes based on a repel algorithm between the Earth's surface and ~ 10 km altitude from sea level, and a nested shell model in the rest of the domain. As a result, truly 3-D stencils are used in the first region, so that the irregularities of the boundary can be properly resolved, while a hybrid FD/RBF-FD scheme is used in the structured nested shell region.

Since the purpose of this test problem is to study the performance of 3-D scattered approximations based on PHS with polynomial augmentation, we have only solved the problem in the scattered

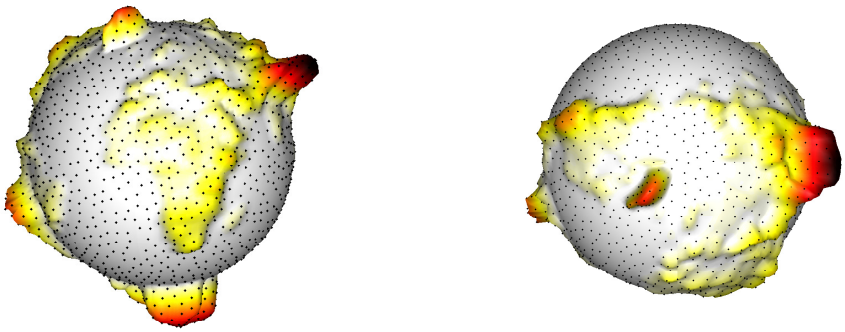

Figure 9: Equatorial view (left) and north pole view (right) of the Earth mapped on a sphere of radius $\xi_0 = 1.15$. The dots are the nodes which discretize the surface of the Earth for the 3-D node distribution $N = 33{,}389$, which is the bottom boundary of the domain Ω.

region, i.e. from the surface of the Earth to $\sim$ 12 km from sea level. Therefore, the Dirichlet boundary conditions (25) are enforced on the bottom boundary $k(\theta, \lambda)$ and the top boundary at 6412 km.

To discretize the domain, we follows a procedure very similar to the one described in Section 4. In this case, the domain is filled up using a nested shell model formed by N_r radially aligned spherical shells, where the number of nodes on each shell N_h has varied so that the horizontal internodal distances are kept constant and equal to the radial one. The nodes are redistributed using a repel algorithm. Those lying outside the domain are discarded. Figure 9 shows an equatorial view (left) and a north pole view (right) of the surface of the Earth mapped on a sphere of radius $\xi_0 = 1.15$. The dots are the nodes which discretize the bottom boundary for a 3-D node distribution with $N = 33{,}389$ nodes.

To approximate equation (24) on the node distribution, we have used PHS r^7 augmented with all polynomial terms of total degree less than or equal to 4 in 3-D, which are formed by $\binom{4+3}{4} = 35$ elements. Following the rule of thumb observed in Section 6, where it was noticed that the stencil size n should be twice the number of polynomial terms included if no ghost nodes were used, the stencil size is set to $n = 70$. Figure 10 shows the sparsity pattern after applying sparse reverse Cuthill-McKee ordering (left) and the eigenvalue distribution (right) of the resulting differentiation matrix that approximates (24). Notice that all the eigenvalues lie very close to the negative real axis. The linear system of equations is solved as described in Section 5, by using a BICGSTAB iterative method with incomplete LU factorization as preconditioner.

The ℓ_2 error and number of BICGSTAB iterations are listed in Table 2 as a function of the number of nodes N when solving this PDE problem. These results are plotted in Figure 11. Notice that the convergence is $O(h^4)$, while the number of iterations increases linearly as a function of $\sqrt[3]{N}$. This is a very interesting result, since it shows that it is possible to achieve forth order convergence in 3-D with a relatively small stencil formed by 70 nodes. We note also that 4th order looks like the optimal order in terms of accuracy and computational cost. Achieving 6th order convergence would requiere including up to $\binom{6+3}{6} = 84$ polynomial terms, which results in a stencil formed by $n = 168$ nodes. Hence, the computational cost of computing the differentiation matrices would be

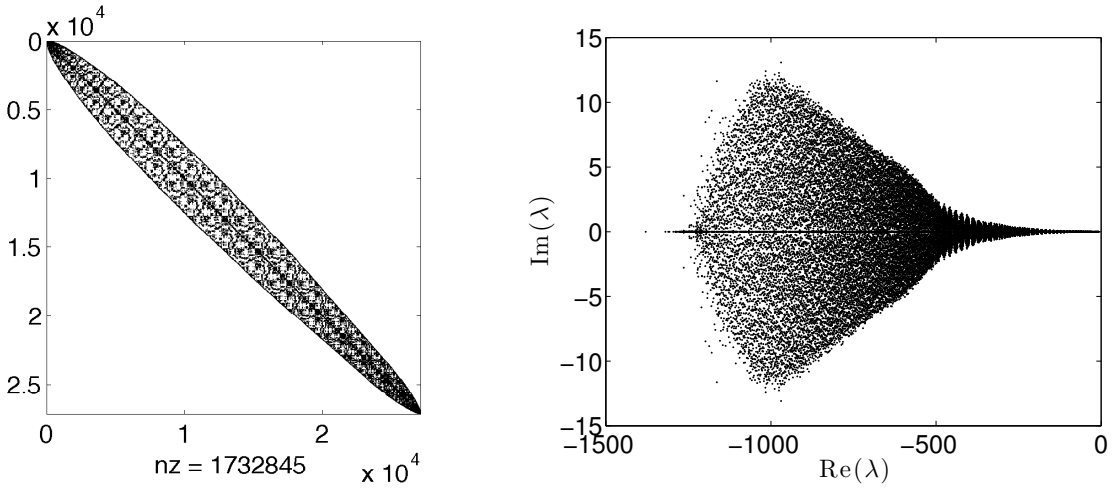

Figure 10: Sparsity pattern after applying sparse reverse Cuthill-McKee ordering (left) and eigenvalue distribution (right) of the differentiation matrix that approximates (24) using PHS r^7 augmented with polynomials up to 4th order and $n = 70$, for $N = 33{,}389$. Note here the different scales on the two axes of the right plot.

Table 2: The specifications for the results shown in Figure 11

N	ℓ_2 error	# iterations
33,389	$1.4{\cdot}10^{-2}$	10
64,048	$8.5{\cdot}10^{-3}$	12
109,455	$4.8{\cdot}10^{-3}$	15.5
255,272	$1.6{\cdot}10^{-3}$	20.5
494,010	$6.4{\cdot}10^{-4}$	25
1,076,812	$2.1{\cdot}10^{-4}$	30.5
1,997,495	$8.6{\cdot}10^{-5}$	38.5

about 14 times higher.

10 Conclusions

A novel RBF-FD method based on Polyharmonic Splines (PHS) with polynomial augmentation is considered in this paper for the numerical solution of elliptic PDE problems. In this work, we have proved that RBF-FD with polynomial augmentation is indeed a equality-constrained quadratic programming problem, where the RBF-FD weights are the solution of the under-determined problem $P^T \underline{w} = L\underline{p}$ minimizing a certain functional.

Several test problems are solved to assess the performance of the method. Through these experiments, we have observed that the convergence order of the method is only given by the polynomial degree while the stencil size mostly influences a multiplying factor. The stencil size also seems to have a fundamental role in the stability of the method. For example, we have observed that for problems where no method is used to handle the boundary, the stencil size should be at least twice the number of polynomial terms for the numerical method to be stable. Otherwise, if no boundaries

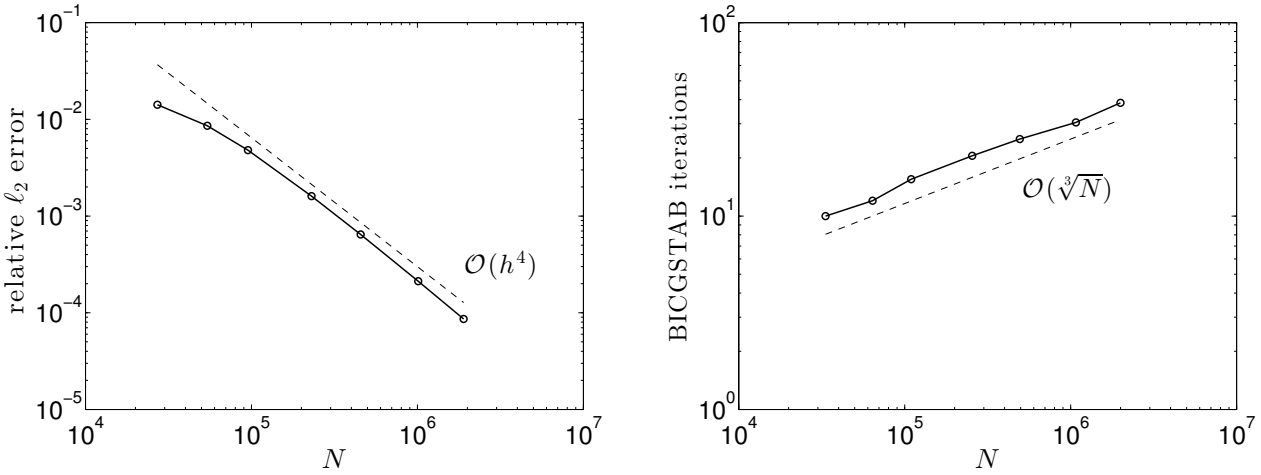

Figure 11: The relative ℓ_2 error (left) and the number of iterations (right) as a function of N.

are present or if a ghost node approach is used to handle the boundary, it sufficies if the stencil size is slighly larger than the number of polynomial terms.

In general, this method shows to be very competitive, having all the advantages of RBF-FD methods (high-order convergence, allowing for local node refinement, geometric flexibility, easy to program, etc) while bypasses the main drawbacks, such as the selection of a shape parameter, ill-conditioning and stagnation errors.

Acknowledgments: This work was supported by NSF Awards DMS-094581. The National Center for Atmospheric Research is sponsored by the NSF.

References

[1] E. Barragy and G.F. Carey, *Stream function-vorticity driven cavity solution using p finite elements*, Computers & Fluids **26** (1997), no. 5, 453–468.

[2] V. Bayona, N. Flyer, G.M. Lucas, and A.J.G. Baumgaertner, *A 3-D RBF-FD elliptic solver for irregular boundaries: modeling the atmospheric global electric circuit with topography*, Geoscientific Model Development Discussions **8** (2015), no. 4, 3523–3564.

[3] O. Botella and R. Peyret, *Benchmark spectral results on the lid-driven cavity flow*, Computers & Fluids **27** (1998), no. 4, 421–433.

[4] P. P. Chinchapatnam, K. Djidjeli, P. B. Nair, and M. Tan, *A compact RBF-FD based meshless method for the incompressible Navier-Stokes equations*, J. Eng. Maritime Env. **223** (2009), 275–290.

[5] T.A. Driscoll and B. Fornberg, *Block pseudospectral methods for Maxwell's equations II: Two-dimensional, discontinuous-coefficient case*, SIAM Journal on Scientific Computing **21** (1999), no. 3, 1146–1167.

[6] E. Erturk, T.C. Corke, and C. Gökçöl, *Numerical solutions of 2-D steady incompressible driven cavity flow at high Reynolds numbers*, International Journal for Numerical Methods in Fluids **48** (2005), no. 7, 747–774.

[7] G. E. Fasshauer, *Meshfree Approximation Methods with MATLAB*, Interdisciplinary Mathematical Sciences - Vol. 6, World Scientific Publishers, Singapore, 2007.

[8] A. I. Fedoseyev, M. J. Friedman, and E. J. Kansa, *Improved multiquadric method for elliptic partial differential equations via PDE collocation on the boundary*, Comput. Math. Appl. **43** (2002), 439–455.

[9] N. Flyer, B. Fornberg, G. Barnett, and V. Bayona, *RBF-FD approximations based on PHS with polynomial augmentation: I. Interpolation*, in preparation.

[10] N. Flyer, G. B. Wright, and B. Fornberg, *Radial basis function-generated finite differences: A mesh-free method for computational geosciences*, Handbook of Geomathematics (Berlin) (W. Freeden, M.Z. Nashed, and T. Sonar, eds.), Springer Verlag, 2014.

[11] B. Fornberg and N. Flyer, *A Primer on Radial Basis Functions with Applications to the Geosciences*, SIAM, Philadelphia, 2015.

[12] B. Fornberg and N. Flyer, *Solving PDEs with radial basis functions*, Acta Numerica **24** (2015), 215–258.

[13] U.K.N.G. Ghia, K.N. Ghia, and C.T. Shin, *High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method*, Journal of computational physics **48** (1982), no. 3, 387–411.

[14] A. Iske, *On the approximation order and numerical stability of local Lagrange interpolation by polyharmonic splines*, Modern Developments in Multivariate Approximation (W. Haussmann, K. Jetter, M. Reimer, and J. Stöckler, eds.), International Series of Numerical Mathematics, vol. 145, Birkhäuser Verlag, Basel, 2003, pp. 153–165.

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

[16] M. Li, T. Tang, and B. Fornberg, *A compact fourth-order finite difference scheme for the steady incompressible Navier-Stokes equations*, International Journal for Numerical Methods in Fluids **20** (1995), no. 10, 1137–1151.

[17] Y.V.S.S. Sanyasiraju and G. Chandhini, *Local radial basis function based gridfree scheme for unsteady incompressible viscous flows*, Journal of Computational Physics **227** (2008), no. 20, 8922–8948.

[18] G.D. Smith, *Numerical Solution of Partial Differential Equations*, Oxford mathematical handbooks, Oxford University Press, 1965.

Appendix C

Manuscript 3. An RBF-FD polynomial method based on polyharmonic splines for the Navier-Stokes equations: Comparisons on different node layouts

An RBF-FD polynomial method based on polyharmonic splines for the Navier-Stokes equations: Comparisons on different node layouts

Gregory A. Barnett
University of Colorado
Department of Applied Mathematics
Boulder, CO 80309 USA
gregory.barnett@colorado.edu

Natasha Flyer
National Center for Atmospheric Research
Institute for Mathematics Applied to Geosciences
Boulder, CO 80305 USA
flyer@ucar.edu*

Louis J. Wicker
NOAA National Severe Storms Laboratory
Norman, OK 73072 USA
louis.wicker@noaa.gov

June 22, 2015

Abstract

Polyharmonic spline (PHS) radial basis functions (RBFs) are used together with polynomials to create local RBF-finite-difference (RBF-FD) weights on different node layouts for spatial discretization of the compressible Navier-Stokes equations at low Mach number, relevant to atmospheric flows. Test cases are taken from the numerical weather prediction community and solved on bounded domains. Thus, attention is given on how to handle boundaries with the RBF-FD method, as well as a novel implementation for the presented approach. Comparisons are done on Cartesian, hexagonal, and quasi-uniformly scattered node layouts. Since RBFs are independent of a coordinate system (and only depend on the distance between nodes), changing the node layout amounts to changing one line of code. In addition, consideration and guidelines are given on PHS order, polynomial degree and stencil size. The main advantages of the present method are: 1) capturing the basic physics of the problem surprisingly well, even at very coarse resolutions, 2) high-order accuracy without the need of tuning a shape parameter, and 3) the inclusion of polynomials eliminates stagnation (saturation) errors.

