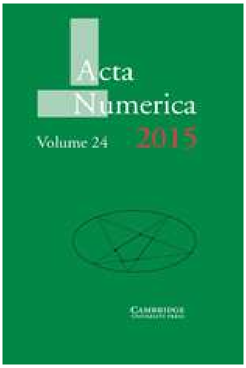


Solving PDEs with radial basis functions

Bengt Fornberg and Natasha Flyer

Acta Numerica / Volume 24 / May 2015, pp 215 - 258
DOI: 10.1017/S0962492914000130, Published online: 27 April 2015

Acta Numerica (2015), pp. 215–258
doi:10.1017/S0962492914000130

Solving PDEs with radial basis functions*

Bengt Fornberg
Department of Applied Mathematics,
University of Colorado,
Boulder, CO 80309, USA
E-mail: fornberg@colorado.edu

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

Finite differences provided the first numerical approach that permitted large-scale simulations in many applications areas, such as geophysical fluid dynamics. As accuracy and integration time requirements gradually increased, the focus shifted from finite differences to a variety of different spectral methods. During the last few years, radial basis functions, in particular in their 'local' RBF-FD form, have taken the major step from being mostly a curiosity approach for small-scale PDE 'toy problems' to becoming a major contender also for very large simulations on advanced distributed memory computer systems. Being entirely mesh-free, RBF-FD discretizations are also particularly easy to implement, even when local refinements are needed. This article gives some background to this development, and highlights some recent results.

CONTENTS

* Colour online for monochrome figures available at journals.cambridge.org/anu.

1. Introduction

The present article is motivated by the recent successes of radial basis functions (RBFs) in the field of computational geoscience. This is quite far from how the RBF methodology first originated. It was proposed by Hardy (1971) in connection with a cartography application that required multivariate scattered-node interpolation. A key non-singularity proof by Micchelli (1986) accelerated the further development and acceptance of RBFs. Pioneering work by Powell (1992) and his collaborators at the University of Cambridge also played a major role in the early history of RBFs. Kansa (1990*a*, 1990*b*) suggested that taking analytic derivatives of RBF interpolants could provide a numerical solution approach for PDEs.

Several monographs on RBFs or with extensive RBF content appeared between 2003 and 2007, by Buhmann (2003), Iske (2004), Wendland (2005) and Fasshauer (2007). *Acta Numerica* has featured two RBF articles, Buhmann (2000) and Schaback and Wendland (2006). These works reflected a growing use of RBFs as a practical computational procedure for increasingly larger-scale applications. Like Fasshauer (2007), the brief monograph by Chen, Fu and Chen (2014) discussed certain RBF approaches for solving PDEs. The perspective presented in this article – as well as in the forthcoming SIAM monograph by Fornberg and Flyer (2015*b*) – is quite different, and will also describe the RBF-FD (RBF-generated finite difference) approach.

We will omit quite large areas of RBF theory that are well described in the previous monographs, and in particular results that are not directly needed for effectively solving PDEs. Attention will, however, be given to ‘flat’ (or near-flat) basis functions, to the use of RBFs for creating weights for RBF-FD formulas, and to the application of RBF/RBF-FD discretizations for solving large-scale benchmark problems, mostly from the geosciences.

2. Background to RBFs for PDEs

PDE discretizations in more than one dimension are often based on meshes, which may be either structured or unstructured, with the latter case best known in the context of finite elements. In the former case, it is relatively easy to approximate derivatives to high orders of accuracy by making finite difference (FD) stencils increasingly wide. That limit provides an alternative way to understand and use pseudospectral (PS) methods (Boyd 2000, Fornberg 1996, Trefethen 2000). A more common way to implement PS methods is via expansions in basis functions, such as tensor products of one-dimensional Fourier or Chebyshev expansions. The computational efficiency of the resulting procedure can in some cases become very high, but this comes at the price of severe regularity constraints on the shape of the computational domain. Spectral element approaches, involving domain decomposition into rectangles (when in two dimensions), together with curvilinear mappings can overcome some of this, and can also permit local refinement in critical areas. However, their implementation is complex and the small node spacing that becomes necessary near internal (artificial) boundaries often severely hurts time-stepping stability conditions.

When solving PDEs, it is very desirable to use entirely mesh-free node distributions, that is, to be able to scatter computational nodes (collocation points) just as needed to fit boundaries and to satisfy spatially variable resolution requirements, but without having to form any local triangles or tetrahedra. Furthermore, with a derivative being a local property of a function, it makes sense to rely on spatially localized approximations. While global approximations can have high formal orders of accuracy, their cost is typically high. This is due both to high operation counts and to costly data flow on modern computers with hierarchical memory structures.

Historically, one can recognize an evolutionary path FD ⇒ PS ⇒ RBF ⇒ RBF-FD that starts by extending from FD methods (first applied to PDEs just over a century ago: see Richardson 1911) to PS methods. It transpires that each PS method can be seen as a special case of an RBF approximation in a certain limit. With the RBF representation, geometric flexibility has been achieved. When RBFs are then used to create weights for scattered-node FD-like stencils (*i.e.*, RBF-FD approximations), approximations have again become 'local', with associated high computational speeds and excellent scaling properties for massively large problem sizes.

Concerning interpolation over scattered nodes, using standard basis functions, the following theorem may at first appear discouraging.

Mairhuber–Curtis theorem (Curtis 1959, Mairhuber 1956). Given any set of basis functions $\{F_k(\mathbf{x}), k = 1, 2, \ldots, N\}$ with $\mathbf{x} \in R^d, d \geq 2$, the problem of determining an interpolant

$$s(\mathbf{x}) = \sum_{k=1}^{N} \lambda_k F_k(\mathbf{x}), \tag{2.1}$$

satisfying $s(\mathbf{x}_k) = f_k$, is singular for infinitely many configurations of distinct nodes $\mathbf{x}_k, k = 1, 2, \ldots, N$.

Proof. The interpolation requirement $s(\mathbf{x}_k) = f_k$ implies that the coefficients λ_k in (2.1) will satisfy the linear system

$$\begin{bmatrix} F_1(\mathbf{x}_1) & F_2(\mathbf{x}_1) & \cdots & F_N(\mathbf{x}_1) \\ F_1(\mathbf{x}_2) & F_2(\mathbf{x}_2) & \cdots & F_N(\mathbf{x}_2) \\ \vdots & \vdots & & \vdots \\ F_1(\mathbf{x}_N) & F_2(\mathbf{x}_N) & \cdots & F_N(\mathbf{x}_N) \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix}. \tag{2.2}$$

In more than one dimension, it is possible to move the nodes continuously so that two nodes end up interchanged, without them having coincided at

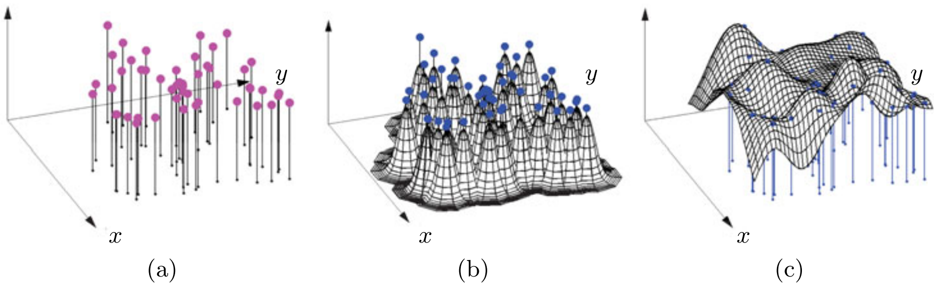


Figure 2.1. Graphical illustration of the RBF concept. (a) Example of two-dimensional scattered data. (b) Basis function set. One rotated Gaussian is located at each data point. (c) The unique linear combination of the Gaussians that agrees with all the provided data.

any time. The effect on the coefficient matrix in (2.2) is that two rows have become interchanged, that is, its determinant has changed sign. By continuity, the determinant must therefore have been zero somewhere along the way. □

The consequence of the theorem above is that vast numbers of seemingly 'innocent' node configurations will give rise to singular systems. The RBF idea for overcoming this issue is sketched in Figure 2.1. The basis functions are radially symmetric here, typically with one centred at each node point $\mathbf{x}_k$, that is, of the form $\phi(\|\mathbf{x}-\mathbf{x}_k\|)$. Here ϕ is a *radial function* (such as $\phi(r=\|\mathbf{x}-\mathbf{x}_k\|)=\mathrm{e}^{-(\varepsilon r)^2}$), ε is a *shape parameter*, and the norm is the standard Euclidean distance function. Again, letting the data value be f_k at node $\mathbf{x}_k, k=1,2,\ldots,N$, the coefficients in the RBF interpolant of $f(\mathbf{x})$,

$$s(\mathbf{x})=\sum_{k=1}^{N}\lambda_k\phi(\|\mathbf{x}-\mathbf{x}_k\|), \tag{2.3}$$

can be found by solving a system very similar to (2.2):

$$\begin{bmatrix} \phi(\|\mathbf{x}_1-\mathbf{x}_1\|) & \phi(\|\mathbf{x}_1-\mathbf{x}_2\|) & \cdots & \phi(\|\mathbf{x}_1-\mathbf{x}_N\|) \\ \phi(\|\mathbf{x}_2-\mathbf{x}_1\|) & \phi(\|\mathbf{x}_2-\mathbf{x}_2\|) & \cdots & \phi(\|\mathbf{x}_2-\mathbf{x}_N\|) \\ \vdots & \vdots & & \vdots \\ \phi(\|\mathbf{x}_N-\mathbf{x}_1\|) & \phi(\|\mathbf{x}_N-\mathbf{x}_2\|) & \cdots & \phi(\|\mathbf{x}_N-\mathbf{x}_N\|) \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix}. \tag{2.4}$$

Moving two nodes so that they change places again interchanges two rows but now also two columns, leaving the sign of the determinant unaffected. Therefore, the singularity argument above no longer applies. The key difference from the assumptions in the Mairhuber–Curtis theorem is that the basis functions $\phi(\|\mathbf{x}-\mathbf{x}_k\|)$ depend on the node locations.