1 Introduction

In applications of radial basis functions (RBFs) for fluid modeling (both incompressible and compressible), infinitely smooth RBFs have traditionally been used due to their spectral convergence

*Corresponding author

properties, with multiquadrics and Gaussians being the most popular [34, 4, 5, 7, 10, 13, 15, 32]. However, fluid flows in nature can exhibit complex rapidly developing features with such steep gradients that spectral accuracy can not be realized on resolutions that are observable or practical. This study offers a different perspective on RBF-based fluid modeling with the following aspects: 1) using odd-ordered polyharmonic spline (PHS) RBF, r^{2m+1} for $m \in \mathbb{N}$ (r is the Euclidean distance between where the RBF is centered and where is evaluated), and 2) in conjunction with higher-order polynomials (degree ≥ 4). From a historical perspective, using this combination for RBF-FD has not been considered, most likely for the following reasons:

1. Before the development of RBF-FD or other flavors of local RBFs [32, 33, 37], applications of RBFs were global. Then, *if* piecewise smooth RBFs were used, they were used in conjunction with low order polynomials, e.g $1, x, y$ in 2-D. The only role of the polynomial was to guarantee non-singularity of the RBF interpolation matrix, which needs to be inverted to derive the differentiation matrices [41, 11]. The role of capturing the physics of complicated fluid flows was the left to the RBFs.

2. Even when used in a global sense, these RBFs were not as popular as infinitely smooth RBFs. For example, r^3, results in an RBF that jumps in the third derivative, giving at best fourth-order accuracy in 1-D (with the order of convergence increasing as the dimension increases (c.f. [28]) assuming smooth data). The curse lies in the fact that as m increases, leading to a smoother RBF, the condition number of the interpolation matrix gravely increases. Thus, in the past, one was limited to keeping m small and having low algebraic accuracy.

3. Lastly, using polynomials on a global scale can be dangerous, since it can lead to Runge phenomena near the boundaries. In contrast, on a local scale as in the RBF-FD method, one is only interested in the approximation at the center of the stencil and not at the edges.

As a result, a new way to use PHS RBFs combined with polynomials in the context of RBF-FD is introduced, such that high-order accuracy is gained with excellent conditioning of RBF-FD interpolation matrix and no saturation error is encountered. Furthermore, there is no need to bother with selecting an optimal shape parameter, which plays an important role in the accuracy of the solution when using infinitely smooth RBFs [12, 1, 8, 31, 6]). We will demonstrate the performance of the modified RBF-FD method for 1) the advection of a scalar in a strong shear flow (a hyperbolic PDE introduced by [24] and popularized by [2]) and 2) the 2D nonhydrostatic compressible Navier-Stokes on bounded domains applied to test cases common in the numerical weather prediction community. Although already broad in scope, the authors further wish to classify the differences, if any, that occur in applying this methodology on different node layouts: 1) Cartesian, 2) hexagonal, and 3) scattered. The rational being that classical finite difference methods, based on polynomials, are usually implemented on Cartesian lattices; hexagonal layouts are optimal in terms of node packing in 2D, supplying information along 3 distinct directions in contrast to Cartesian layouts where information is aligned only along 2 directions; and scattered nodes allow for geometric flexibility of the domain and the ability of node refinement.

The paper is organized as follows: Section 2 very briefly introduces RBFs. Section 3 discusses the calculation of RBF-FD weights using polynomials. Section 4 demonstrates how the inclusion of polynomials with PHS eliminates stagnation (saturation) errors. Section 5 discusses the node sets that are used and how boundaries and hyperviscosity are handled. Section 6 applies the methodology on the various node layouts, giving detailed results from test cases that are standard in the numerical weather prediction community. Lastly, Section 7 summarizes the observations of this paper.

2 A brief introduction to Radial Basis Functions

An RBF is a d-dimensional radially symmetric function $\phi : \mathbb{R}^d \to \mathbb{R}$ that depends only on the Euclidean distance between where the RBF is centered, $\mathbf{x}_c$ and where it is evaluated, $\mathbf{x}$. That is, regardless of dimension, its argument is always a scalar defined by $r = \|\mathbf{x}-\mathbf{x}_c\|_2$, where $\|\cdot\|_2$ denotes the Euclidean distance. RBFs come in two flavors: 1) Piecewise smooth and 2) infinitely smooth. The former features a jump in some derivative and thus can only lead to algebraic convergence. The latter is a C^∞ function and can lead to spectral convergence when the data is sufficiently smooth. Only PHS RBFs do not depend on a parameter ε that controls the shape of the RBF (which influences both the conditioning of the matrices and the accuracy of the results [39, 30]). This last comment is a strong motivation of this paper since, by using PHS RBF, one avoids the difficulty of dealing with a shape parameter and yet can achieve high-order accuracy. For theoretical aspects of PHS (a class of conditionally positive definite radial functions), see [11, 41]. Common RBFs of both categories are given in Table 1, where $\mathbf{x}_c$ represents where the RBF is centered.

Table 1: Some common choices for radial functions $\phi(r)$

Piecewise smooth RBFs	$\phi(r = \|\mathbf{x} - \mathbf{x}_c\|_2)$
Polyharmonic Splines (PHS) [9]	$r^{2m} \log r,\ m \in \mathbb{N}$ $r^{2m+1},\ m \in \mathbb{N}^0$
Matern [25]	$\frac{2^{1-m}}{\Gamma(m)} r^m K_m(\varepsilon r), m > 0,$ (Bessel K-function)
Compact support ('Wendland' [40])	$(1-\varepsilon r)_+^m p(\varepsilon r)$, p certain polynomials, $m \in N$
Infinitely smooth RBFs	
Gaussian (GA)	$e^{-(\varepsilon r)^2}$
Multiquadric (MQ)	$\sqrt{1+(\varepsilon r)^2}$
Inverse Multiquadric (IMQ)	$1 \Big/ \sqrt{1+(\varepsilon r)^2}$
Inverse Quadratic (IQ)	$1 \Big/ \left(1+(\varepsilon r)^2\right)$

3 Calculation of the PHS RBF-FD differentiation weights with polynomials

The differentiation weights are derived so that the resulting linear system becomes exact for all RBF interpolants $s(\mathbf{x})$ of the form $s(\mathbf{x}) = \sum_{i=1}^n \lambda_i \phi(\|\mathbf{x} - \mathbf{x}_i\|_2) + \{p_l(\mathbf{x})\}$ with the constraints $\sum_{i=1}^n \lambda_i p_l(\mathbf{x_i}) = 0$, where $p_l(\mathbf{x})$ are all polynomials up to degree l in the dimension of the problem. These constraints enforce that the RBF basis reproduces polynomials up to degree l as well as ensure that the far-field RBF expansion is regularized (i.e. does not blow up) [17]. They also are known as the vanishing moment conditions [23].

It can then be shown (see Section 5.1.4 in [18]) that the above leads to the following linear system for the differentiation weights,

$$\left[\begin{array}{ccc|ccc} \|\mathbf{x}_1-\mathbf{x}_1\|_2^{2m+1} & \cdots & \|\mathbf{x}_1-\mathbf{x}_n\|_2^{2m+1} & 1 & x_1 & y_1 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \|\mathbf{x}_n-\mathbf{x}_1\|_2^{2m+1} & \cdots & \|\mathbf{x}_n-\mathbf{x}_n\|_2^{2m+1} & 1 & x_n & y_n \\ \hline 1 & \cdots & 1 & 0 & 0 & 0 \\ x_1 & \cdots & x_n & 0 & 0 & 0 \\ y_1 & \cdots & y_n & 0 & 0 & 0 \end{array}\right] \left[\begin{array}{c} w_1 \\ \vdots \\ w_n \\ \hline w_{n+1} \\ w_{n+2} \\ w_{n+3} \end{array}\right] = \left[\begin{array}{c} L\|\mathbf{x}-\mathbf{x}_1\|_2^{2m+1}\big|_{\mathbf{x}=\mathbf{x}_c} \\ \vdots \\ L\|\mathbf{x}-\mathbf{x}_n\|_2^{2m+1}\big|_{\mathbf{x}=\mathbf{x}_c} \\ \hline L1|_{\mathbf{x}=\mathbf{x}_c} \\ Lx|_{\mathbf{x}=\mathbf{x}_c} \\ Ly|_{\mathbf{x}=\mathbf{x}_c} \end{array}\right] \tag{1}$$

where for simple illustration purposes, we have included only up linear polynomials. The weights w_{n+1} to w_{n+3} are ignored after the matrix is inverted. Solving (1) will give one row of the differentiation matrix (DM) that contains the weights for approximating L at x_c. Thus this process is repeated N times over all nodes in the domain, giving a preprocessing cost of $O(n^3N)$. Since $n << N$, it should be noted that the DM usually becomes over 99% empty. As a result, we do not actually store the DM but only its nonzero entries.

4 Stagnation error, PHS order, and polynomial degree

Stagnation (saturation) error is defined as the convergence either stagnating or increasing as resolution increases. For infinitely smooth RBFs, as the resolution increases (i.e. r decreases), the shape parameter must increase to maintain the condition number of the matrix in (1), resulting in stagnation error since the more peaked the infinitely smooth RBFs become as ε increases, the less accurate the approximation. In contrast, PHS RBF *with* polynomials can achieve high-order algebraic convergence without encountering saturation error or the difficulty of finding an optimal value of the shape parameter ε for good accuracy, which has been a central focus of quite a few studies [16, 22, 12, 1, 8]. It should be noted that if polynomials are not included with the PHS RBF, stagnation error is encountered since boundary errors at the edge of the RBF-FD stencil can be quite large and penetrate toward the center of the stencil where the interpolant or any derivative is being approximated. Further investigation of the effects of adding polynomials to RBF-FD approximations for both infinitely smooth and PHS RBFs is given in [14].

Since locally all smooth functions are well represented by Taylor expansions, then under refinement, the RBF-FD approximation must reproduce polynomial behavior. In the following numerical studies, it indeed was found that the convergence rate is dictated not by the order of the PHS, m, but by the highest degree of polynomials, l, used. In addition, for PDEs with only first-order spatial derivatives, as in all the test cases, the convergence rate can be expected to be $O(h^l)$. The reason being is that the error for polynomial interpolation is $O(h^{l+1})$, but one order in h is lost in approximating a first derivative. These observations are in excellent agreement with Figure 1, where d/dx of the smooth function $f(x,y) = 1 + \sin(4x) + \cos(3x) + \sin(2y)$ is approximated with two different PHS RBFs, r^3 and r^7, augmented with polynomials (e.g. poly 3 is augmentation of the matrix in (1) with the 10 polynomials up to degree 3 in 2D). The approximation is at the center of 37 node hexagonal stencil with the evaluation nodes of the derivative in its vicinity. The slopes in the two panels of Figure 1 are identical, the only difference being that the constant that multiplies the order of convergence is slightly smaller for r^7, thus giving a marginally higher accuracy for a given resolution. It is still important to note that the RBFs play a crucial role in safety against singularities due to particular node layouts.

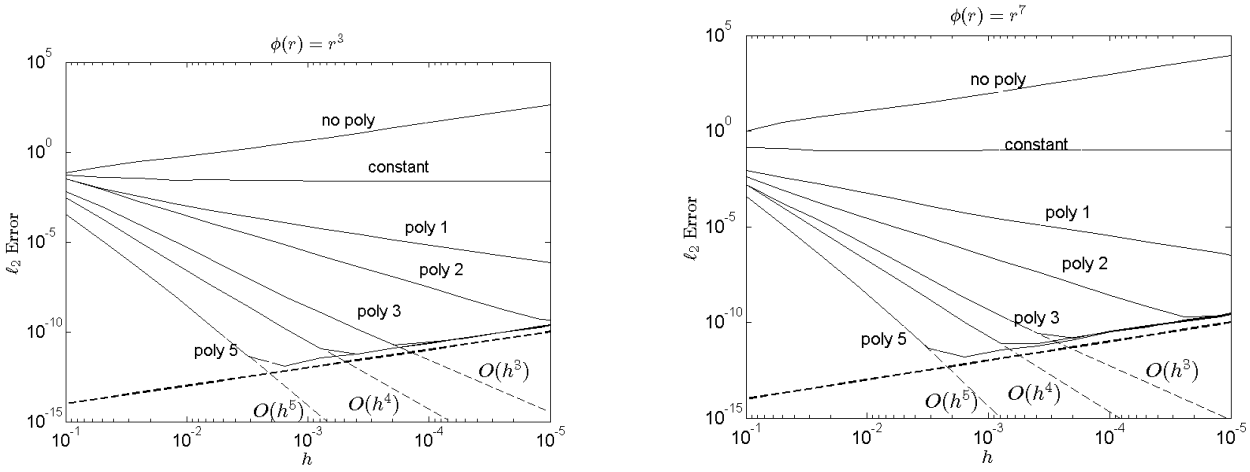

Figure 1: The convergence rate in approximating d/dx of the function $f(x, y) = 1 + \sin(4x) + \cos(3x) + \sin(2y)$ on a 37 node RBF-FD stencil based on r^3 and r^7, augmented with polynomials as described in the text. The dashed line marks machine round-off errors in standard double precision of $10^{-15}/h$ for approximating the first derivative.

5 Node sets, Ghost nodes, and Hyperviscosity

Section 5.1 overviews the various node sets considered in this study. Section 5.2 discusses how to increase accuracy near boundaries with the use of ghost nodes and Section 5.3 discusses the need for and type of hyperviscosity used, introducing a novel way of implementing hyperviscosity with PHS and polynomials.

5.1 Node-sets

Unlike traditional methods, RBF-FD has the advantage of being equally simple to apply on any set of nodes. Figure 2 shows the three types of node-layouts that will be considered in the present study. First, we consider Cartesian since they are the lattices classical finite differences (FD) are usually implemented on. Next, it is well known that hexagonal node sets are the optimal packing strategy in 2D (i.e. for a fixed area, one can fit the most number of nodes). They have also been considered an optimal layout for differentiation stencils, since information is aligned along three different directions as opposed to only two with Cartesian layouts. Although FD have been sporadically implemented over the decades on hexagonal node layouts, they have never caught favor do to the complexity of the implementation. However, now with RBF-FD, implementation is simple. Lastly, scattered node layouts are considered as they have the great advantage of geometric flexibility that will be needed in future applications of RBF-FD on irregular domains and/or with local node refinement.

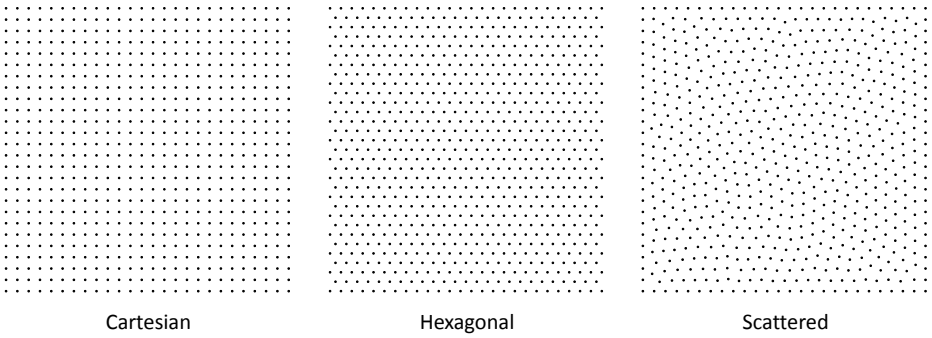

Figure 2: Three different types of node-distributions that are used to solve the 2D test problems.

5.2 Ghost nodes

Near boundaries, stencils become one-sided, leading to a deterioration of the approximation due to Runge phenomenon. In order to ameliorate this effect, one layer of ghost nodes can be placed just outside the domain. Once the ghost nodes are placed, function values at these locations can be solved for by enforcing additional constraints at the boundary nodes. For example, if the upper boundary of a rectangular domain is free-slip, then $\partial u/\partial z = 0$, and this condition can be used to solve for values of u at the ghost nodes. Similarly, the PDE itself can be enforced on the boundary, giving an extra constraint.

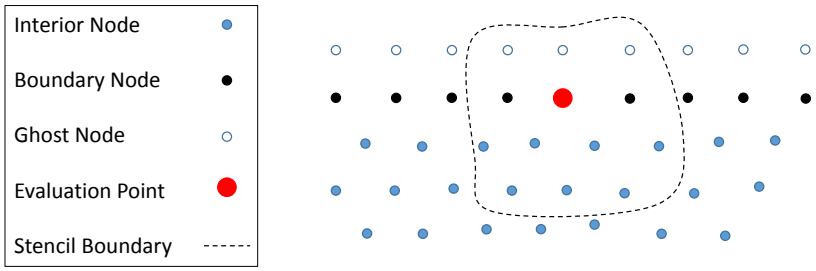

Figure 3: An example of an RBF-FD stencil that might be used for enforcing $\partial u/\partial z = 0$ on the top boundary.

For each node on the top boundary, one ghost node is placed just outside the boundary, as shown in Figure 3. The function values for u at the ghost nodes are obtained by enforcing $\partial u/\partial z = 0$ at each of the top boundary nodes simultaneously. This will lead to a coupled system with as many equations as there are ghost nodes. In the case illustrated in Figure 3, the system will be tridiagonal, as there are three unknown values at the ghost points for each evaluation node on the boundary.

The following is a more detailed discussion on how the ghost node values are calculated. Suppose there are N_I interior nodes, N_B boundary nodes, and N_G ghost nodes, and let the total number of nodes be $N = N_I + N_B + N_G$. For each top boundary node, approximate the differentiation weights for $\partial/\partial z$ as given in Section 3. This will result in a sparse $N_B \times N$ DM (here, called W). Thus,

$$\frac{\partial u}{\partial z} = 0 \quad \text{is approximated by}$$
$$W\underline{u} = \underline{0}.$$

The function values $\underline{u}$ are organized according to where they are located:

$$\begin{aligned}\underline{u}_I &: \text{function values at interior nodes}\\ \underline{u}_B &: \text{function values at boundary nodes}\\ \underline{u}_G &: \text{function values at ghost nodes}\end{aligned}$$

Then, the condition $W\underline{u} \approx \underline{0}$ can be written as

$$W_I\underline{u}_I + W_B\underline{u}_B + W_G\underline{u}_G = \underline{0}, \tag{2}$$

where the matrix W has similarly been split into pieces according to the three different types of nodes:

$$\begin{aligned}W_I\ (N_B \times N_I) &: \text{weights applied on interior nodes}\\ W_B\ (N_B \times N_B) &: \text{weights applied on boundary nodes}\\ W_G\ (N_B \times N_G) &: \text{weights applied on ghost nodes}\end{aligned}$$

Finally, (2) is used to solve for $\underline{u}_G$, the function values at the ghost nodes:

$$\underline{u}_G = -W_G^{-1}\left(W_I\underline{u}_I + W_B\underline{u}_B\right).$$

Once the function values at the ghost node are known, they can be used for the approximation of other derivatives that appear in the governing equations.

5.3 Hyperviscosity with PHS and polynomials

When the viscosity of the fluid μ is small (such as the case with air $\approx 10^{-5}m^2/s$), there is essentially no natural diffusion in the governing equations, and high-frequency errors will grow to dominate a numerical solution. To achieve time stability with the RBF-FD method, it has been shown that adding a relatively small amount of hyperviscosity to the right-hand-side of the governing equations eliminates the contaminating high-frequency noise while keeping the numerically relevant portion of the solution intact [19, 15, 3].

The hyperviscosity operator takes the form $\gamma\Delta^k$, where k is the power the Laplacian and γ is a scaling parameter. The integer k controls which frequencies are most affected, with larger values of k giving stronger damping of high frequencies and weaker damping of low frequencies. As has been shown in [19, 15, 3], for good stability and accuracy, the parameter γ is directly proportional to the number of nodes in the domain N or conversely the resolution h, as well as k. Thus, for a square-type domain in 2D $h \sim 1/\sqrt{N}$, and $\gamma = ch^{-2k}$, where c is a constant that is generally set for the problem at hand and independent of the resolution h, (e.g. for the NS test cases $c = 2^{-6}$ regardless of the resolution or node layout used).

The hyperviscosity operator Δ^k is particularly simple to apply to an odd-powered PHS RBF, regardless of the spatial dimension. The Laplace operator in d dimensions for a radially symmetric function is given by $\Delta = \partial^2/\partial r^2 + ((d-1)/r)\partial/\partial r$. Apply this to $\phi(r) = r^m$ results in

$$\Delta\left(\|\mathbf{x}\|^m\right) = m\left[m + (d-2)\right]\|\mathbf{x}\|^{m-2}. \tag{3}$$

In other words, applying the Laplace operator to a PHS RBF of degree m gives a new PHS RBF of degree $m-2$. Using the above relationship, one can evaluate $\Delta^k\left(\|\mathbf{x}\|^m\right)$, and the new RBF will be continuous provided that $(m \geq 2k+1)$. Higher-order Laplacians can be implemented by applying (3) repeatedly. It should noted that the order of the PHS used for spatial discretization need not be that used for hyperviscosity. However, it was found experimentally that the simplest approach for the needed inclusion of polynomials was to use up to the same degree as that for discretization.

6 Numerical Studies

The first test case, inviscid transport of a scalar variable in a strongly sheared vortex flow, is a case of pure advection with a known analytical solution, so that the convergence properties of the method can be tested. It was originally proposed in [24] and then considered in the context of applying limiters in [2, 35]. The second set of tests is based on the work presented in [38], where a cold descending bubble in a neutrally-stratified atmosphere develops into a traveling density current with the formation of Kelvin-Helmholtz rotors. It is now considered a classic test case in nonhydrostatic atmospheric modeling. The third test [29] (with similar studies in [21]) simulates a rising thermal air bubble and nicely illustrates how instability patterns at the leading edge of the thermal are dependent on the node layout when the the dynamic viscosity is that of air.

In all cases, $n = 37$ node stencils are used for spatial discretization, since both Cartesian and hexagonal layouts have perfectly symmetric stencils at that number, as seen in Appendix A. Although not essential, stencil symmetry is beneficial in that it it provides information evenly for approximating an operator at the center of the stencil. In all cases, time stepping is done with a 4th-order Runge-Kutta scheme (RK4).

6.1 Advective transport of a scalar variable

In this test, a circular scalar field is stretched and deformed into a crescent by vortex-like velocity field that then reverses and returns it back to its original position and shape. The governing equation is defined on $[0, 1] \times [0, 1]$ in x and y and given by

$$\frac{\partial \psi}{\partial t} = -\frac{\partial}{\partial x}\left(u\psi\right) - \frac{\partial}{\partial y}\left(v\psi\right).$$

The scalar ψ is advected by the following divergence-free velocity field

$$u\left(x, y, t\right) = u_\theta\left(r, t\right)\sin\theta, \qquad v\left(x, y, t\right) = -u_\theta\left(r, t\right)\cos\theta.$$

with period T, where

$$u_\theta\left(r, t\right) = \frac{4\pi r}{T}\left[1 - \cos\left(\frac{2\pi t}{T}\right)\frac{1 - (4r)^6}{1 + (4r)^6}\right],$$

and

$$r = \sqrt{\left(x - 0.5\right)^2 + \left(y - 0.5\right)^2}, \qquad \theta = \tan^{-1}\left(\frac{y - 0.5}{x - 0.5}\right).$$

In order to create a test problem with no boundary effects, the nodes on the interior of the domain near the boundary, as well as the function values associated with them, are simply reflected over the boundary (thus no boundary conditions are needed), forming perfectly symmetric boundary stencils - half of which are then ghost nodes and half interior nodes. The boundary is then time stepped with the rest of the interior of the domain. An example of a symmetric 19 node boundary stencil is given in Figure 4.

The initial condition for ψ is a cosine bell

$$\psi|_{t=0} = \begin{cases} \frac{1+\cos(\pi\tilde{r})}{2} & \tilde{r} \leq 1 \\ 0 & \tilde{r} > 1, \end{cases} \tag{4}$$

where $\tilde{r} = 5\sqrt{\left(x - 0.3\right)^2 + \left(y - 0.5\right)^2}$. Using a classic Runge-Kutta 4th-order (RK4), ψ is advanced in time from $t = 0$ until $t = T$ (one period), at which point it should have ideally returned to its original height and position.

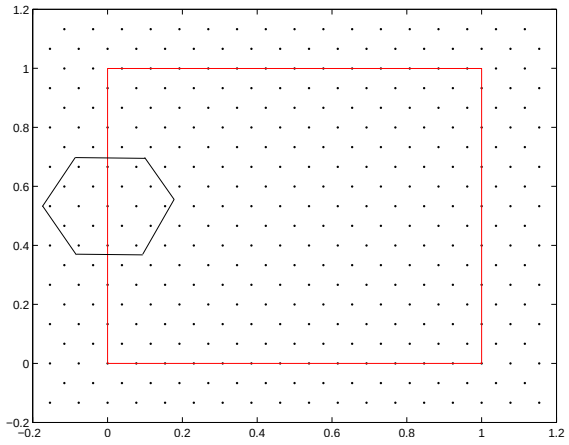

Figure 4: An illustration of a hexagonal 19 node boundary stencil in which the nodes and there associated function values are reflected outside the domain. This setup is used for the advection of a scalar transport test case.

6.1.1 Accuracy of solution, convergence, and effect of different node layouts

Figure 5 shows the time evolution for solution (assuming a period of $T = 1$), with corresponding velocity field, on a hexagonal node layout. The specifications of the resolution, time-step, basis functions and hyperviscosity used are given in the caption of the figure. Although, this is a high resolution case with a total of $N = 40401$ nodes in the domain or a node spacing of $h = 0.005$, it should be noted that the maximum amplitude of ψ has only increased by 0.07% and has only gone below zero by -0.08%, as can be seen in Figure 5a. In fact, when the scalar field is in its state of highest deformation at $t = 0.25, 0.75$, the error in the maximum amplitude is not more than 0.001. Furthermore, the CFL criterion for an RK4 stability domain dictates a time step of $\Delta t < \Delta x \cdot 2\sqrt{2}/(\max \mathbf{u})$, which for this case translates into $\Delta t < 7.0(10)^{-4}$, a factor of only 2.12 larger than the time step taken of $\Delta t = h/15 = 0.005/15 = 3.3(10)^{-4}$.

In order to observe the long-time errors in the method, the solution is advanced for 100 periods as shown in the left panel of Figure 6. Even after so many revolutions, the height of the scalar field has decreased only by 4%, with a slight distortion from its circular shape. The right panel of Figure 6 shows a dispersive error pattern with a maximum value of 0.13. The ℓ_2 and ℓ_∞ errors are 0.125 and 0.141, respectively.

For a 37 node stencil, the highest degree polynomials that can be used on all three node sets is fifth degree. Both hexagonal and scattered node sets can handle sixth degree polynomials but not Cartesian layouts. This is because on such a lattice the nodes approach non-unisolvency, resulting in the column vectors of the polynomial portion of the matrix in (1) becoming linearly dependent. Also, in order to demonstrate that the polynomial degree controls the convergence and not the PHS order, r^3 PHS are now used instead of r^9 with up to fifth-degree polynomials on Cartesian, hexagonal and scattered nodes. Figure 7 illustrates this for three resolutions, $h = 0.02, 0.01, 0.005$. For any given resolution, all nodes sets perform roughly the same, both with regard to the minimum and maximum function values and the errors in the ℓ_2 and ℓ_∞ norm. Comparing the the maximum and minimum of the solution for hexagonal nodes on $h = 0.005$ between Figures 7 and 5, it can been seen that using r^9 gives slightly better accuracy. This phenomena was noted in Section 4. Figure 8 shows the convergence rate corresponding to the cases given in Figure 7. From the discussion in Section 4, the convergence rate should be $O(h^l)$, where l is the highest degree of polynomials used,

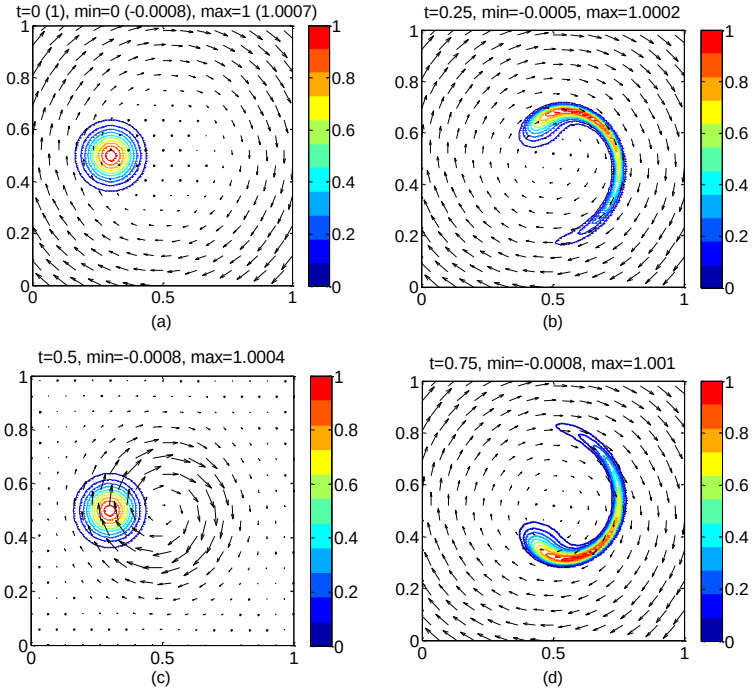

Figure 5: Time series of the solution for ψ at: (a) $t = 0(1)$, (b) $t = 0.25$, (c) $t = 0.5$, (d) $t = 0.75$, with the corresponding minimum and maximum values at each time. Contour lines are in intervals of 0.05. An $n = 37$ node stencil with r^9 PHS and up to 4^{th}-order polynomials on a hexagonal node layout of 201 by 201 is used. The time-step is h/15, where here $h = 1/200 = 0.005$. A hyperviscosity of $-2^{-14}\Delta^4$ is also implemented.

in this case 5. Fifth-order convergence is indeed seen for all node sets in Figure 8. Also in this figure, the RBF-FD method is compared to a 5th-order upwind scheme with and without a WENO limiter, both of the latter exhibiting a third-order convergence rate. The reason for the comparison is that this order FD upwinding scheme is the type used in the Weather Research and Forecasting (WRF) Model (http://www.wrf-model.org/). The time steps for both methods are comparable, with less than a 1% difference.