Table 2.1. Some common choices for radial functions.

Type of basis function	Radial function $\phi(r)$
Piecewise smooth RBFs	
Polyharmonic spline (PHS)	$r^m, m = 1, 3, 5, \ldots$
	$r^m \log(r), m = 2, 4, 6, \ldots$
Compact support ('Wendland')	$(1 - \varepsilon r)_+^m p(\varepsilon r)$, p certain polynomials
Infinitely smooth RBFs	
Gaussian (GA)	$\mathrm{e}^{-(\varepsilon r)^2}$
Multiquadric (MQ)	$\sqrt{1 + (\varepsilon r)^2}$
Inverse quadratic (IQ)	$1/(1 + (\varepsilon r)^2)$
Inverse multiquadric (IMQ)	$1/\sqrt{1 + (\varepsilon r)^2}$
Bessel (BE) $(d = 1, 2, \ldots)$	$J_{d/2-1}(\varepsilon r)/(\varepsilon r)^{d/2-1}$

2.1. *Different RBF types*

Table 2.1 lists a number of RBF types. For most of these, we will show in Section 2.2 that the system (2.4) can never be singular, for any number of (distinct) nodes scattered in any number of dimensions.

2.1.1. *Piecewise smooth RBFs*

The listed 'piecewise smooth' radial functions will cause a singularity at the origin of the associated RBF and, in the compactly supported 'Wendland' case, also at $r = 1/\varepsilon$. This is entirely acceptable in many applications, but puts them at a disadvantage in other cases, such as when seeking accurate solutions to convection-type PDEs over long times (Fornberg and Piret 2008). The property of compactly supported RBFs to produce sparse rather than full linear systems is advantageous in some contexts such as image rendering, but less so when approximating PDEs, since the differentiation matrices that result from them nevertheless become full matrices.

PHS-type RBFs are associated with several optimality results, such as interpolating scattered data with the least possible overall curvature (Duchon 1977, Powell 1992). They are also of particular interest in the context of RBF-FD.

It can be noted that $\phi(r) = r^3$ in one dimension reproduces cubic splines, albeit with highly unusual end conditions. With slight modifications in the form of (2.3), one can, however, obtain either 'natural' or 'not-a-knot' splines. Similar modifications can also be applied to other RBF types and for scattered nodes in higher dimensions, offering easy-to-apply approaches for enhancing the accuracy at domain boundaries (Fornberg, Driscoll, Wright and Charles 2002).

2.1.2. Infinitely smooth RBFs

As noted above, $\phi(r) = r^3$ in one dimension leads to a cubic spline, featuring a jump in the third derivative at each node. Disregarding possible boundary effects, its accuracy is well known to be $O(h^4)$ on a grid with spacing h. Similarly, $\phi(r) = r^5$ leads to $O(h^6)$-errors, *etc.* This raises the obvious question why one would use radial functions that cause jumps in any derivative. For the infinitely smooth ones, there are no such jumps, and that suffices to obtain *spectral accuracy* – better than any algebraic order $O(h^p)$, $p \in \mathbb{N}$, assuming that no counterpart to the polynomial Runge phenomenon arises (Madych and Nelson 1992).

All smooth radial functions (of which there are many more options than are listed in Table 2.1) feature a shape parameter, denoted by ε. While one could also apply an ε-scaling to the PHS functions, such as $\phi(r) = r^m$, that is, use $\phi(r) = (\varepsilon r)^m$, this would serve no purpose since $(\varepsilon r)^m = \varepsilon^m r^m$, and the scale factor would then vanish analytically by the time the interpolant $s(\mathbf{x})$ is obtained.

2.2. Non-singularity theorems

Following Bochner (1933), we will first show that the RBF matrix

$$A = \begin{bmatrix} \phi(\|\mathbf{x}_1 - \mathbf{x}_1\|) & \phi(\|\mathbf{x}_1 - \mathbf{x}_2\|) & \cdots & \phi(\|\mathbf{x}_1 - \mathbf{x}_N\|) \\ \phi(\|\mathbf{x}_2 - \mathbf{x}_1\|) & \phi(\|\mathbf{x}_2 - \mathbf{x}_2\|) & \cdots & \phi(\|\mathbf{x}_2 - \mathbf{x}_N\|) \\ \vdots & \vdots & & \vdots \\ \phi(\|\mathbf{x}_N - \mathbf{x}_1\|) & \phi(\|\mathbf{x}_N - \mathbf{x}_2\|) & \cdots & \phi(\|\mathbf{x}_N - \mathbf{x}_N\|) \end{bmatrix} \tag{2.5}$$

is guaranteed to be non-singular for GA RBFs, no matter how the nodes (assumed to be distinct) are scattered in any number of dimensions. This result will then be generalized to several other RBF types.

2.2.1. Gaussian RBFs

A real symmetric matrix A is positive definite if and only if $\boldsymbol{\alpha}^T A \boldsymbol{\alpha} > 0$ for every real vector $\boldsymbol{\alpha} \neq \mathbf{0}$. All eigenvalues are then positive, and the matrix will be non-singular. The proof that the A-matrix for GA RBFs is positive definite can be carried out in three steps.

Step 1: Recall the Fourier transform of Gaussians. We define the one-dimensional Fourier transform (FT) as

$$u(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \widehat{u}(\omega)\ \mathrm{e}^{\mathrm{i}\omega x}\, \mathrm{d}\omega,$$
$$\widehat{u}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} u(x)\ \mathrm{e}^{-\mathrm{i}\omega x}\, \mathrm{d}x.$$

Applying the one-dimensional result

$$u(x) = \mathrm{e}^{-\varepsilon^2 x^2} \Leftrightarrow \widehat{u}(\omega) = \frac{1}{\sqrt{2}\varepsilon}\ \mathrm{e}^{-\omega^2/(4\varepsilon^2)}$$

d times, we obtain the d-dimensional case

$$u(\mathbf{x}) = \mathrm{e}^{-\varepsilon^2\|\mathbf{x}\|^2} \Leftrightarrow \widehat{u}(\boldsymbol{\omega}) = \frac{1}{2^{d/2}\varepsilon^d}\ \mathrm{e}^{-\|\boldsymbol{\omega}\|^2/(4\varepsilon^2)}. \tag{2.6}$$

Inverting $\widehat{u}(\boldsymbol{\omega})$ back to physical space produces the identity

$$\mathrm{e}^{-\varepsilon^2\|\mathbf{x}\|^2} = \frac{1}{(2\pi)^{d/2}} \int_{R^d} \frac{1}{2^{d/2}\varepsilon^d} \mathrm{e}^{-\|\boldsymbol{\omega}\|^2/(4\varepsilon^2)}\ \mathrm{e}^{\mathrm{i}\,\mathbf{x}\cdot\boldsymbol{\omega}}\, \mathrm{d}\boldsymbol{\omega}. \tag{2.7}$$

It may at first seem that this way to rewrite the GA radial function $\mathrm{e}^{-\varepsilon^2\|\mathbf{x}\|^2}$ has introduced a lot of extra complexity. However, the key point will turn out to be that $\mathbf{x}$, appearing quadratically as $\|\mathbf{x}\|^2$ in the exponent in the left-hand side, appears only linearly, as $\mathbf{x}$, in one of the exponents in the right-hand side.

Step 2: Proof that *A* is positive semidefinite. Let

$$\boldsymbol{\alpha}\ = [\alpha_1, \ldots, \alpha_N]^T \neq \mathbf{0}.$$

Then

$$\begin{aligned}
\boldsymbol{\alpha}^T A\, \boldsymbol{\alpha} &= \sum_{j=1}^{N}\sum_{k=1}^{N} \alpha_j\alpha_k \mathrm{e}^{-\varepsilon^2\|\mathbf{x}_j-\mathbf{x}_k\|^2} \quad \text{(apply (2.7))}\\
&= \sum_{j=1}^{N}\sum_{k=1}^{N} \alpha_j\alpha_k \frac{1}{(2\pi)^{d/2}} \int_{R^d} \frac{1}{(2\varepsilon^2)^{d/2}} \mathrm{e}^{-\|\boldsymbol{\omega}\|^2/(4\varepsilon^2)}\ \mathrm{e}^{\mathrm{i}\,(\mathbf{x}_j-\mathbf{x}_k)\cdot\boldsymbol{\omega}}\, \mathrm{d}\boldsymbol{\omega}\\
&= \frac{1}{(2\varepsilon)^d\pi^{d/2}} \int_{R^d} \mathrm{e}^{-\|\boldsymbol{\omega}\|^2/(4\varepsilon^2)} \left(\sum_{j=1}^{N}\sum_{k=1}^{N} \alpha_j\alpha_k\ \mathrm{e}^{\mathrm{i}\,(\underline{x}_j-\underline{x}_k)\cdot\boldsymbol{\omega}} \right) \mathrm{d}\boldsymbol{\omega}.
\end{aligned}$$

The double sum inside the integral can be written as

$$\left(\sum_{j=1}^{N} \alpha_j\ \mathrm{e}^{\mathrm{i}\mathbf{x}_j\cdot\boldsymbol{\omega}}\right) \overline{\left(\sum_{k=1}^{N} \alpha_k\ \mathrm{e}^{\mathrm{i}\mathbf{x}_k\cdot\boldsymbol{\omega}}\right)} = \left\| \sum_{m=1}^{N} \alpha_m\ \mathrm{e}^{\mathrm{i}\mathbf{x}_m\cdot\boldsymbol{\omega}} \right\|^2 \geq 0.$$

Thus $\boldsymbol{\alpha}^T A\, \boldsymbol{\alpha} \geq 0$, and we have shown that the matrix A is positive semidefinite.

Step 3: Proof that A is positive definite. Based on the result above, it only remains to show that

$$\sum_{m=1}^{N} \alpha_m\, \mathrm{e}^{\mathrm{i}\, \mathbf{x}_m \cdot \boldsymbol{\omega}}$$

cannot be identically zero (as a function of $\boldsymbol{\omega}$) unless all the coefficients α_m are zero. Several different short proofs for this are available (Fasshauer 2007, Fornberg and Flyer 2015*b*, Powell 1992).

2.2.2. Some other RBF types

If (2.6) is replaced by

$$u(\mathbf{x}) = f(\varepsilon \|\mathbf{x}\|) \Leftrightarrow \widehat{u}(\boldsymbol{\omega}) = g(\|\boldsymbol{\omega}\|/\varepsilon)$$

with $g(\|\boldsymbol{\omega}\|) > 0$, the replacement for the leading factor $(\mathrm{e}^{-\|\boldsymbol{\omega}\|^2/(4\varepsilon^2)})$ inside the integral in (2.7) will again be positive, and the positive definiteness proof will carry through just as in the GA case. This situation arises for many types of compactly supported RBFs, for example.

Another variation of the non-singularity proof (related to the theory of *completely monotone functions*: Schoenberg 1938), proceeds as follows. Taking the inverse Laplace transform of $\phi(\sqrt{r})$ for different radial functions $\phi(r)$ gives formulas such as

$$\begin{array}{lll} \text{IQ} & \dfrac{1}{1+(\varepsilon r)^2} = \displaystyle\int_0^\infty \mathrm{e}^{-s} \mathrm{e}^{-s(\varepsilon r)^2}\, \mathrm{d}s, \\ \text{IMQ} & \dfrac{1}{\sqrt{1+(\varepsilon r)^2}} = \displaystyle\int_0^\infty \frac{\mathrm{e}^{-s}}{\sqrt{\pi s}} \mathrm{e}^{-s(\varepsilon r)^2}\, \mathrm{d}s. \end{array}$$

In all cases when the factor in front of $\mathrm{e}^{-s(\varepsilon r)^2}$ inside the integral is positive, we observe that (using IQ as an illustration)

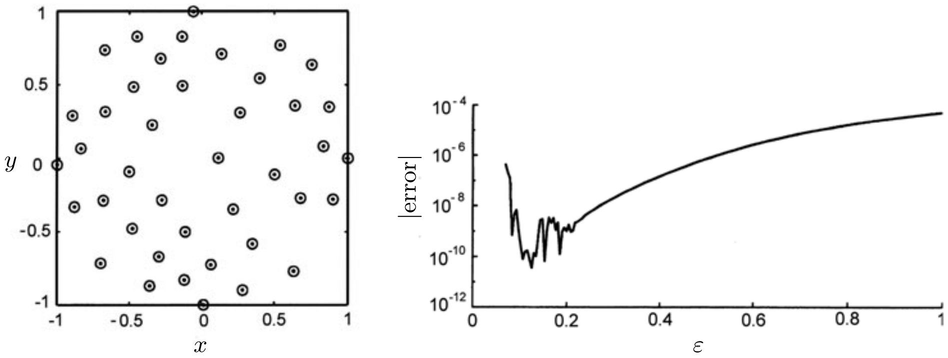
$$\begin{aligned} \boldsymbol{\alpha}^T A \boldsymbol{\alpha} &= \sum_{j=1}^{N} \sum_{k=1}^{N} \alpha_j \alpha_k \frac{1}{1+\varepsilon^2 \|\mathbf{x}_j - \mathbf{x}_k\|^2} \\ &= \int_0^\infty \mathrm{e}^{-s} \left(\sum_{j=1}^{N} \sum_{k=1}^{N} \alpha_j \alpha_k \mathrm{e}^{-s\, \varepsilon^2 \|\mathbf{x}_j - \mathbf{x}_k\|^2} \right) \mathrm{d}s. \end{aligned}$$

From the non-singularity proof for GA RBFs, we know that the double sum is positive whenever the vector $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_N]^T$ is not identically zero. Therefore, the integral and thus the quantity $\boldsymbol{\alpha}^T A \boldsymbol{\alpha}$ will also be positive, that is, A is a positive definite matrix.

The proofs above do not directly apply to the commonly used MQ case. It transpires, however, that non-singularity is again assured, with the (symmetric) A-matrix now having one positive eigenvalue and all the remaining ones negative. The original proof by Micchelli (1986) was later simplified in Powell (2005).

3. Near-flat RBFs

With ε available as a free parameter, it is natural to explore how the choice of ε influences the accuracy obtained. A typical test is shown in Figure 3.1. As first noted by Tarwater (1985), it often happens that the error decreases rapidly with ε until the calculation suddenly breaks down due to the increasing ill-conditioning of the linear system (2.4). This may suggest that a trade-off will be required between accuracy and numerical conditioning (described as an 'uncertainty principle' in Schaback 1995). It was soon realized, however, that the RBF interpolation problem actually does not become ill-conditioned in this flat basis function limit, and that the apparent problem was particular to the *RBF-Direct* procedure: solution of (2.4) followed by evaluation of (2.3). RBF-Direct uses ill-conditioned expansion coefficients λ_k as intermediate quantities for arriving at what should be a well-conditioned result (Driscoll and Fornberg 2002, Fornberg, Wright and Larsson 2004). Several well-conditioned stable numerical algorithms were



(a)

(b)

Figure 3.1. (a) A set of 41 scattered nodes in the unit circle. (b) The error in the max norm when the test function $f(x,y) = 59/\big(67 + \big(x + \frac{1}{7}\big)^2 + \big(y - \frac{1}{11}\big)^2\big)$ is interpolated using these nodes, displayed as a function of the shape parameter ε.

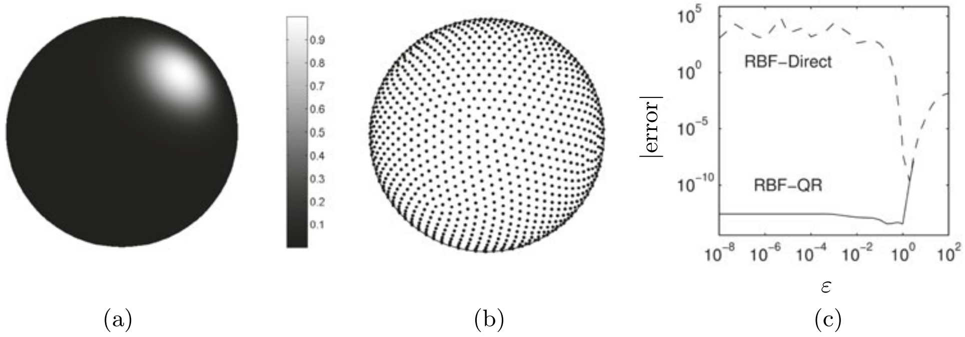


Figure 3.2. (a) Test function $f(x) = \exp\big(-7(x+\frac{1}{2})^2 - 8(y+\frac{1}{2})^2 - 9(z-\frac{1}{\sqrt{2}})^2\big)$. (b) $N = 1849$ ME (minimal energy) nodes on the surface of the unit sphere. (c) MQ interpolation errors (in the max norm), as functions of ε, when using RBF-Direct versus using the stable RBF-QR algorithm. The RBF-QR error level seen here for small ε is unrelated to the nearby machine rounding level of 10^{-16}.

subsequently developed (see Section 3.2), giving results as seen in one typical case in Figure 3.2. Sometimes, the most accurate ε-range can be reached with RBF-Direct. In other cases, such as the one illustrated here, this requires a stable algorithm.