It should be remembered that this test case was set up to investigate what the numerical results for the proposed RBF-FD method would be under no boundary effects. So, if there is the unusual circumstance that the solution does not interact with any boundaries in a bounded domain *and* no refinement will be needed, then solving the problem on a Cartesian lattice will give just as good results as hexagonal. Furthermore, the fact that scattered nodes performed just as well as the other two layouts is of great benefit since it paves the way for the ability to implement local node refinement.

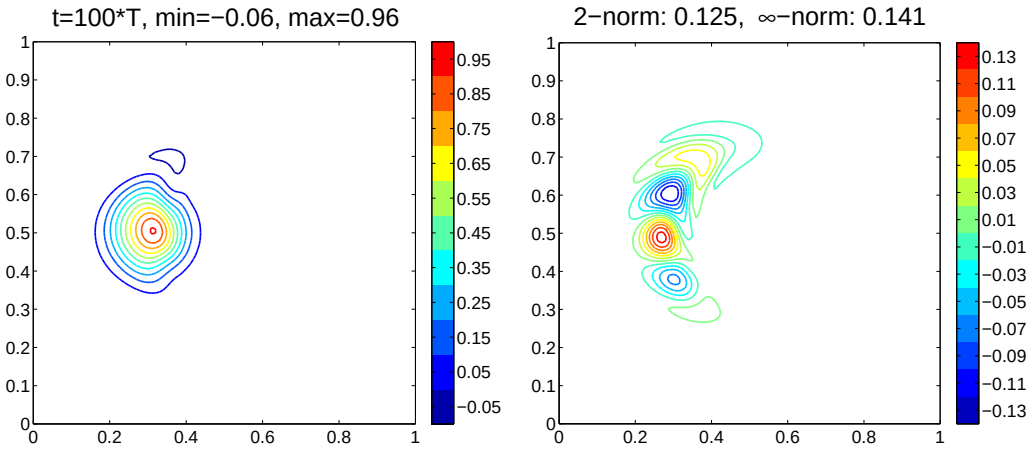

Figure 6: Solution (left panel) and error (right panel) at $t = 100T$ on 40,401 ($h = 0.005$) hexagonal nodes using $\phi(r) = r^9$ with up to 4^{th} degree polynomials on a 37-node stencil and Δ^3-type hyperviscosity.

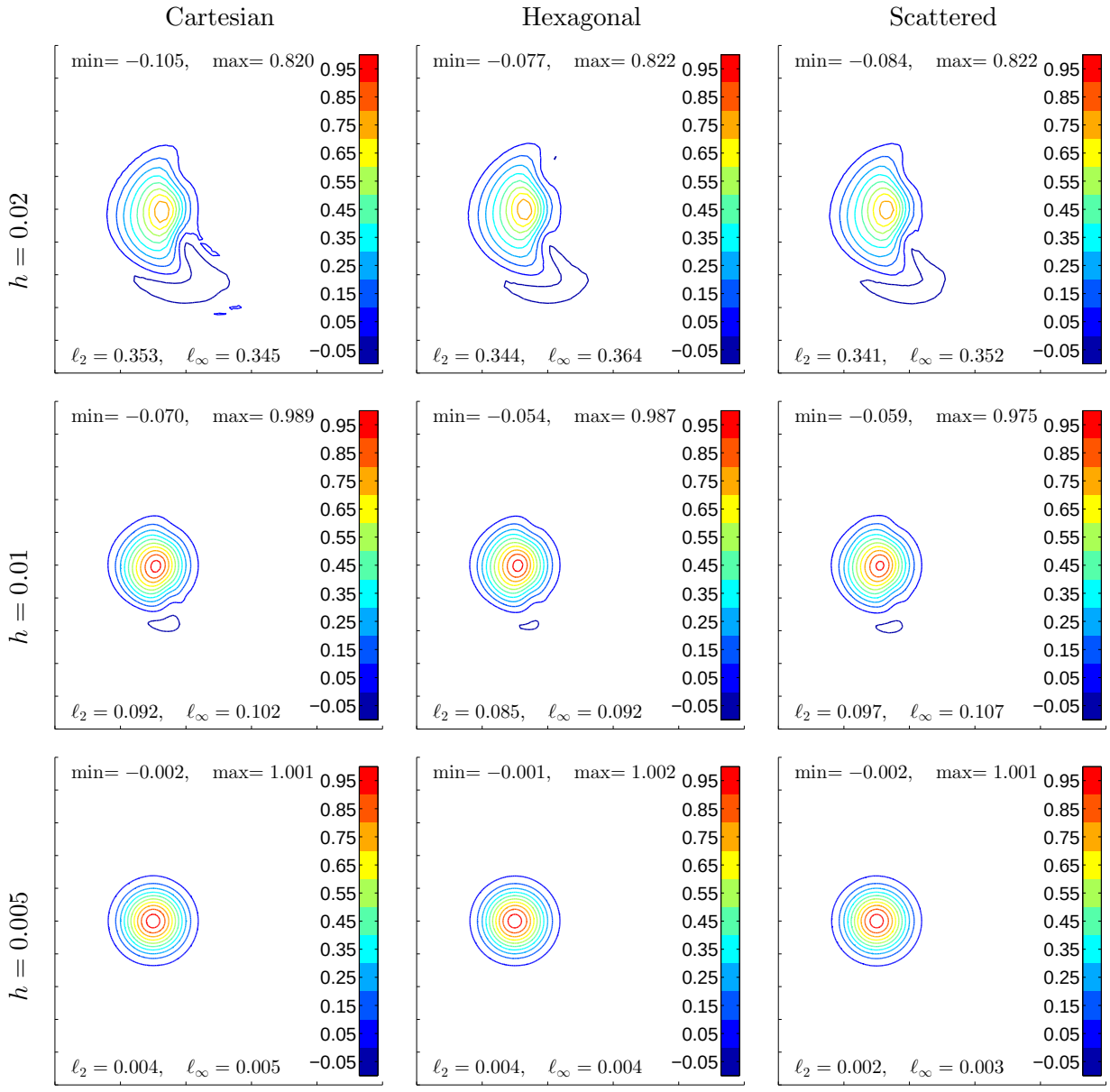

Figure 7: Plots of solutions at $t = T = 1$ using the three different types of node-sets. $\phi(r) = r^3$, polynomials up to degree 5 on a 37-node stencil and Δ^3-type hyperviscosity were used. The amount of hyperviscosity, γ, varies between node-sets, but is on the $O(10)^{-12}$ to $O(10)^{-14}$.

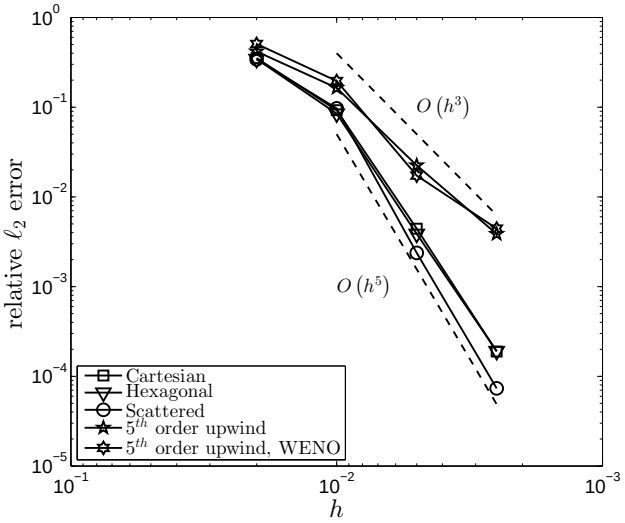

Figure 8: Convergence plots of relative error $\left(\frac{\|\psi_{\text{approx}}-\psi_{\text{exact}}\|_2}{\|\psi_{\text{exact}}\|_2}\right)$ vs. $(h = 0.02, 0.01, 0.005, 0.0025)$. In all cases, $\phi(r) = r^3$ with polynomials up to degree 5 on a 37-node stencil and Δ^3-type hyperviscosity were used. The error decreases $O\left(h^5\right)$, which is expected since up to 5^{th} degree polynomials were included and only first derivatives need to be approximated in the PDE.

6.2 Governing Equations for Navier-Stokes test cases

In all test cases below, the set of governing equations is the 2D nonhydrostatic compressible Navier-Stokes equations at low Mach number, M≈ 0.1, in a rectangular or square domain. The equations are given by

$$\begin{aligned}
\frac{\partial u}{\partial t} &= -u\frac{\partial u}{\partial x} - w\frac{\partial u}{\partial z} - c_p\theta\frac{\partial \pi}{\partial x} + \mu\Delta u,\\
\frac{\partial w}{\partial t} &= -u\frac{\partial w}{\partial x} - w\frac{\partial w}{\partial z} - c_p\theta\frac{\partial \pi}{\partial z} - g + \mu\Delta w,\\
\frac{\partial \theta}{\partial t} &= -u\frac{\partial \theta}{\partial x} - w\frac{\partial \theta}{\partial z} + \mu\Delta\theta,\\
\frac{\partial \pi}{\partial t} &= -u\frac{\partial \pi}{\partial x} - w\frac{\partial \pi}{\partial z} - \frac{R_d}{c_v}\pi\left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z}\right),
\end{aligned}$$

where u and w are the velocities in the horizontal and vertical directions, respectively, $\pi = \left(\frac{P}{P_0}\right)^{R_d/c_p}$ is the non-dimensional Exner pressure $\left(P_0 = 1\times 10^5 \text{ Pa}\right)$, and $\theta = \frac{T}{\pi}$ is the potential temperature. The constants $c_p = 1004$ and $c_v = 717$ are the specific heat at constant pressure and the specific heat at constant volume, respectively, with the gas constant for dry air being $R_d = c_p - c_v = 287$. Additional parameters are $g = 9.81 m/s^2$, the gravitational constant, and μ, the dynamic viscosity. Furthermore, it is assumed that all quantities to be solved for, $[u, w, \theta, \pi]^T$, are perturbations $(')$ to a background state $(^-)$ that is in hydrostatic balance, i.e. the fluid is initially at rest, $\overline{u} = \overline{w} = 0$, and the background Exner pressure is a linear function of height z, $\dfrac{d\overline{\pi}}{dz} = -\dfrac{g}{c_p\overline{\theta}}$. Substituting this latter relation into the equations above and writing $\theta = \overline{\theta} + \theta'$ and $\pi = \overline{\pi} + \pi'$ (where the $(')$ symbol has been dropped for reading clarity) yields the governing equations to be

used for computation:

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} - w\frac{\partial u}{\partial z} - c_p\left(\overline{\theta}+\theta\right)\frac{\partial \pi}{\partial x} + \mu\Delta u, \tag{5}$$

$$\frac{\partial w}{\partial t} = -u\frac{\partial w}{\partial x} - w\frac{\partial w}{\partial z} - c_p\left(\overline{\theta}+\theta\right)\frac{\partial \pi}{\partial z} + \frac{g\theta}{\overline{\theta}} + \mu\Delta w, \tag{6}$$

$$\frac{\partial \theta}{\partial t} = -u\frac{\partial \theta}{\partial x} - w\frac{\partial \theta}{\partial z} + \mu\Delta\theta, \tag{7}$$

$$\frac{\partial \pi}{\partial t} = -u\frac{\partial \pi}{\partial x} - w\left(\frac{d\overline{\pi}}{dz} + \frac{\partial \pi}{\partial z}\right) - \frac{R_d}{c_v}\left(\overline{\pi}+\pi\right)\left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z}\right). \tag{8}$$

In the following studies, the perturbation notation ($'$) is generally included when reporting results to keep in mind these are perturbation quantities.

6.3 Numerical Set-up for the NS cases

The governing equations (5)-(8) are solved numerically using a method-of-lines (MOL) approach. PHS RBFs, $\phi(r) = r^7$, with polynomials up to fourth degree are used on a stencil-size of $n = 37$ to approximate all spatial derivatives locally. The remaining system of first order ODEs in time is solved with RK4. A Δ^3-type hyperviscosity is applied in all cases to damp high-frequency modes. The time-step for all test cases as a function of resolution is

node-spacing (h):	800m	400m	200m	100m	50m	25m
Δt:	2s	1s	$\frac{1}{2}$s	$\frac{1}{4}$s	$\frac{1}{8}$s	$\frac{1}{16}$s

Table 2 below gives the domain size, and number of nodes used as a function of resolution for the numerical studies of the NS test cases.

Table 2: Information regarding the computational domain for each test case. The number of nodes (N) is for hexagonal nodes.

Test (domain size ($x \times z$ in km))	Resolution (m)	$\approx N$
Straka Density Current [38] ($[-25.6, 25.6] \times [0, 6.4]$)	800	720
	400	2,700
	200	10,000
	100	38,500
	50	152,650
Translating Density Current[42] ($[0, 36] \times [0, 6.4]$)	800	500
	400	1,900
	200	7,040
	100	27,040
	50	107,350
Rising Thermal Bubble ($[0, 10] \times [0, 10]$)	200	2,980
	100	11,760
	50	46,720
	25	185,430

6.4 Density Current

In the density current test case [38], a hydrostatic neutral atmosphere is perturbed by a C^1 bubble in the potential temperature. A mass of cold air falls to the ground and develops three smooth and distinct Kelvin-Helmoltz rotors as it spreads to the sideways. This test has become widely used in weather modeling community for assessing the ability in new numerical schemes to capture the physics in nonhydrostatic fluid flows [27, 20, 36, 26]. Figure 9 shows the behavior of the numerical solution in time from $t = 0$s until the final time, $t = 900$s.

The computational domain is $[-25.6, 25.6] \times [0, 6.4]$ km^2, and the governing equations (5)-(8) are solved with a viscosity of $\mu = 75$ m^2/s.