If the nodes are lattice-based, it can happen that the RBF interpolant diverges when $\varepsilon \to 0$ (Fornberg, Larsson and Wright 2006, Fornberg and Wright 2004), although never in the GA case (Schaback 2005), a fact that contributes to making GA a popular RBF choice. For node sets with some irregularity, the interpolant will in the flat $\varepsilon \to 0$ limit take the form of a multivariate polynomial (Driscoll and Fornberg 2002, Fornberg *et al.* 2004). One reason that small ε is often better than $\varepsilon \to 0$ is that, with RBF interpolants converging to polynomials, the boundary accuracy often deteriorates due to the Runge phenomenon (Fornberg and Zuev 2007). In the high-degree polynomial case, Chebyshev-style node clustering near the boundaries is the most frequently used remedy (in spite of disadvantages, such as causing adverse stability conditions in the context of explicit time-stepping of PDEs). As was noted in Section 2.1.1, a number of additional options are available for RBFs.

3.1. The ill-conditioning of the A-matrix

Sideways translates of near-flat basis functions all look the same, and it is intuitively obvious that they must form a very ill-conditioned base to expand in. Just how bad it is can readily be quantified (Fornberg and Zuev 2007). For example, when using infinitely smooth RBFs on scattered nodes in two dimensions, the eigenvalues of the A-matrix form distinct groups, following

Table 3.1. Numbers of eigenvalues of different sizes (powers of ε) for different geometries and types of shape parameter.

Geometry	Power of ε								
	0	2	4	6	8	10	12	14	...
1-D non-periodic	1	1	1	1	1	1	1	1	...
1-D on circle periphery	1	2	2	2	2	2	2	2	...
2-D non-periodic	1	2	3	4	5	6	7	8	...
2-D on spherical surface	1	3	5	7	9	11	13	15	...
3-D non-periodic	1	3	6	10	15	21	28	36	...

the specific pattern

$$\begin{aligned}
&\{O(1)\},\\
&\{O(\varepsilon^2), O(\varepsilon^2)\},\\
&\{O(\varepsilon^4), O(\varepsilon^4), O(\varepsilon^4)\},\\
&\{O(\varepsilon^6), O(\varepsilon^6), O(\varepsilon^6), O(\varepsilon^6)\},\\
&\dots
\end{aligned} \tag{3.1}$$

until the last eigenvalue is reached, causing the last group to possibly contain fewer eigenvalues than the general pattern would suggest. Different choices of scattered-node locations or of RBF types (IQ, MQ, or GA), make no difference in this regard. However, use of lattice-based nodes or Bessel-type RBFs result in exceptions (with smaller groups, implying worse conditioning). More concisely, we can write the eigenvalue pattern above as

$$1, 2, 3, 4, \dots, \tag{3.2}$$

indicating how many eigenvalues there are of orders $\varepsilon^0, \varepsilon^2, \varepsilon^4, \varepsilon^6$, *etc.* Table 3.1 shows some more such sequences. The patterns are readily recognizable: for example, in the d-dimensional non-periodic case, the kth entry is $\binom{d+k-2}{k-1}$. Given these patterns, one can immediately calculate the orders of both $\operatorname{cond}(A)$ and $\det(A) = \prod_{k=1}^{n} \lambda_k$ as functions of n (here the λ_k denote the eigenvalues of A). For the examples in Figures 3.1 and 3.2, $\operatorname{cond}(A)$ becomes equal to $O(\varepsilon^{-16})$ and $O(\varepsilon^{-84})$, respectively.

3.2. Overview of some stable algorithms

The most straightforward approach for calculating in the small ε regime is to use extended precision arithmetic. The main drawback is that the cost usually becomes excessive. Given the results quoted in Section 3.1, one can determine in advance just how many digits of precision would be needed as functions of N and ε in various geometrical settings. For example, in the case shown in Figure 3.2, lowering the ε-value for onset of ill-conditioning by a factor of 100 (about what is needed in this case to 'safely' reach the optimal accuracy range) increases cond(A) by a factor of $100^{84} = 10^{168}$, showing that the arithmetic precision would have to be increased from 16 to about 180 digits.

Some types of preconditionings and SVD enhancements have been suggested for the RBF-Direct approach. While preconditioning can speed up certain iterative procedures (see Fasshauer 2007, Chapter 34), this does not address the issue that significant information has already been lost when the coefficient matrix A is formed (with all its entries virtually the same when ε is small). Recovery of such missing information is challenging or impossible.

Stable algorithms produce the same interpolant $s(\mathbf{x})$ as mathematically defined by (2.3) and (2.4), but without involving the ill-conditioned expansion coefficients λ_k. By using only computational steps that remain well-conditioned even when $\varepsilon \to 0$, standard double-precision arithmetic suffices. So far, two main classes of stable algorithms have been developed. The first realizations of these were denoted *Contour-Padé* (Fornberg and Wright 2004) and *RBF-QR* (Fornberg and Piret 2007), respectively. Related to the latter is the recent *RBF-GA* algorithm (Fornberg, Lehto and Powell 2013).

3.2.1. Contour-Padé algorithm

Although ε is typically a real-valued quantity, it can be extended to complex values. Focusing on the GA case, it can be shown that the interpolant $s(\mathbf{x}, \varepsilon)$, for any fixed evaluation point $\mathbf{x}$, then becomes a meromorphic function of ε (*i.e.*, with poles as its only singularities across the finite complex ε-plane). Furthermore, it is known that $s(\mathbf{x}, 0)$ is finite even as $\varepsilon \to 0$. The origin $\varepsilon = 0$ must therefore be a removable singularity of $s(\mathbf{x}, \varepsilon)$. The actual algorithm requires a number of technicalities to be addressed, but its key principle is that Cauchy's integral theorem allows the evaluation of an analytic function at a point (such as $\varepsilon = 0$) using an integration path that does not need to come anywhere close to it, that is, the path can follow such a large circle around the origin in the ε-plane that RBF-Direct can safely be used along it. In its original form, the Contour-Padé algorithm is now mostly of historical interest, having established the feasibility of stable algorithms.

3.2.2. RBF-QR algorithm

As we have noted repeatedly, translates of near-flat RBFs form a basis that is ill-suited for immediate numerical use. This naturally raises the question whether the underlying approximation *space* is also bad, or if the conditioning issue can be resolved by finding an alternative good basis in exactly the same space. The latter turns out to be the case, leading to the follow-up issue of how one can carry out the basis conversion by analytic means also in scattered-node cases, that is, so that no numerical cancellations will arise anywhere in the process.

One can draw a parallel to the set of monomials $P = \{1, x, x^2, \ldots, x^{100}\}$ versus Chebyshev polynomials $T = \{T_0, T_1, T_2, \ldots, T_{100}\}$ over $x \in [-1, 1]$. Both sets span exactly the same function space, yet the monomials are an ill-conditioned base. For numerical work, it is critical that the Chebyshev polynomials are available in some type of closed form, for example, as $T_n(x) = \cos(n \arccos x)$, or through a three-term recursion, and need not be obtained by numerically forming different linear combinations of the monomials.

The RBF-QR method offers a systematic approach for converting a set of near-flat basis functions with scattered centres to a well-conditioned base for exactly the same space, in a numerically stable manner. It was first implemented for nodes on the surface of a sphere (Fornberg and Piret 2007), and more recently (in the special case of GA RBFs) for arbitrary node sets in one, two, and three dimensions (Fornberg, Larsson and Flyer 2011, Larsson, Lehto, Heryudono and Fornberg 2013).

3.2.3. RBF-GA algorithm

The RBF-QR algorithm involves extensive manipulations of power series expansions. Rather than expanding to the extent that remainders can be ignored, the RBF-GA algorithm utilizes shorter expansions combined with exact remainder formulas, for GA RBFs expressible in terms of incomplete gamma functions. This leads to a stable algorithm that is free from both infinite expansions and inexact truncations. It applies to GA RBFs in any number of dimensions, and is at present both the algebraically simplest and the computationally fastest stable option available (at around 10 times the cost of RBF-Direct, in either two or three dimensions). Although it may be slightly less accurate than RBF-QR in some cases (such as for large lattice-like node sets), it is nevertheless well-suited for generating RBF-FD approximations.

4. Three examples of solving PDEs using global RBFs

The three examples illustrate implementation issues and resulting accuracies, as well as how PDE complexity has been increased over the last decade, from Poisson's equation in a simple two-dimensional domain to a nonlinear time-dependent PDE system describing mantle convection in a three-dimensional spherical shell. In the former case, perfectly well-understood solutions were reproduced, whereas in the latter case, it provided physical insights not previously reached by any other investigative method.

4.1. Poisson's equation

We consider as our test problem Poisson's equation on a domain Ω, with a Dirichlet condition on the boundary $\partial\Omega$:

$$\begin{aligned} u(\mathbf{x}) &= g(\mathbf{x}) \text{ on boundary } \partial\Omega, \\ \Delta u(\mathbf{x}) &= f(\mathbf{x}) \text{ in interior of } \Omega. \end{aligned} \tag{4.1}$$

This is discretized at node locations $\mathbf{x}_1, \ldots, \mathbf{x}_{N_B}$ on $\partial\Omega$ and $\mathbf{x}_{N_B+1}, \ldots, \mathbf{x}_N$ within Ω.

4.1.1. Two strategies for RBF discretization

The two main discretization approaches can be summarized as follows (described in the two-dimensional Poisson case for simplicity).

Kansa's formulation. Let the solution to (4.1) be of the form

$$u(\mathbf{x}) = \sum_{j=1}^{N} \lambda_j \phi(\|\mathbf{x} - \mathbf{x}_j\|). \tag{4.2}$$

Enforcing this at all nodes gives a linear system for the λ_j of the following structure:

$$\begin{bmatrix} \phi(\|\mathbf{x} - \mathbf{x}_j\|)|_{\mathbf{x}=\mathbf{x}_i} \\ ---------- \\ \triangle\phi(\|\mathbf{x} - \mathbf{x}_j\|)|_{\mathbf{x}=\mathbf{x}_i} \end{bmatrix} \begin{bmatrix} \underline{\lambda} \end{bmatrix} = \begin{bmatrix} \underline{g} \\ - \\ \underline{f} \end{bmatrix}, \tag{4.3}$$

where $i = 1, \ldots, N_B$ for the upper matrix block, and $i = N_B + 1, \ldots, N$ for the lower block. This straightforward approach has proved to be widely successful, even if rare possibilities for singularities have been noted (Hon and Schaback 2001).

Symmetric formulation. The assumed form of the solution is now changed from (4.2) to

$$u(\mathbf{x}) = \sum_{j=1}^{N_B} \lambda_j \phi(\|\mathbf{x} - \mathbf{x}_j\|) + \sum_{j=N_B+1}^{N} \lambda_j \triangle\phi(\|\mathbf{x} - \mathbf{x}_j\|),$$

that is, we use $\triangle\phi(\|\mathbf{x} - \mathbf{x}_j\|)$ rather than $\phi(\|\mathbf{x} - \mathbf{x}_j\|)$ as RBF at the interior

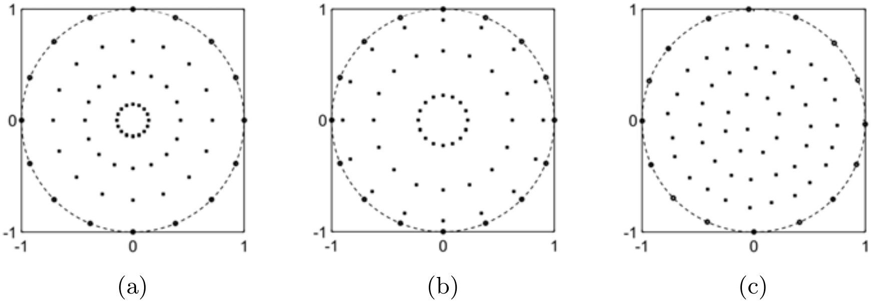


Figure 4.1. How polar-type grids create highly non-uniform resolutions in different directions near the origin. (a) Polar, equispaced in radius. (b) Polar, Chebyshev along each diameter. (c) Irregular (but avoiding clustering), as typically used in RBF contexts.

nodes. The counterpart to (4.3) becomes (in abbreviated notation)

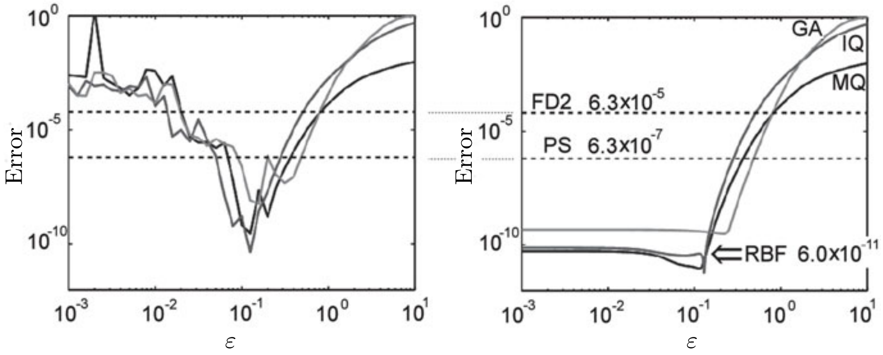
$$\left[\begin{array}{c|c} \phi & \triangle\phi \\ \hline \triangle\phi & \triangle^2\phi \end{array}\right]\left[\underline{\lambda}\right] = \left[\begin{array}{c} \underline{g} \\ \hline \underline{f} \end{array}\right],$$

with (for the standard RBF choices) a guaranteed symmetric and positive definite coefficient matrix (Fasshauer 1997, Wu 1992). Although this is an obvious advantage, actual numerical performance of the two approaches seems relatively comparable, with different studies suggesting slight advantages either way, for example, Larsson and Fornberg (2003) and Power and Barraco (2002).

Generalizations to other linear or nonlinear operators are straightforward. If Newton's method is used, the cost per iteration becomes comparable to that of solving a linear case, as either will require the solution of a full $N \times N$ linear system.

4.1.2. Test calculation: circular domain

Naturally, the earliest implementations of RBFs for PDEs were focused on showing that the approach is viable for very simple test problems. We summarize the study by Larsson and Fornberg (2003), since this also compared RBF-Direct with Contour-Padé (the only stable choice in 2003). In order to allow easy comparisons of RBFs against FD2 (second-order FD) and PS methods, the domain was chosen as the unit circle: see Figure 4.1. All the node sets had $N_B = 16$ nodes on the boundary $\partial\Omega$ and $N_I = 48$ nodes in the interior of Ω. For FD2, the nodes were equispaced in both angle and



(a) (b)

Figure 4.2. Max norm errors, as functions of ε, when solving a two-dimensional Poisson test problem using three choices of RBFs, GA, IQ and MQ, using (a) RBF-Direct and (b) Contour-Padé. The dashed lines across the two subplots compare the accuracies reached by FD2, PS (both independent of ε).

radius, and for PS again equispaced in angle, but of Chebyshev type radially (across $-1 \leq r \leq 1$, with angle $0 \leq \theta < \pi$). For RBF, the nodes were somewhat irregularly scattered. Figure 4.2 shows a typical result. Here Kansa's approach is applied to (4.1) with $g(\mathbf{x})$ and $f(\mathbf{x})$ selected in such way that the equation has as its solution $u(\mathbf{x}) = 100/(100 + (x - 0.2)^2 + 2y^2)$. Even when using RBF-Direct, the RBF approach is seen to be the most accurate option (if the optimal ε is used). The use of a stable algorithm not only improves the accuracy further still, but also makes the choice of 'optimal' ε very much less critical.

It can be noted that a second-order method (such as FD2, or second-order finite elements) gains a factor of 4 in accuracy when step sizes are halved, that is, in two dimensions when four times as many nodes are used. The error is then inversely proportional to the number of nodes. In the present test case, the errors for MQ and IQ RBFs are roughly 10^{-6} times those for FD2, implying that, in order to match the RBF accuracy, FD2 would need the node count $N = 64$ to be increased by a factor of about one million.

4.2. Reaction–diffusion equations on curved surfaces

Solving PDEs over curved surfaces has a substantial history, both in terms of application areas and with regard to numerical approaches. Some different methods (including RBFs) are discussed in Shankar, Wright, Fogelson and Kirby (2013). RBFs are particularly well-suited to the task, since they avoid the singularities that are intrinsic to any surface-bound coordinate system, exemplified for a sphere with the two poles if using spherical coordinates. Another key advantage is that spectral accuracy readily becomes available

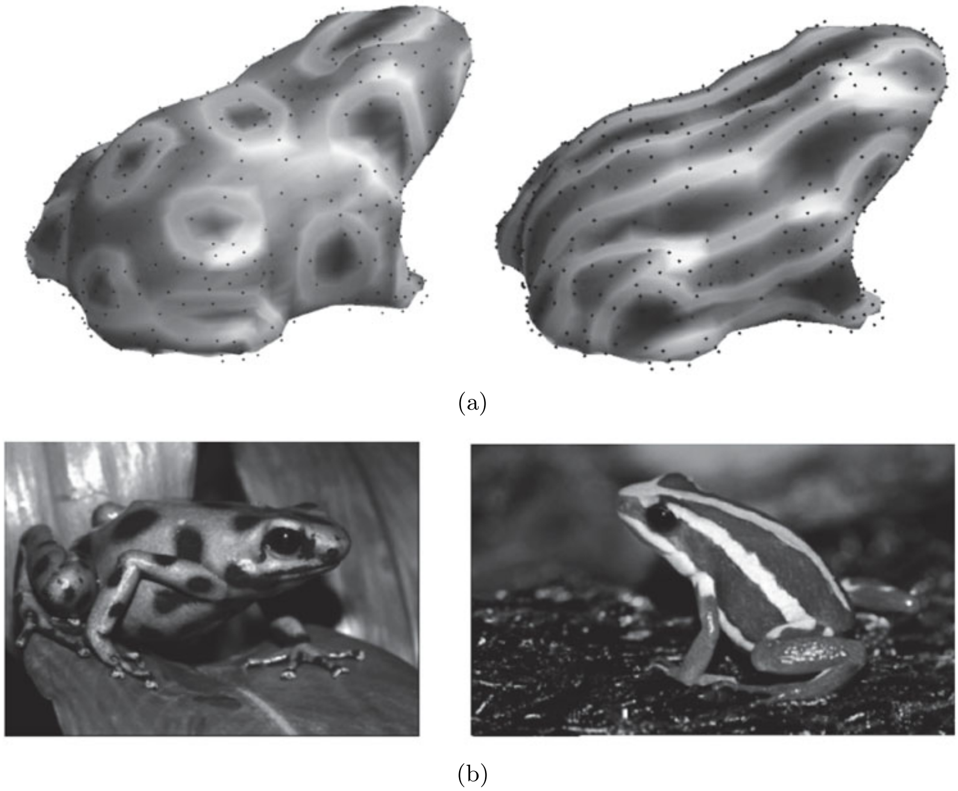


Figure 4.3. (a) Patterns produced by RBF solutions of the Brusselator reaction–diffusion equations for two different parameter settings. (b) The skin patterns on two frog species (Tabasara rain frog, and poison dart frog, respectively). Illustration provided by Cécile Piret.