Define $\overline{\theta}$ and $\overline{\pi}$

Let $\overline{T}(z) = T_s - \frac{g}{c_p} z$ be the background state for temperature, where $T_s = 300$ is the temperature at the ground surface in Kelvin. Then, the background states for potential temperature and Exner pressure are given by

$$\overline{\theta} = T_s, \qquad \overline{\pi}(z) = \frac{\overline{T}(z)}{\overline{\theta}} = 1 - \frac{g}{c_p T_s} z.$$

Define initial conditions

The vector of unknowns is initially zero except for the potential temperature.

$$u|_{t=0} = 0, \quad w|_{t=0} = 0, \quad \pi'|_{t=0} = 0.$$

The $\left(C^1\right)$ initial condition for θ' is derived via a cool cosine bubble in the temperature T defined by

$$T'|_{t=0} = \begin{cases} -\frac{15}{2} \left\{1 + \cos\left[\pi_c r(x,z)\right]\right\}, & r(x,z) \le 1, \\ 0, & r(x,z) > 1, \end{cases} \tag{9}$$

where $\pi_c = 3.14159\ldots$ is the standard trigonometric constant and

$$r(x,z) = \sqrt{\left(\frac{x - x_c}{x_r}\right)^2 + \left(\frac{z - z_c}{z_r}\right)^2}, \qquad \begin{array}{l} (x_c, z_c) = (0 \text{ km}, 3 \text{ km}), \\ (x_r, z_r) = (4 \text{ km}, 2 \text{ km}). \end{array}$$

Then, the initial condition for θ' can be found by dividing by $\overline{\pi}$:

$$\theta'|_{t=0} = \left[\theta - \overline{\theta}\right]_{t=0} = \left.\frac{T}{\pi}\right|_{t=0} - T_s = \left.\frac{\overline{T} + T'}{\overline{\pi}}\right|_{t=0} - T_s = \left[\overline{\theta} + \frac{T'}{\overline{\pi}}\right]_{t=0} - T_s = \frac{T'|_{t=0}}{\overline{\pi}}.$$

Define boundary conditions

The problem is periodic in the x direction with the following conditions on the top and bottom boundaries in z:

$$w' = 0\ , \qquad \frac{\partial u'}{\partial z} = 0, \qquad \frac{\partial \theta'}{\partial z} = 0.$$

These are the only boundary conditions necessary to solve the governing equations. However, enforcing the vertical momentum equation (6) on the top and bottom boundaries and assuming

that perturbation in the pressure gradient balances the perturbation in the potential temperature leads to the following condition for π',

$$\frac{\partial \pi'}{\partial z} = \frac{g\theta'}{c_p \overline{\theta}\left(\overline{\theta} + \theta'\right)}.$$

Furthermore, since the dynamic viscosity μ for air is $\approx 10^{-5}$, a good approximation on the top and bottom boundaries is $\Delta w' = 0$ or $\partial^2 w'/\partial z^2 = 0$ since $w = 0$ on these boundaries. While these two extra boundary conditions on π' and w' conditions are not required, they allow for the use of ghost nodes in all four variables. In summary, the lateral boundaries are periodic, and the complete set of boundary conditions enforced on the top and bottom boundaries is:

$$w = \frac{\partial^2 w}{\partial z^2} = \frac{\partial u}{\partial z} = \frac{\partial \theta'}{\partial z} = 0, \qquad \frac{\partial \pi'}{\partial z} = \frac{g\theta'}{c_p \overline{\theta}\left(\overline{\theta} + \theta'\right)}.$$

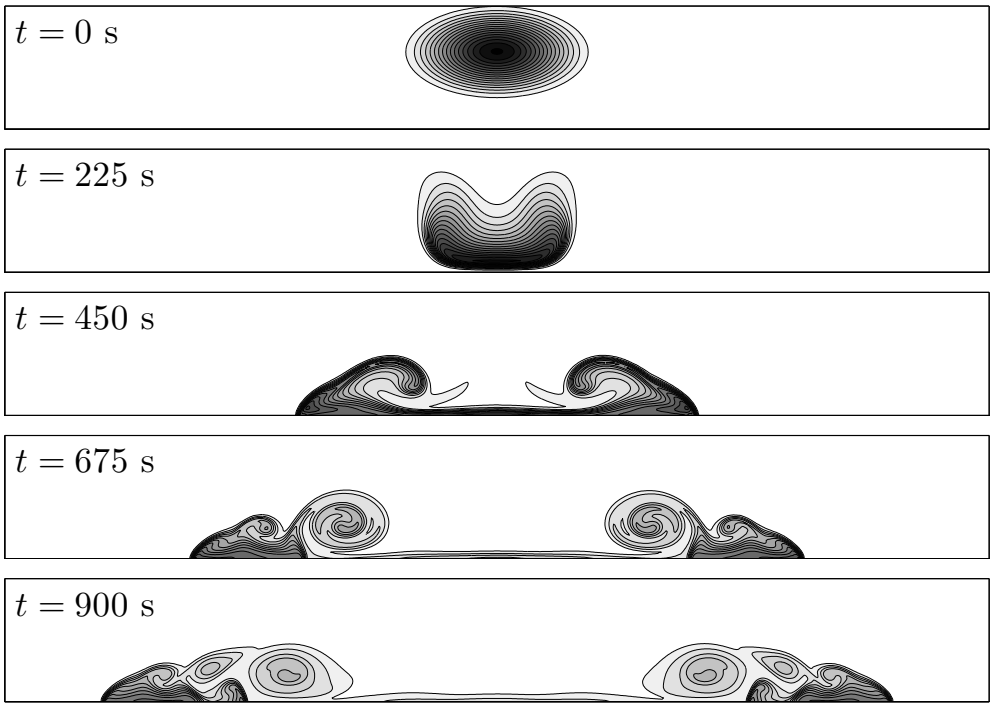

Figure 9: Time evolution of the potential temperature perturbation θ' for the density current test case with $\mu = 75m^2/s$ at a 100m resolution on hexagonal nodes.

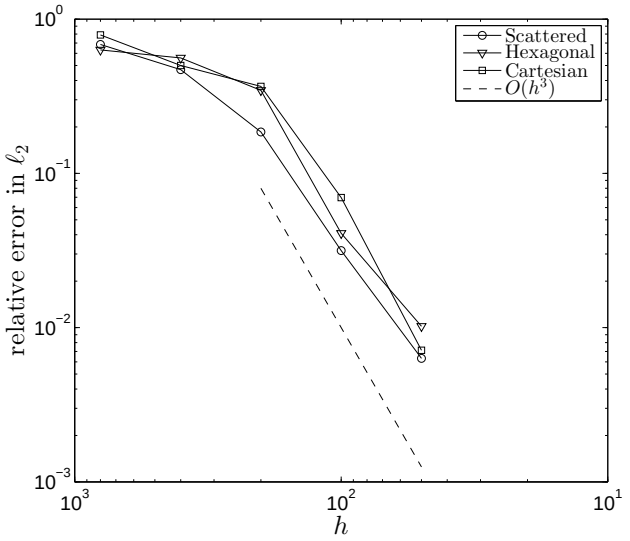

Figure 10: Convergence behavior for θ' in the Straka density current test case with $\mu = 75\, m^2/s$. The h =800m, 400m, 200m, 100m, and 50m errors were calculated using the 25m RBF-FD reference solution.

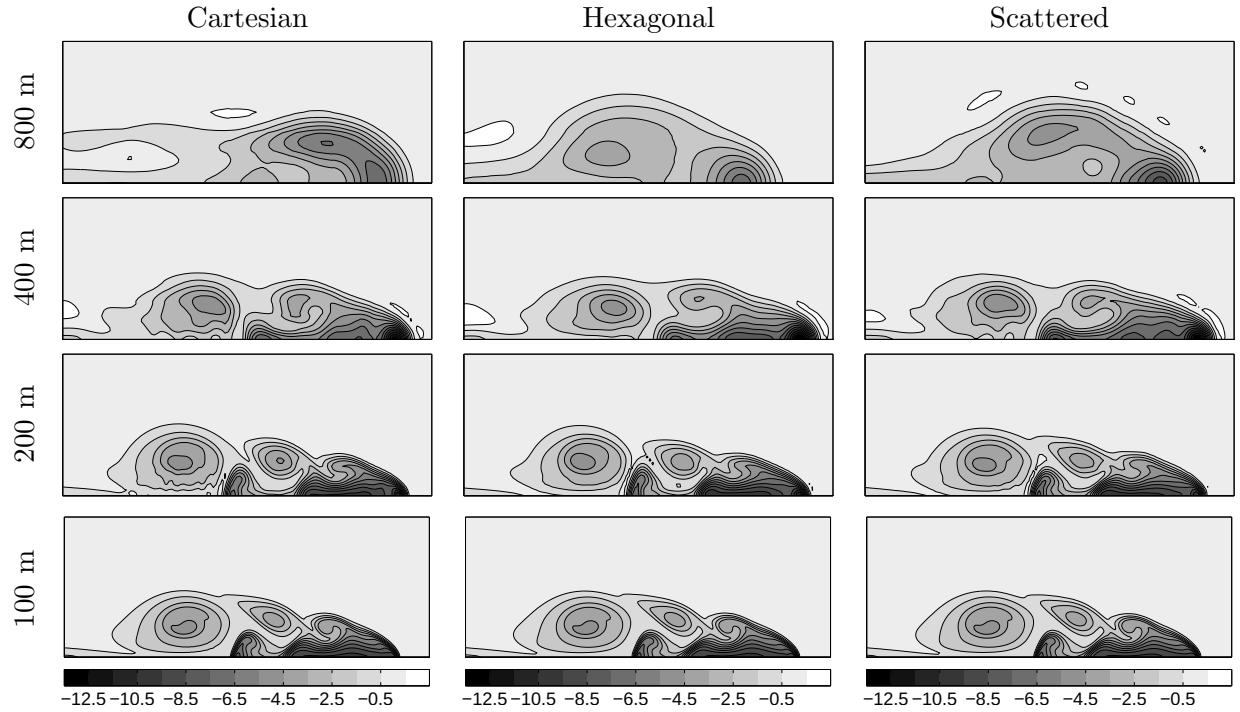

Figure 11: The solution at the final time for the density current test case solved on different node layouts for resolutions 800m, 400m, 200m, 100m. Only half the solution is shown to enlarge details. Contours for the density current begin at -0.5K and are in intervals of 1K. The white areas are enclosed by a contour of 0.5K.

As can be seen in Figure 10, the error in the ℓ_2 norm is approximately the same regardless of the node layout with a third-order convergence rate for node resolutions 200m and less. However, convergence rates do not illustrate how the physics is being resolved with regard to where the data is sampled (i.e. in terms of the node layout). As a result, in Figure 11, the solution of the density current test case is given on the three different node layouts discussed in Section 5 for four different resolutions, varying from 800m to 100m. In the highest resolution displays (100m), although all node layouts seemed to have converged to the same solution, differences can be noted in Table 3, where a 25m test run on hexagonal nodes is used as a reference solution. Results from a 50m test run on the 3 different node sets are also given in the table. On hexagonal and scattered nodes the minimum θ' has indeed converged by 100m, while the maximum θ' is still a fifth of a degree off. Similar error percentages can be found in w' and in the front location at these fine resolutions of 100m and 50m, noting that Cartesian nodes perform slightly worse. Notice that the front for the 100m Cartesian is at the same location as achieved by a 200m hexagonal layout.

At coarser resolutions, such as 400m and 800m, values in the table will be far off the converged 25m solution. Instead, noting physical features of the solution in Figure 11, such as 1) at what resolution do the rotors begin to form, their shape and where, and 2) how much cold air has been entrenched in each rotor, will give a better idea of the capability of the node layout to capture the physics. The following observations can be made:

1. At 800m - approximately 720 nodes in the domain: The hexagonal and scattered node calculations give more clear evidence of the first (largest) rotor being formed. The -3.5K contour in the hexagonal case (circular inner most contour in the first rotor) is even close to its final position if compared to the 100m case. Although the first rotor for the scattered case is not quite as nicely formed as in the hexagonal case, it has entrenched more cold air, having a -4.5K contour (teardrop shape). Notice that at 100m the -4.5K contour is the coldest that appears. In comparison, the 800m Cartesian has barely any rotor formation and is much more wildly oscillatory, which is also noted by the fact that the maximum θ' is 2.43K, at least 1.3K larger than for the other node layouts. See Table 3. Also note in the table that the error in the front location decreases from 4% to 2%, when hexagonal nodes are used opposed to Cartesian, with scattered given an intermediate error of 3%. Both hexagonal and scattered nodes undershoot the correct position while Cartesian overshoot it.

2. At 400m - approximately 2700 nodes in the domain: At this resolution, oscillations due to boundary error effects especially in the first rotor are very pronounced on the Cartesian layout; this carries over even to the 200m resolution for this node case. For scattered nodes there are minor oscillations in the solution. For the hexagonal case, barely any are evident. Formation of the second rotor has the nicest intact shape with the least amount of oscillation in both the hexagonal and scattered, with the latter having entrenched a slightly larger amount of cold air (notice the size of the -3.5K contour teardrop-shaped area in the second rotor of the scattered case).

The differences between the columns of subplots reflect only the intrinsic resolution capabilities of the different node layouts for capturing the physics. The traditional Cartesian choice is the least effective one. At every resolution level, the hexagonal and scattered choices give better accuracy than the Cartesian one. The advantage of generalizing from hexagonal to quasi-uniformly scattered nodes, is that it then becomes easy to implement spatially variable node densities, i.e. to do local refinement in select critical areas. It is very important to note that this major increase in geometric flexibility (from hexagonal to quasi-uniformly scattered) hardly has any negative effect at all on the accuracy that is achieved, nor on the algorithmic complexity of the code.

Table 3: Resolution (h), minimum and maximum values for θ' and w, and front location at various resolutions for the density current test case with $\mu = 75m^2/s$. The front location was determined by the -0.5K contour line.

	h (m)	min $\{\theta'\}$	max $\{\theta'\}$	min$\{w'\}$	max$\{w'\}$	front (m)
Cartesian	800	-7.74	2.43	-9.19	11.00	16,079
	400	-13.45	1.10	-15.21	16.36	16,013
	200	-12.15	0.57	-16.59	17.49	15,799
	100	-9.84	0.27	-16.14	13.45	15,500
	50	-9.71	0.04	-15.96	12.86	15,424
Scattered	800	-8.60	1.11	-10.11	10.05	15,477
	400	-12.03	1.13	-13.26	12.79	15,747
	200	-10.40	0.42	-15.90	13.60	15,597
	100	-9.70	0.21	-16.00	13.12	15,447
	50	-9.70	0.02	-15.95	12.87	15,422
Hexagonal	800	-6.90	1.00	-11.53	9.61	15,101
	400	-13.38	0.98	-12.93	10.11	15,721
	200	-11.42	0.44	-15.90	14.34	15,501
	100	-9.70	0.20	-15.90	12.96	15,444
	50	-9.70	0.01	-15.93	12.90	15,420
Reference	25	-9.70	0.00	-15.93	12.90	15,418

6.4.1 Low-Viscosity Density Current $\mu = 2 \times 10^{-5}$ $\mathbf{m^2/s}$

Here, the density current test case is repeated, except with the dynamic viscosity μ set to that of air. The purpose of this test case is to show that one can stably time step the RBF-FD method in a completely turbulent regime. The same amount of hyperviscosity as well as the same time step are used in this test case as in the one with $\mu = 75\, m^2/s$. Time stability is governed solely by the fact that the time step could not exceed the speed of sound in air.

At such low viscosity, the solution enters the turbulent regime. In such regimes, there is no convergence to any solution as energy cascades to smaller and smaller scales, eventually entering the sub-grid scale domain. Nevertheless, it is interesting to observe whether the model remains stable in this regime. Figure 12 shows the solution at 100m, 50m and 25m resolutions on the three different node layouts. For any given resolution the solution looks completely different depending on the node layout. This is to be expected as changing the node layout in practically the absence of explicit viscosity is equivalent to introducing slight perturbations in the solution. A more robust illustration of this will be given in the test case of a rising thermal bubble, Section 6.5.

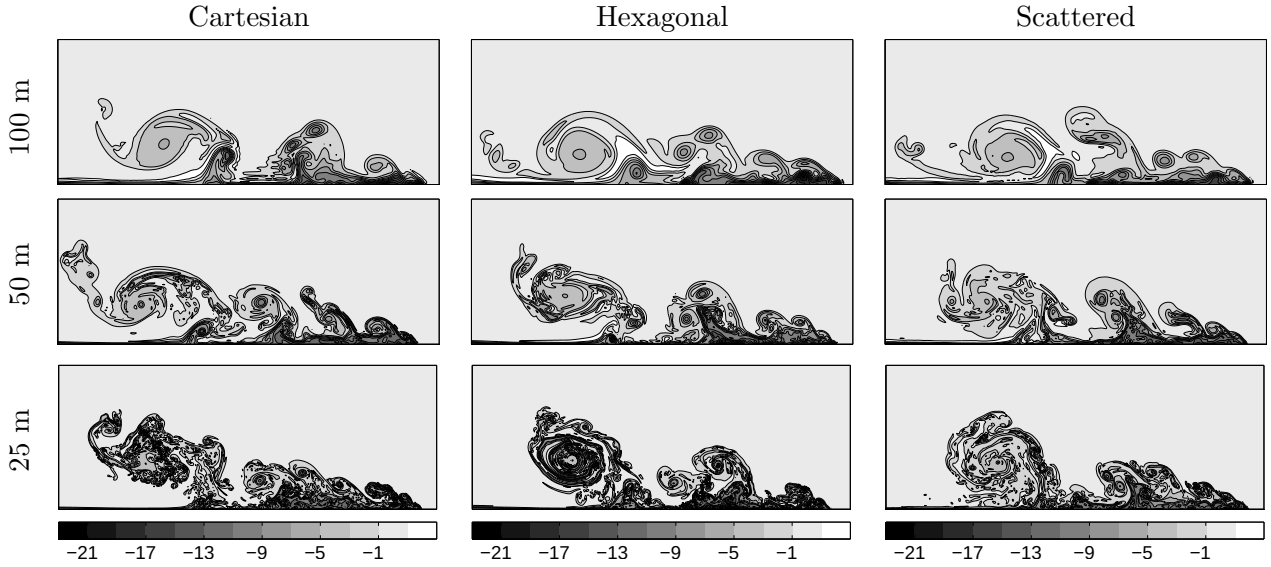

Figure 12: Potential temperature perturbation θ' for the low-viscosity density current 6.4.1 using 100m, 50m, and 25m nodes at $t = 900s$.

6.4.2 Translating Density Current, $\mu = 75\, m^2/s$

This test is the same as in Section 6.4, except that the domain is now $[0, 36] \times [0, 6.4]$ km^2 and there is a horizontal background wind of $\overline{u} = 20$ m/s. The size of the domain is set up so that at $t = 900$s the two "halves" of the solution should be symmetric about $x = 18$km. The introduction of a background mean flow introduces a large difference in the movement of each half of the solution. The right portion of the outflow has horizontal velocities ~ 50 m/s, while the left portion has velocities ~ 10 m/s. As a result, it tests the ability of the scheme to translate the features of the solution at the correct speeds and to generate the correct rotor structures that arise from the local shearing instabilities. For this case, only Cartesian and hexagonal nodes are considered, as the case tests the degree to which symmetry is broken in the two halves of the solution at the final time. Figure 13 illustrates the time series of the potential temperature θ' field, showing how the right part of the solution is advected through the the right side of the domain, with the front locations facing one another at 900s (instead of facing the lateral boundaries as in the previous test case).

In Figure 14, the two halves of the solution are compared about the line of symmetry (18km) for Cartesian and hexagonal nodes from 800m to 100m resolutions. The general observations that can be seen are:

1. The 800m hexagonal node layout performs highly superior to the Cartesian both in terms of symmetry between the two sides, intactness of the large rotor, and its relative location when compared to the 100m case.

2. At both the 400m and 200m Cartesian case, the left half of the solution that has been advected through the right boundary displays a significant amount of Runge phenomena ('wiggles' in the contour lines near the boundary at 200m and severe distortion of the primary and secondary rotor at 400m). This is not the case for hexagonal nodes, which at 400m and 200m, shows relatively nice symmetry between the two sides.

3. At 100m, there is no distinction between the two node sets.

The actual front locations, in terms of there distance from the 18km mark, are given in Table 4. Note that the front on the right is farther from the line of symmetry (18km mark) than the one that has been advected through the boundary for the resolutions 800m to 100m. At 50m, both the Cartesian and hexagonal case is symmetric about the 18km mark. However the distance from the front to the line of symmetry varies between the two cases, 2586m versus 2595m, respectively.

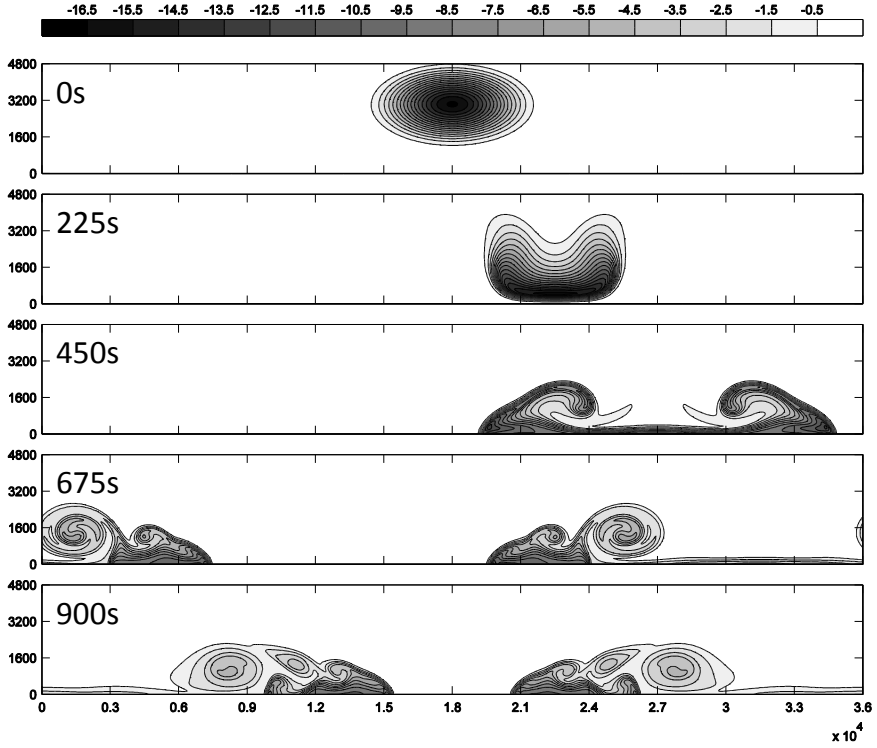

Figure 13: Time evolution of the potential temperature θ' for the translating density current test case. Snapshots were generated using the 100m RBF-FD solution on hexagonal nodes.

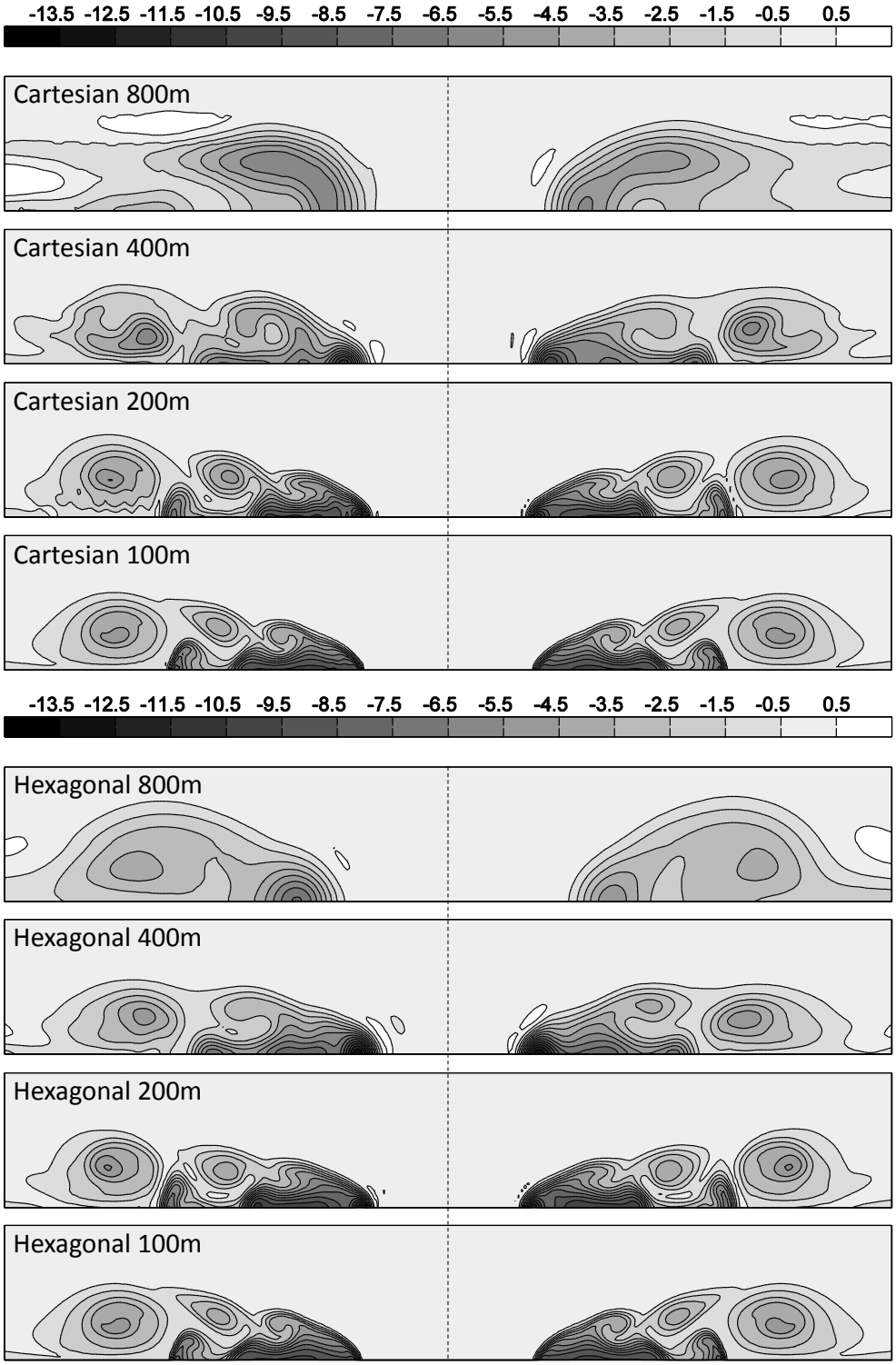

Figure 14: Potential temperature perturbation θ' for the the translating density current at $t = 900s$ for Cartesian and hexagonal nodes at given resolutions. The dashed line is the $x = 18$km mark about which the two halves should be symmetric.

Table 4: Left and right front locations for the translating density current test case 6.4.2 as given by the distance from the -0.5K contour line to the center of the domain, $x = 18$km. All values are calculated at the final time, $t = 900$s. Results are for $\phi(r) = r^7$ with up to fourth degree polynomials on a 37-node stencil. A 25m reference solution is given for the hexagonal nodes.

	h	left front (m)	right front (m)
Cartesian	800	3,065	3,215
	400	1,915	2,085
	200	2,098	2,205
	100	2,487	2,512
	50	2,586	2,586
Hexagonal	800	3,410	3,575
	400	1,956	2,046
	200	2,165	2,260
	100	2,555	2,580
	50	2,595	2,595
Reference	25	2,595	2,595

6.5 Rising Thermal Bubble

With this last test case, the paper comes full circle in that the presented RBF-FD method is tested on a problem with sharp gradients and very little boundary interaction as in the advective transport of a scalar variable, but is modeled by the same 2D nonhydrostatic compressible Navier-Stokes equations as in the density current tests. The only difference is the initial condition is given by a C^0 cone-shaped perturbation. The bubble is warmer than the surrounding atmosphere and thus rises toward the top boundary. However, the time interval and domain size are chosen so that the bubble never interacts with the boundaries.

There are two variations for this test case:

1. $\mu = 10$ m^2/s: By adding a small amount of explicit viscosity, the convergence behavior of the solution can be studied.

2. $\mu = 2\times10^{-5}$ m^2/s: At such low viscosity, the bubble is in a turbulent regime, and the behavior of the true solution is unknown. The purpose of the test is to see if the RBF-FD method can give reasonable results when the initial condition is not even continuously differentiable as well as observe how the instability pattern at the leading edge of thermal changes with the node layout.