(in contrast to surface-triangularization-based finite element discretizations, for example).

The solution of PDEs over biological surfaces was pioneered by Turing (1952) in the context of pattern formation on animals. Both this topic, and also other processes occurring on cell surfaces and on other types of biological membranes, have since received extensive mathematical and numerical attention. The solutions presented in Piret (2012) use global RBFs, in combination with the *orthogonal gradient method* (OGr), allowing a single 'cloud' of nodes to be used both for defining the surface and for discretizing the PDE. Figure 4.3 illustrates an $N = 560$ node set in the shape of a frog, and two RBF-generated solutions to the *Brusselator equations* over this surface. This nonlinear reaction–diffusion system closely models actual formation of skin patterns on animals (for which the time evolution gets frozen at some embryonic stage). The very high accuracy of the RBF approach is evident in Figure 4.3, as the finest resolved features have about four points per wavelength, to be compared to the theoretical limits of 2 for Fourier-PS and π for Chebyshev-PS.

Fuselier and Wright (2013) describe solutions to another convection–diffusion-type PDE (the *Barkley model*), again over surfaces of biological objects. The global RBF approach was in this case somewhat different (a 'projection' approach, for which the surfaces were given in the form of level surfaces of specified three-dimensional functions).

4.3. Mantle flow in a spherical shell

A number of increasingly large geoscience-oriented test cases were solved using global RBF-based spatial discretization between 2007 and 2011. The geometries were at first confined to the surface of a sphere (Flyer and Lehto 2010, Flyer and Wright 2007, Flyer and Wright, 2009), and then followed by a three-dimensional mantle convection simulation (Wright, Flyer and Yuen 2010). These works were all summarized in Flyer and Fornberg (2011). We highlight here the mantle flow simulation, since it decisively brought RBF-based simulations from 'just another approach that can work on toy problems' to (i) confirming a physical prediction previously outside numerical reach, and (ii) doing so using a PC, against supercomputer calculations employing the full range of traditional methodology approaches (see Table 4.1, which abbreviates a more extensive table in Wright *et al.* 2010).

The physical scenario is as follows: the flow is incompressible; the temperature (T) is governed by a mixed convective–diffusive PDE; the momentum is governed by Stokes flow, an elliptic PDE; the impermeable boundaries are slip-free (Neumann boundary conditions in the angular direction), with $T = 1$ at the core and $T = 0$ at the crust. The coupled system of three PDEs is approximated by RBF discretization on each of many concentric spherical shells, together with Chebyshev-PS discretization radially: see Figure 4.4(a). Since no analytic solutions are available, isoviscous flow at low $Ra = 7\,000$ (within the steady-state regime, where Ra is the Rayleigh number) has become a commonly used benchmark. The standard initial condition in this case is a combination of fourth-order spherical harmonics times linear decay in the radial direction. The summary in Table 4.1 compares results for the global variables, $Nu_{\text{crust}}, Nu_{\text{core}}, \langle V_{\text{RMS}} \rangle, \langle T \rangle$ (Nu is the Nusselt number, V_{RMS} is the root mean square velocity, and $\langle \cdot \rangle$ indicates globally averaged quantities). For this test, energy conservation implies that solutions should satisfy $Nu_{\text{crust}} = Nu_{\text{core}}$. The RBF-CH method, using a much lower level of discretization, achieves near-perfection in terms of accuracy compared to the previously most accurate method, the Romberg extrapolated SPH-FD method. The RBF-CH simulation was the only one that was run on standard PC hardware.

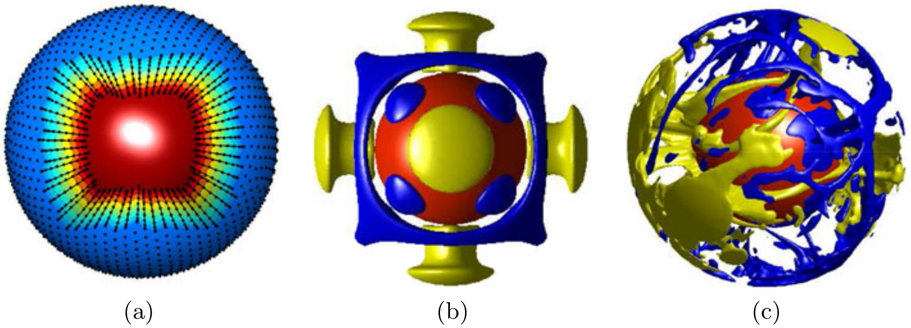


Figure 4.4. Mantle convection. (a) RBF-CH discretization. (b) Steady-state solution at $Ra = 7\,000$ (colour online: yellow, upwelling; blue, downwelling; red, core). (c) Snapshot of a $Ra = 10^6$ solution after a time corresponding to about 4.5 times the age of the Earth.

Table 4.1. Comparison of methods in the literature for the standard $Ra = 7\,000$ case.

Method	No. of nodes	Nu_{crust}	Nu_{core}	$\langle V_{\text{RMS}} \rangle$	$\langle T \rangle$	Reference
RBF-CH	36 800	3.6096	3.6096	31.0820	0.21578	[1]
SPH-FD	552 960	3.6086	–	31.0765	0.21582	[2]
SPH-FD	extrapolated	3.6096	3.6096	31.0821	0.21577	[2]
FE	393 216	3.6254	3.6016	31.09	0.2176	[3]
FV	663 552	3.5983	3.5984	31.0226	0.21594	[4]
FD	12 582 912	3.6083	–	31.0741	0.21639	[5]

[1] Wright *et al.* (2010)
[2] Harder and Hansen (2005)
[3] Zhong *et al.* (2008)
[4] Stemmer *et al.* (2006)
[5] Kameyama *et al.* (2008)

Figure 4.4(c) shows a snapshot from a $Ra = 10^6$ simulation, dominated by turbulent convection. This is a much more physically realistic case, since $Ra \approx 10^7$ for the current Earth. This RBF-CH simulation is the only spectral model in the literature to be run in spherical geometry at such a high Ra. It showed an instability at $Ra = 70\,000$ that had been theorized (Bercovici, Schubert, Glatzmaier and Zebib 1989) but remained controversial, as it had not been seen in any previous numerical simulations. These mantle flow simulations demonstrate strikingly that global RBFs can be very competitive even on standard PCs.

5. Basic properties of RBF-FD approximations

RBF-FD combines many key strengths of RBFs with those of traditional FD approximations. The primary factor behind their development was the high computational cost of global RBFs. When using RBF-Direct, finding an interpolant or calculating a differentiation matrix (DM) each cost $O(N^3)$ operations for N nodes, with an additional $O(N^2)$ operations each time a DM is applied (*e.g.*, during time-stepping). In parallel with the successful application usage of global RBFs, as described above, efforts were under way on several fronts to dramatically reduce these costs. Of several potentially viable approaches (such as ‘fast algorithms’ based on multipole ideas, innovative preconditioners, *etc.*), RBF-FD is at present the leading option. The rest of this article will be devoted to this.

The RBF-FD concept was first outlined in a conference presentation by Tolstykh (2000). Shortly afterwards it was introduced a number of times independently, for instance by Shu, Ding and Yeo (2003), Wang and Liu (2002) and Wright (2003). Since this approach is still in rapid development, the present discussion will not attempt to be comprehensive but only highlight how it has already proved to be highly competitive against previous alternatives. Active application areas not discussed here include elasticity (Kee, Liu and Lu 2008, Tolstykh and Shirobokov 2003), flame propagation (Bayona and Kindelan 2013, Kindelan, Bernal, Gonzalez-Rodriguez and Moscoso 2010), and mechanics (Chinchapatnam, Djidjeli, Nair and Tan 2009, Rodrigues, Roque and Ferreira 2013).