6.5.1 Case $\mu = 10$ m^2/s

The computational domain is $[0, 10] \times [0, 10]$ km^2. The hydrostatic background states are defined by $\overline{\theta} = T_s$ and $\overline{\pi}(z) = 1 - \frac{g}{c_p T_s} z$, with $T_s = 300$K being the surface temperature. The horizontal and vertical velocities and the Exner pressure perturbation (π') are initially zero, while the potential temperature perturbation is prescribed as a warm cone-shaped "bubble" with a jump in the first derivative $\left(C^0\right)$:

$$\theta'\big|_{t=0} = 2\max\left\{0, 1 - r(x,z)/R\right\}.$$

Here, $R = 1.5$ km is the radius of the bubble, and

$$r(x,z) = \sqrt{(x-x_c)^2 + (z-z_c)^2}, \qquad (x_c, z_c) = (5 \text{ km}, 3 \text{ km}).$$

The same boundary conditions as in the density current problem are enforced on the top and bottom boundaries

$$w = \frac{\partial^2 w}{\partial z^2} = \frac{\partial u}{\partial z} = \frac{\partial \theta'}{\partial z} = 0, \qquad \frac{\partial \pi'}{\partial z} = \frac{g\theta'}{c_p \overline{\theta}\left(\overline{\theta} + \theta'\right)}. \tag{10}$$

The lateral boundary conditions are given by

$$u = \frac{\partial^2 u}{\partial x^2} = \frac{\partial w}{\partial x} = \frac{\partial \theta'}{\partial x} = \frac{\partial \pi'}{\partial x} = 0.$$

Figure 15 shows the time series of the solution for a 25m resolution ($N = 185,730$) on hexagonal nodes using r^7 with up to 4th-order polynomials and a Δ^3-type hyperviscosity. The main purpose of this test is to make sure that, under refinement, all node layouts converge to the same solution (as this will not be the case in the next variation of the test) and to see if the convergence rate follows the predictions of Section 4, even with a C^0 initial condition. Figure 16 shows the final solution for the three different node layouts from a resolution of 200m to 25m. At 25m resolution, all solutions are visually identical. At coarser resolutions, as 100m, Cartesian and hexagonal nodes are more similar with scattered nodes having more incongruities at the leading edge of the rising bubble. A possible reason for this could be symmetry-breaking associated with scattered node layouts that would affect areas of large shear.

In terms of convergence, Figure 17 shows that even though the initial condition is C^0, the method does achieve 4th order convergence as predicted when using up to fourth degree polynomials. All nodes sets converge at fourth order under refinement, with Cartesian giving the best accuracy for $h \leq 100$m. Note that at coarser resolutions from 400m too 100m only second order convergence is achieved.

Table 5 shows to what degree the solution has converged in terms of how high the bubble should have risen, θ', and w'. Note that when comparing against the 12.5m hexagonal node reference solution, all solutions in all variables have converged by 25m. In terms of $\max\{\theta'\}$ and $\max\{w'\}$ all solutions have converged by 50m. The $\min\{\theta'\}$ is almost identical for all node sets while $\min\{w'\}$ varies between node sets for coarser resolutions. In terms of bubble height Cartesian nodes seem to perform the best.

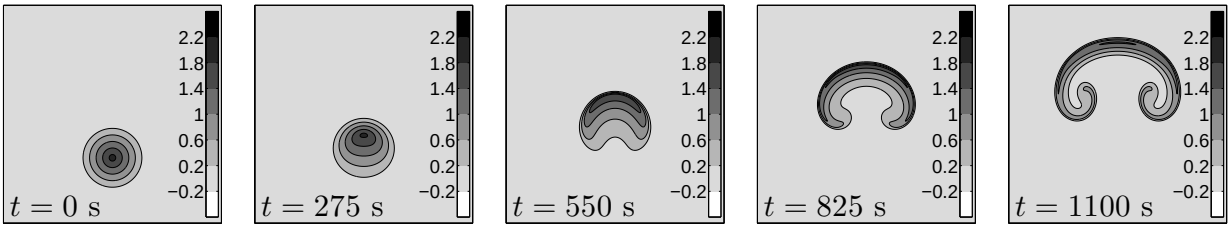

Figure 15: Time evolution of the potential temperature θ' for the $\mu = 10$ m^2/s rising thermal bubble. Snapshots were generated using the 25m RBF-FD solution on hexagonal nodes.

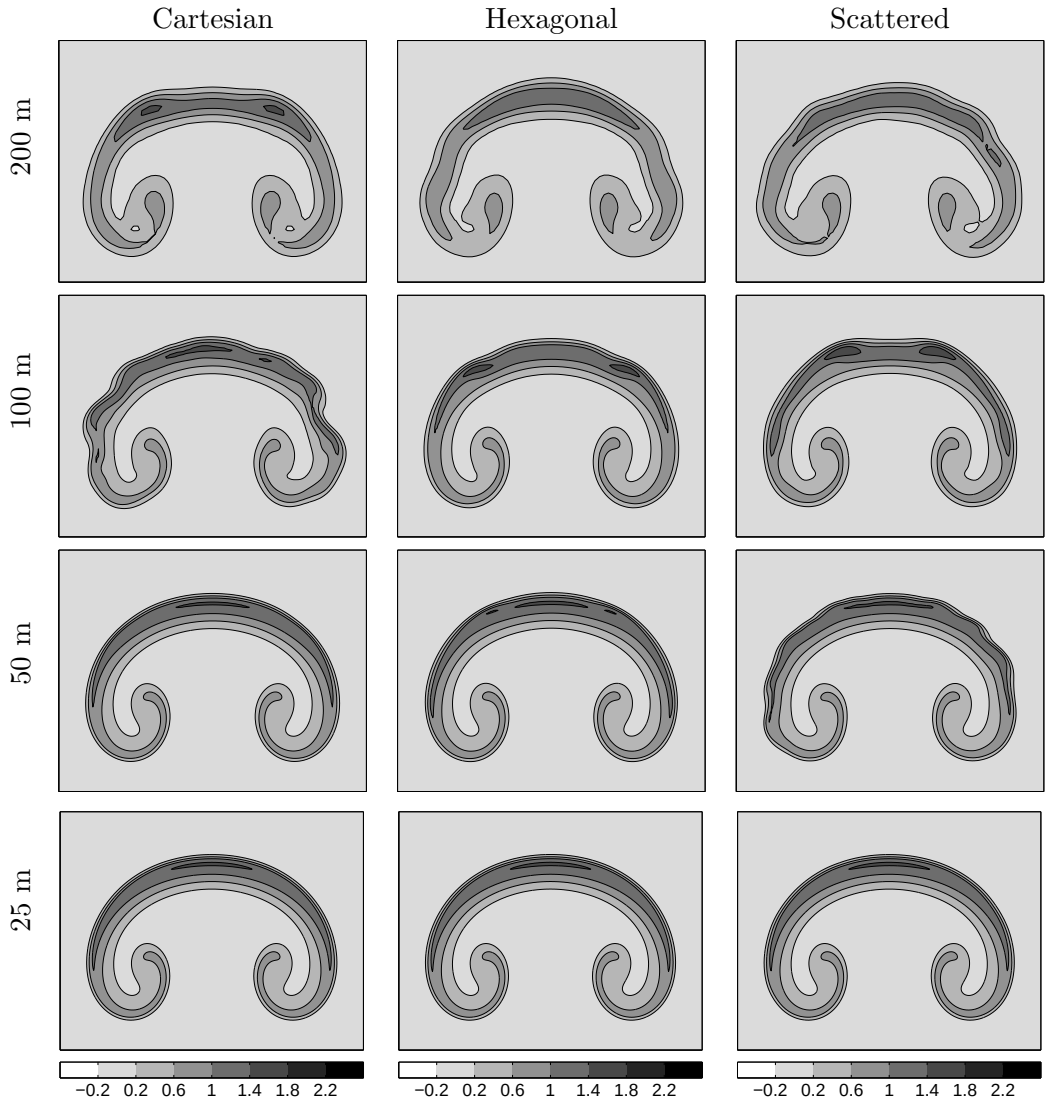

Figure 16: Numerical solutions for the rising thermal bubble with $\mu = 10$ m^2/s (6.5.1) on the three different types of node distributions at various resolutions, shown at the final simulation time, $t = 1100$s.

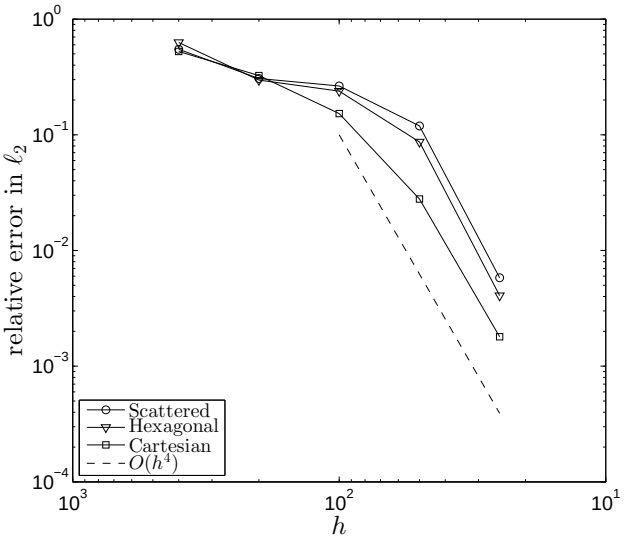

Figure 17: Convergence behavior for θ' in the rising thermal bubble test case 6.5.1. The h =400m, 200m, 100m, 50m, and 25m errors were calculated using the 12.5m RBF-FD reference solution.

Table 5: Resolution (h), minimum and maximum values for θ' and w', and bubble height at various resolutions for the rising thermal bubble 6.5.1. Results are for $\phi(r) = r^7$ with up to fourth degree polynomials on a 37-node stencil. The bubble height was determined by the intersection of the 0.1K contour and the line $x = 5$km.

	h (m)	min $\{\theta'\}$	max $\{\theta'\}$	min$\{w'\}$	max$\{w'\}$	bubble height (m)
Cartesian	200	-0.11	1.46	-7.56	11.06	8,467
	100	-0.08	1.53	-7.93	11.34	8,539
	50	-0.02	1.43	-7.87	11.43	8,534
	25	0.00	1.43	-7.74	11.43	8,535
Hexagonal	200	-0.11	1.36	-7.67	11.12	8,686
	100	-0.09	1.65	-8.05	11.49	8,527
	50	-0.02	1.43	-7.75	11.43	8,553
	25	0.00	1.43	-7.74	11.43	8,535
Scattered	200	-0.11	1.25	-7.74	10.92	8,581
	100	-0.09	1.48	-8.42	11.40	8,557
	50	-0.02	1.43	-8.22	11.43	8,525
	25	0.00	1.43	-7.75	11.43	8,535
Reference	12.5	0.00	1.43	-7.74	11.43	8,535

6.5.2 Case $\mu = 2 \times 10^{-5}$ $\mathbf{m}^2/\mathbf{s}$

The rising thermal bubble test case is repeated with the viscosity μ set to 2×10^{-5}. The first purpose of this test case is simply to demonstrate that the proposed RBF-FD method, implemented with such low viscosity and a C^0 initial condition, has complete time stability using the same time step and amount of hyperviscosity as is the previous section. Secondly, we are interested in observing how the instability pattern at the leading edge of thermal bubble evolves as the node layout changes. Normally, in numerical testing, to see different evolutions of a solution the initial condition is perturbed. However with RBF-FD, one has the flexibility of leaving the initial condition intact

and perturbing the node layout, which in a turbulent regime will lead to different evolutions of the solution. This can be seen in Figure 18. At 200m, there is not much difference between the bubbles. However as can be seen in the 25m results, the shear instability layer at the leading edge of the bubble (darkest contours) develops tight eddies whose structure varies significantly depending on the node layout. In both the Cartesian and hexagonal case, the eddy development is completely symmetric about the midpoint of the bubble due to the symmetry in the node layout, while in the scattered node layout this is not the case (a seemingly more realistic scenario for modeling warm air entrainment in the atmosphere). Furthermore, the scale of the eddies and the degree to which they excite finer scale instabilities varies between the node sets. The Cartesian node layout produces the largest scale eddies as well as the smallest amount of eddies. In contrast, the hexagonal nodes produce a rather strange bubble shape with very fine scale eddy structure.

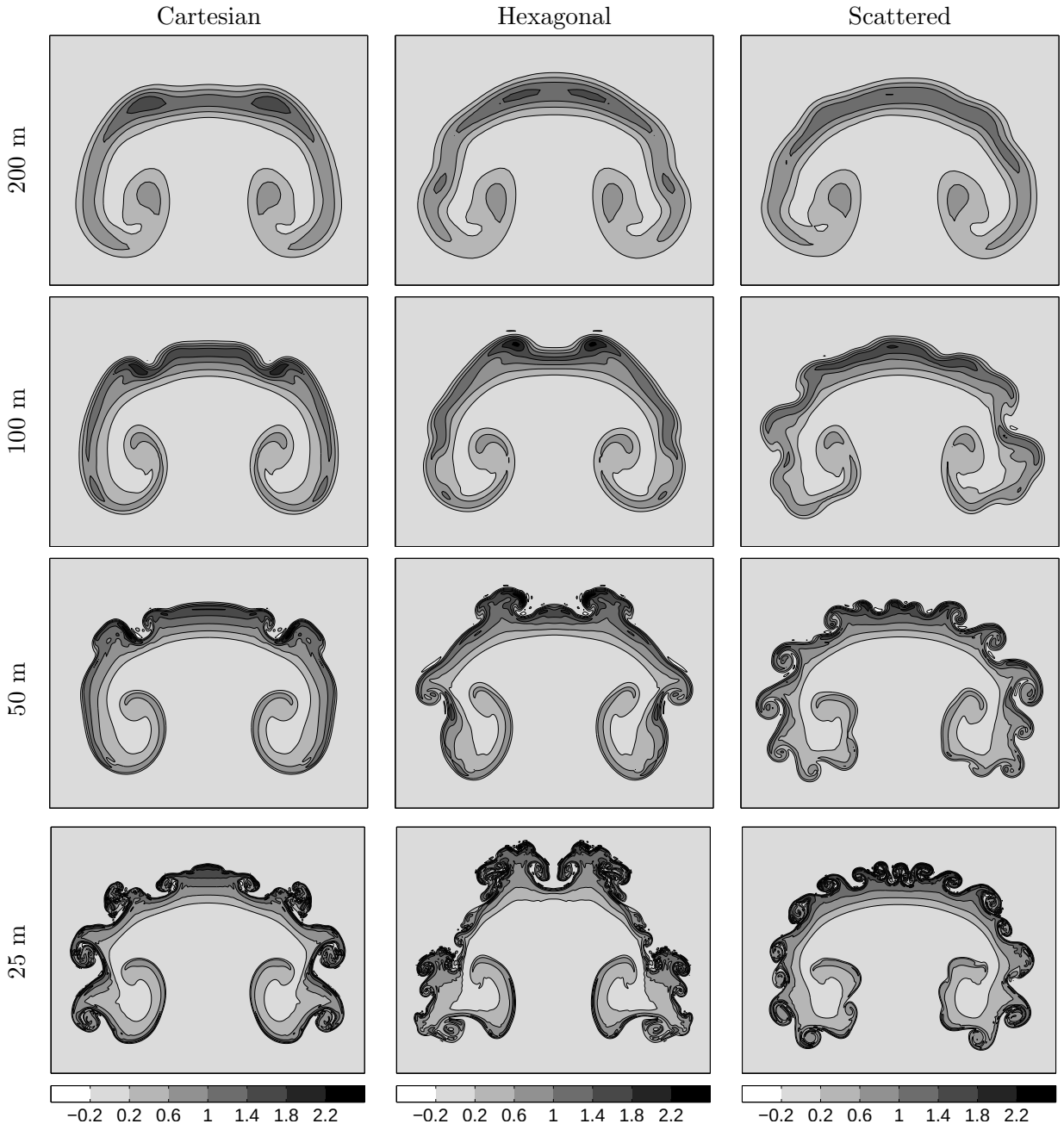

Figure 18: Numerical solutions for the rising thermal bubble with $\mu = 2 \times 10^{-5}$ m^2/s on the three different types of node distributions at various resolutions. All results are shown at the final simulation time, $t = 1100$s.

7 Conclusions and summary

In this paper, a modified RBF-FD method is introduced to construct differentiation weights based on a combined RBF-polynomial basis, using RBF polyharmonic splines ($\phi(r) = r^m$) with polyno-

mial functions up to degree l in the given dimension of the problem. The method is applied to three standard test cases in the numerical weather prediction community [2, 38, 29], with the latter two based on the Navier-Stokes (NS) equations. In addition, the effect of node layout (Cartesian, hexagonal, or scattered) on the error as well the qualitative character of the solution is considered. The following observations are made:

1. Under refinement, with the inclusion of polynomials, stagnation (saturation) error is evaded.

2. In the absence of boundary effects, the convergence rate is controlled, not by the order of the PHS, but by the highest degree polynomials used.

3. Increasing the order of the PHS marginally increases the accuracy, as the constant that multiples the convergence rate decreases.

4. For stable configurations that require no tuning of the hyperviscosity (e.g. $\gamma = 2^{-6}h^{-2k}$ for all NS tests), the number of nodes in the stencil, n, should be approximately twice the number of polynomial basis functions, $(l+1)(l+2)/2$ in 2D. Hence on an $n = 37$ node stencil, up to fourth-order polynomials (15 in 2D) are used for the NS equations.

5. In the absence of boundary effects, for a hyperbolic PDE, neither the character of the solution, the error, nor the convergence rate is sensitive to the node layout.

6. In the presence of boundaries, the solution on Cartesian nodes exhibited significant oscillations (Runge phenomena) near the boundary. This is not the case with hexagonal nodes, which was the most effective node layout in correctly capturing the physics, especially at lower resolutions.

7. In all cases, quasi-uniformly scattering the nodes showed no detriment to the quality of the solution, error, or convergence and in the majority of the cases performed better than Cartesian layouts.

8. Decreasing the viscosity by 6 orders of magnitude, (i.e. increasing in the Reynolds number by the same factor), does not require any change to the time step or amount of hyperviscosity added.

9. In the turbulent regime, the type of node layout heavily impacts the location and structure of eddy development on the leading edge of the thermal as well as the degree of excitation of finer scale instabilities.

Acknowledgements The authors would like to thank Professor Bengt Fornberg and Dr. Victor Bayona for useful comments and discussions. Dr. Flyer and Mr. Gregory Barnett would like to acknowledge the support of NSF grant DMS-094581. The National Center for Atmospheric Research is sponsored by NSF.

A Symmetric Stencils

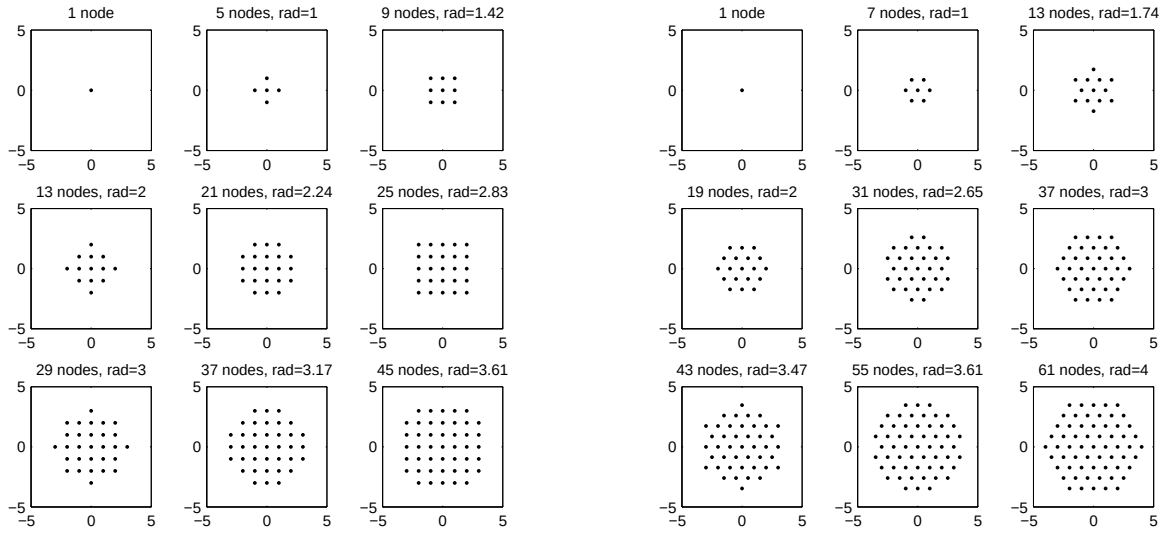

Figure 19: Symmetric stencils for cartesian and hexagonal nodes. Note that 13 and 37 are the only reasonably small stencil-sizes held in common.

References

[1] V. Bayona, M. Moscoso, and M. Kindelan, *Optimal constant shape parameter for multiquadric based RBF-FD method*, J. Comput. Phys. **230** (2011), 7384–7399.

[2] P. N. Blossey and D. R. Durran, *Selective monotonicity preservation in scalar advection*, J. Comput. Phys. **227(10)** (2008), 5160–5183.

[3] E. Bollig, N. Flyer, and G. Erlebacher, *Solution to PDEs using radial basis function finite-differences (RBF-FD) on multiple GPUs*, J. Comput. Phys. **231** (2012), 7133–7151.

[4] G. Chandhini and Y.V.S.S. Sanyasiraju, *Local RBF-FD solutions for steady convection-diffusion problems*, Int. J. Num. Meth. Eng. **72** (2007), 352–378.

[5] H.Q. Chen and C. Shu, *An efficient implicit mesh-free method to solve two-dimensional compressible euler equations*, Int. J. Mod. Phys. C **16(3)** (2005), 439–454.

[6] A. H.-D. Cheng, *Multiquadric and its shape parameter - A numerical investigation of error estimate, condition number, and round-off error by arbitrary precision computation*, Eng. Anal. Bound. Elem. **36** (2012), 220–239.

[7] P. P. Chinchapatnam, K. Djidjeli, P. B. Nair, and M. Tan, *A compact RBF-FD based meshless method for the incompressible Navier-Stokes equations*, J. Eng. Maritime Env. **223** (2009), 275–290.

[8] O. Davydov and D. T. Oanh, *On the optimal shape parameter for Gaussian radial basis function finite difference approximation of the Poisson equation*, Comp. Math. with Appl. **62** (2011), 2143–2161.

[9] J. Duchon, *Splines mimimizing rotation-invariant semi-norms in Sobolev space*, Constructive Theory of Functions of Several Variables, Springer Lecture Notes in Math **21** (1977), 85–100.

[10] A. Emdadi, E. J. Kansa, N. A. Libre, M. Rahimian, and M. Shekarchi, *Stable PDE solutuion methods for large multiquadric shape parameters*, Comp. Mod. Eng. Sci. **25** (2008), 23–41.

[11] G. E. Fasshauer, *Meshfree Approximation Methods with MATLAB*, Interdisciplinary Mathematical Sciences - Vol. 6, World Scientific Publishers, Singapore, 2007.

[12] G. E. Fasshauer and J. G. Zhang, *On choosing "optimal" shape parameters for RBF approximation*, Num. Alg. **45** (2007), 345–368.

[13] N. Flyer and B. Fornberg, *Radial basis functions: Developments and applications to planetary scale flows*, Computers and Fluids **46** (2011), 23–32.

[14] N. Flyer, B. Fornberg, G. A. Barnett, and V. Bayona, *On the role of polynomials in RBF-FD approximations: Interpolation and accuracy*, in preparation (2015).

[15] N. Flyer, E. Lehto, S. Blaise, G. B. Wright, and A. St-Cyr, *A guide to RBF-generated finite differences for nonlinear transport: Shallow water simulations on a sphere*, J. Comput. Phys **231** (2012), 4078–4095.

[16] T. A. Foley, *Near optimal parameter selection for multiquadric interpolation*, J. Appl. Sci. Comput. **1** (1994), 54–69.

[17] B. Fornberg, T. A. Driscoll, G. Wright, and R. Charles, *Observations on the behavior of radial basis functions near boundaries*, Comput. Math. Appl. **43** (2002), 473–490.

[18] B. Fornberg and N. Flyer, *A Primer on Radial Basis Functions with Applications to the Geosciences*, SIAM, Philadelphia, 2015.

[19] B. Fornberg and E. Lehto, *Stabilization of RBF-generated finite difference methods for convective PDEs*, J. Comput. Phys. **230** (2011), 2270–2285.

[20] F. X. Giraldo and M. Restelli, *A study of spectral element and discontinuous Galerkin methods for the navier-stokes equations in nonhydrostatic mesoscale atmospheric modeling: Equation sets and test cases*, J. Comput. Phys. **227** (2008), 3849–3877.

[21] W.W. Grabowski and T.L. Clark, *Cloud-environment interface instability: Rising thermal calculations in two spatial dimensions*, Mon. Wea. Rev. **48(4)** (1991), 527–546.

[22] C.-S. Huang, C.-F. Leeb, and A. H.-D. Cheng, *Error estimate, optimal shape factor, and high precision computation of multiquadric collocation method*, Eng. Anal. Bound. Elem. **31** (2007), 614–623.

[23] A. Iske, *On the approximation order and numerical stability of local Lagrange interpolation by polyharmonic splines*, Modern Developments in Multivariate Approximation (W. Haussmann, K. Jetter, M. Reimer, and J. Stöckler, eds.), International Series of Numerical Mathematics, vol. 145, Birkhäuser Verlag, Basel, 2003, pp. 153–165.

[24] R. J. LeVeque, *High-resolution conservative algorithms for advection in incompressible flow*, SIAM J. Numer. Anal **33** (1996), 627–665.

[25] B. Matérn, *Stochastic models and their application to some problems in forest surveys and other sampling investigations*, Ph.D. thesis, University of Stockholm, Stockholm, Sweden, June 1960.

[26] M. R. Norman, R. D. Nair, and F. H. M. Semazzi, *A low communication and large time step explicit finite-volume solver for non-hydrostatic atmospheric dynamics*, J. Comput. Phys. **230** (2011), 1567–1584.

[27] K.V. Ooyama, *A dynamic and thermodynamic foundation for modeling the moist atmosphere with parameterized microphysics*, J. Atm. Sci **58** (2001), 2072–2102.

[28] M. J. D. Powell, *The theory of radial basis function approximation in 1990*, Advances in Numerical Analysis, Vol. II: Wavelets, Subdivision Algorithms and Radial Functions (W. Light, ed.), Oxford University Press, Oxford, UK, 1992, pp. 105–210.

[29] A. Robert, *Bubble convection experiments with a semi-implicit formulation of the euler equations*, J. Atmosph. Sci. **50** (1993), 1865–1873.

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

[31] M. Scheuerer, *An alternative procedure for selecting a good value for the parameter c in RBF-interpolation*, Adv. Comput. Math. **34** (2011), 105–126.

[32] Y. Y. Shan, C. Shu, and Z. L. Lu, *Application of local MQ-DQ method to solve 3D incompressible viscous flows with curved boundary*, Comp. Mod. Eng. & Sci. **25** (2008), 99–113.

[33] Y. Y. Shan, C. Shu, and N. Qin, *Multiquadric finite difference (MQ-FD) method and its application*, Adv. Appl. Math. Mech. **1** (2009), 615–638.

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

[35] W. C. Skamarock, *Positive-definite and monotonic limiters for unrestricted-time-step transport schemess*, Mon. Wea. Rev **24** (2006), 22412250.

[36] W. C. Skamarock and J. B. Klemp, *A time-split nonhydrostatic atmospheric model for weather research and forecasting applications*, J. Inst. Math. Appl. **227** (2008), 3465–3485.

[37] D. Stevens, H. Power, M. Lees, and H. Morvan, *The use of PDE centers in the local RBF Hermitean method for 3D convective-diffusion problems*, J. Comput. Phys. **228** (2009), 4606–4624.

[38] J.M. Straka, R.B. Wilhelmson, L.J. Wicker, J.R. Anderson, and K.K. Droegemeier, *Numerical solutions of a nonlinear density current: a benchmark solution and comparisons*, Int. J. Num. Meth. Fluids **17** (1993), 1–22.

[39] A. E. Tarwater, *Parameter study of Hardy's multiquadric method for scattered data interpolation*, Technical Report UCRL-54670, Lawrence Livermore National Laboratory, 1985.

[40] H. Wendland, *Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree*, Adv. Comput. Math. **4** (1995), 389–396.

[41] ———, *Scattered Data Approximation*, Cambridge Monographs on Applied and Computational Mathematics, vol. 17, Cambridge University Press, Cambridge, 2005.

[42] L. J. Wicker and W.C. Skamarock, *Time-splitting methods for elastic models using forward time schemes*, Mon. Wea. Rew. **130** (2002), 2088–2097.

Appendix D

Guidelines and Observations for PHS RBF-FD with Polynomials

- The number of RBFs in the basis should be greater than the number of polynomials. If there are equally many polynomials as RBFs in the basis, then the invertibility of the combined matrix is subject to the same conditions that are present in the pure-polynomial "Vandermonde" system. That is, unisolvencey is difficult to satisfy on every stencil.

- A large condition number of the combined RBF-polynomial collocation matrix on a given stencil is not a good indication of the effectiveness of the scheme there (see Section 4.3). Increases in the condition number under refinement can be eliminated using the appropriate polynomial scaling [20].

- One should first choose the highest degree polynomial, and then choose the stencil-size large enough to accomodate the polynomials. For example, after choosing the polynomial degree to be 4, then perhaps one would decide to use a 30-node stencil, so that there are twice as many RBFs as polynomials in each local interpolant.

- Increased regularity of the node-set typically necessitates a larger ratio of RBFs to polynomials. In other words, for a given polynomial order, one can usually get away with a smaller stencil-size on scattered nodes than on cartesian or hexagonal nodes.

- Time-dependent problems with solid boundaries and explicit time-stepping appear to be limited in the order of convergence which can be attained. In other words, fourth or fifth order is probably the best one can achieve for time-dependent PDEs with solid boundaries.

If polynomials of higher degree are included, eigenvalues of the time-stepping matrix stray too far into the right half-plane, and cannot be recovered with hyperviscosity.

- In practice, when the stencil-size is modest, polynomials can be included using the simple polynomial basis (see Table 3.1) with no negative effects. Even though the corresponding monomial basis in 1-D is notoriously ill-conditioned, the polynomial degree does not get high enough for this to be a problem in more than one dimension.