5.1. RBF-FD weights

Traditional FD approximations are grid-based and, when multidimensional, typically combine one-dimensional approximations. FD weights are determined so that the approximations become exact for polynomials of degree as high as possible. Some effective algorithms for generating FD weights are given in Fornberg (1998). The polynomial approach does not generalize well to scattered nodes in more than one dimension, with the Mairhuber–Curtis theorem being just one reason. Instead of relying on multivariate polynomials, one can enforce the exact result for all the RBFs that are centred at the nodes of the stencil of size n. Straightforward algebra will then show that the weights w_k at the stencil nodes $\mathbf{x}_k, k = 1, 2, \ldots, n$ can be obtained by solving the linear system

$$\begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix}\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} L\phi(\|\mathbf{x}-\mathbf{x}_1\|)|_{\mathbf{x}=\mathbf{x}_c} \\ L\phi(\|\mathbf{x}-\mathbf{x}_2\|)|_{\mathbf{x}=\mathbf{x}_c} \\ \vdots \\ L\phi(\|\mathbf{x}-\mathbf{x}_n\|)|_{\mathbf{x}=\mathbf{x}_c} \end{bmatrix}. \tag{5.1}$$

The matrix A is the same as the one in (2.4), and $\mathbf{x}_c$ is the location at which the stencil is approximating the L-operator (typically chosen as a node point near the stencil centre). A common generalization of (2.3) is to add multivariate polynomials to the RBF basis, and then impose matching constraints. For instance, if one includes linear terms in a two-dimensional case, (5.1) should be replaced by

$$\left[\begin{array}{ccc|ccc} & & & 1 & x_1 & y_1 \\ & A & & \vdots & \vdots & \vdots \\ & & & 1 & x_n & y_n \\ \hline 1 & \cdots & 1 & & & \\ x_1 & \cdots & x_n & & 0 & \\ y_1 & \cdots & y_n & & & \end{array}\right] \left[\begin{array}{c} w_1 \\ \vdots \\ w_n \\ \hline w_{n+1} \\ w_{n+1} \\ w_{n+3} \end{array}\right] = \left[\begin{array}{c} L\phi(\|\mathbf{x}-\mathbf{x}_1\|)|_{\mathbf{x}=\mathbf{x}_c} \\ \vdots \\ L\phi(\|\mathbf{x}-\mathbf{x}_n\|)|_{\mathbf{x}=\mathbf{x}_c} \\ \hline L\;1\;|_{\mathbf{x}=\mathbf{x}_c} \\ L\;x\;|_{\mathbf{x}=\mathbf{x}_c} \\ L\;y\;|_{\mathbf{x}=\mathbf{x}_c} \end{array}\right], \tag{5.2}$$

where only the weights $w_1, w_2, \ldots, w_n$ should be used. For a derivation, see (Fornberg and Flyer 2015*b*). The pattern in (5.2) generalizes directly to higher dimensions and different polynomial orders.

5.2. Node distributions

While Cartesian lattices are commonly used for FD and PS methods, hexagonal lattices (in case of two dimensions) generally allow for more cost-effective discretizations (see Section 6.2). Such lattices have been used only rarely in the past because of algebraic complexity, and difficulties with both local refinements and with generalizations to higher dimensions. When using RBFs, and especially RBF-FD, all these concerns vanish. Later in this article, Figure 6.8 will illustrate other advantages with deviating from Cartesian grids. Quasi-uniform scattered-node sets are often highly effective as well as easily generated. The Delaunay-based algorithm in Persson and Strang (2004) offers one convenient option. In the case of two dimensions, the algorithm described in Fornberg and Flyer (2015*a*) is particularly fast.

On Cartesian lattices, nodes are usually sequentially ordered by the lattice directions. For scattered nodes, the ordering is in principle arbitrary. However, both for achieving fast memory access (with node sets that do not fit into high-speed cache memory) and for optimal convergence rate with certain iterative linear solvers, the node ordering needs to be optimized. Reorderings based on reverse Cuthill–McGee or ‘locality sensitive hashing’ can be highly beneficial (Bollig, Flyer and Erlebacher 2012).

5.3. Time stabilization: hyperviscosity

For a purely convective PDE, there should not be any solution modes that feature long-term growth or decay. In the case of linear spatial operators, this can be studied via eigenvalue analysis. If the *method of lines* (MOL) discretization for $\partial u/\partial t = Lx$ takes the form $\mathrm{d}\mathbf{u}/\mathrm{d}t = D\,\mathbf{u}$, the eigenvalues of the 'differentiation matrix (DM)' D should be purely imaginary. RBF-FD approximations for L introduce a low level of 'jitter' on the eigenvalues, typically scattering them small distances to each side of the imaginary axis, with the physically relevant ones typically scattered the least. Whereas a small distance to the left of the axis is generally harmless (causing spurious modes to decay slowly), small scatter to the right causes exponential growth in time. What is needed is an approach that leaves the physically relevant (smooth) eigenvalues/modes intact, but 'nudges' spurious oscillatory ones from the right half-plane over into the left one. This can be achieved by hyper-viscosity (Fornberg and Lehto 2011), adapted from turbulence simulations. As an additional benefit, this permits the use of larger (and therefore more accurate) RBF-FD stencils. Without this enhancement, stencils in two dimensions can rarely exceed around $n = 8 - 12$ nodes, whereas with the enhancement, n-values up to around 100 were instrumental in obtaining the high accuracies reported in Flyer *et al.* (2012) and Fornberg and Lehto (2011). There are at present two main hyperviscosity approaches, best-suited for global RBFs and for RBF-FD approximations, respectively.

5.3.1. The A^{-1} method

This approach applies to RBF types with the A-matrix positive definite (*e.g.*, GA, IQ and IMQ but not MQ or PHS). As noted in Section 3.1, the A-matrix eigenvalues will decrease very rapidly to zero if ε is small. The corresponding eigenvectors at the same time become increasingly oscillatory. The matrix A^{-1} will have the same eigenvectors, but its eigenvalues are the inverses for those of A, that is, they will start out $O(1)$ and then rapidly become very large (and again all positive). Hence, adding a term $-\gamma\, A^{-1}\mathbf{u}$ with a very small constant $\gamma > 0$ to the right-hand side of a MOL discretization of a convective PDE $(\mathrm{d}/\mathrm{d}t)u = Lu$ will leave all the physically relevant (reasonably smooth) modes essentially intact, but will rapidly damp out all highly oscillatory (spurious noise) modes.

5.3.2. Powers of the Laplacian

The concept is again to leave smooth modes intact, but to quickly damp out rapidly oscillating high ones. Adding a small multiple of the Laplacian operator Δ to the PDE's right-hand side would damp high modes, but also interfere with low ones (which represent physical information). The analysis and test results in Flyer *et al.* (2012) and Fornberg and Lehto (2011) show that using relatively high powers of Δ achieves what is needed. These references discuss implementation issues, for example, guidelines for powers and multiplying factors to use, and convenient formulas for GA-type RBFs.

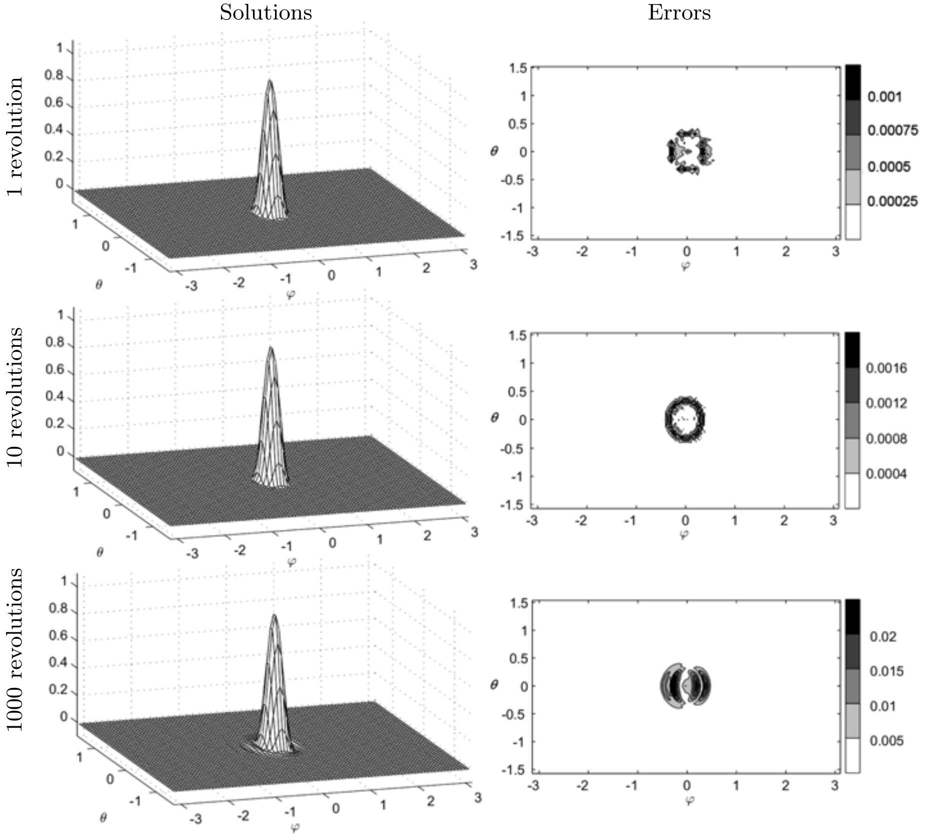


Figure 5.1. Numerical solution and magnitude of the errors for the solid-body rotation test case, using the stabilized RBF-FD approach. The displays are over the (φ, θ)-plane, with $\varphi \in [-\pi, \pi], \theta \in [-\pi/2, \pi/2]$.

A standard test case for studying how well a solution is advected intact, that is, without trailing wavetrains or diffusion, is known as solid-body rotation (Williamson *et al.* 1992). An initial condition, such as a C^1 cosine bell, is advected around the unit sphere at an angle α tilted relative to the polar axis. The governing equation in spherical coordinates is given by

$$\frac{\partial h}{\partial t} + (\cos\alpha - \tan\theta \sin\varphi \sin\alpha)\frac{\partial h}{\partial \varphi} + \cos\varphi \sin\alpha \frac{\partial h}{\partial \theta} = 0. \tag{5.3}$$

Using $N = 25\,600$ MD nodes, a stencil size $n = 74$, GA RBFs with $\varepsilon = 8$, and Δ^8-type hyperviscosity, the long-term evolution is illustrated in Figure 5.1. In spite of the very long integration time (1 000 revolutions around the sphere), there are no visible hints of instabilities or even of loss in peak height (here less than 1%). The main errors remain right at the base of the cosine bell, where there is a jump in the second derivative.

5.4. Compact (implicit) approximations to elliptic PDEs

Since a derivative is a 'local' property of a function, there is something intuitively contradictory about enhancing the order of an FD approximation by invoking data located increasingly far away. When the task is to solve a PDE (rather than just to approximate an operator), *compact approximations* offer a different opportunity for improving the order of accuracy. For finite differences, the concept has a long history (Collatz 1960, Fox 1947) with several more recent enhancements available, such as to nonlinear PDEs in two and three dimensions (Gupta 1991, Lele 1992, Li, Tang and Fornberg 1995, Zhai, Feng and He 2013).

Before considering compact approximations in scattered-node RBF-FD cases, we illustrate the basic idea in the case of approximating

$$\triangle u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$

on a two-dimensional lattice, with spacing h in each direction. The most obvious FD approximation can be written as

$$\begin{bmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{bmatrix} u/h^2 = \triangle u + O(h^2). \tag{5.4}$$

Using only a 3×3 stencil size, it is impossible to find weights that improve the accuracy above second order. Extending the stencil to five nodes in both directions permits fourth-order accuracy, but causes problems when solving the PDE $\triangle u = f$.

(i) The centre weight becomes smaller in magnitude than the sum of magnitudes of the remaining weights, that is, *diagonal dominance* is lost. This damages the convergence rate of many iterative schemes, and it also opens up the possibility of system singularities.

(ii) Wider stencils need more boundary information than is readily available.

Taylor expansions will however reveal that, if the task is not to approximate $\triangle u$ but to solve $\triangle u = f$, then

$$\begin{bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{bmatrix} u/(6h^2) = \begin{bmatrix} & 1 & \\ 1 & 8 & 1 \\ & 1 & \end{bmatrix} f/12 + O(h^4), \tag{5.5}$$

and for the special case of solving $\triangle u = 0$,

$$\begin{bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{bmatrix} u/(6h^2) = 0 + O(h^6).$$

The latter approximations suffer neither of the two problems noted above, but nevertheless achieve significantly improved levels of accuracy.

Equation (5.5) can be recast as a compact approximation to $\triangle u$ as

$$[1]\;\Delta u = \begin{bmatrix} \frac{1}{4} & 1 & \frac{1}{4} \\ 1 & -5 & 1 \\ \frac{1}{4} & 1 & \frac{1}{4} \end{bmatrix} u/h^2 + \begin{bmatrix} & -\frac{1}{8} & \\ -\frac{1}{8} & & -\frac{1}{8} \\ & -\frac{1}{8} & \end{bmatrix} \Delta u + O(h^4). \tag{5.6}$$

RBF-FD counterparts to (5.6) for scattered nodes can readily be generated, as described in Fornberg and Flyer (2015*b*) and Wright and Fornberg (2006). The latter reference provides several test examples, showing that the advantages noted above for compact formulas carry over from lattice-based FD cases to scattered-node RBF-FD cases.

6. Three examples of solving PDEs with RBF-FD

6.1. The shallow water equations on a sphere

The equations in a three-dimensional Cartesian coordinate system for a rotating fluid are

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u}\cdot\nabla)\mathbf{u} - f(\mathbf{x}\times\mathbf{u}) - g\nabla h, \tag{6.1}$$

$$\frac{\partial h}{\partial t} = -\nabla\cdot(h\mathbf{u}), \tag{6.2}$$

where f is the Coriolis force, $\nabla = \partial_x\hat{\mathbf{i}} + \partial_y\hat{\mathbf{j}} + \partial_z\hat{\mathbf{k}}$, $\mathbf{u} = u\hat{\mathbf{i}} + v\hat{\mathbf{j}} + w\hat{\mathbf{k}}$ is the velocity vector, h is the geopotential height and $\mathbf{x} = \{x, y, z\}^T$ represents the position vector. Working in Cartesian coordinates requires a projection operator that confines the motion to the surface of the sphere, that is,

$$\nabla \to \mathbf{P}\nabla = [\mathbf{p}_x\cdot\nabla, \mathbf{p}_y\cdot\nabla, \mathbf{p}_z\cdot\nabla],$$

where

$$\mathbf{P}\nabla = \begin{bmatrix} (1-x^2) & -xy & -xz \\ -xy & (1-y^2) & -yz \\ -xz & -yz & (1-z^2) \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{bmatrix} = \begin{bmatrix} \mathbf{p}_x\cdot\nabla \\ \mathbf{p}_y\cdot\nabla \\ \mathbf{p}_z\cdot\nabla \end{bmatrix}.$$

Notice that each component of the projected gradient for a given direction is a linear combination of the other three. In addition, the right-hand side of (6.1) needs to be projected, with the modified differential operators, in the corresponding $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$, and $\hat{\mathbf{k}}$ directions. For example, in the case of the u momentum equation (corresponding to the velocity in the x direction), this results in

$$\frac{\partial u}{\partial t} = -\mathbf{p}_x \tag{6.3}$$

$$\underbrace{\cdot \begin{bmatrix} u(\mathbf{p}_x\cdot\nabla)u + v(\mathbf{p}_y\cdot\nabla)u + w(\mathbf{p}_z\cdot\nabla)u \\ u(\mathbf{p}_x\cdot\nabla)v + v(\mathbf{p}_y\cdot\nabla)v + w(\mathbf{p}_z\cdot\nabla)v \\ u(\mathbf{p}_x\cdot\nabla)w + v(\mathbf{p}_y\cdot\nabla)w + w(\mathbf{p}_z\cdot\nabla)w \end{bmatrix} + f\begin{bmatrix} yw - zv \\ zu - xw \\ xv - yu \end{bmatrix} + g\begin{bmatrix} (\mathbf{p}_x\cdot\nabla) \\ (\mathbf{p}_y\cdot\nabla) \\ (\mathbf{p}_z\cdot\nabla) \end{bmatrix} h\ \Bigg]}_{\text{right-hand side.}}$$

Notice that the only differential operator L that needs to be discretized is $\mathbf{P}\nabla$, and it can be calculated as in Section 5.1. It was noticed that the addition of any polynomials beyond a constant did not affect the results. For further details see Flyer *et al.* (2012), which also provides a greatly simplified way to calculate the projected gradient for the sphere.

By adding a forcing term h_{mtn} to the right-hand side of the geopotential height h equation in (6.2), flow over a mountain can be simulated (Takacs 1988, Williamson *et al.* 1992). Two mountain profiles, one where h_{mtn} is a C^1 cone and the other a C^∞ mountain, are considered to illustrate the sensitivity of high-order methods to Gibbs phenomena. This is important because topographical features are rarely even C^1. Figure 6.1 shows the results for a 15-day run using Runge–Kutta fourth-order (RK4) time-stepping, with the reference solution given by a discontinuous Galerkin (DG) shallow water model (Blaise and St-Cyr 2012), where each element contains 12×12 Legendre quadrature nodes to represent the solution, which results in a total of 884 736 degrees of freedom and an average resolution around 26 km. The results are given in Figure 6.1. The key differences between the two columns of panels is that (i) even though the C^∞ Gaussian mountain is slightly steeper than the C^0 mountain, there are no high-frequency waves emanating throughout the domain, and (ii) after $n = 31$, stencil size has no bearing on convergence or accuracy with the C^1 cone forcing. This latter fact is that with non-smooth forcing, the only way to increase accuracy is to increase resolution about the base of the mountain and not the order of the method.

We consider three reference solutions: (i) the DG reference solution mentioned above, (ii) a spherical harmonic solution from the DWD (Deutscher Wetterdienst, German National Weather Service) that has a spectral

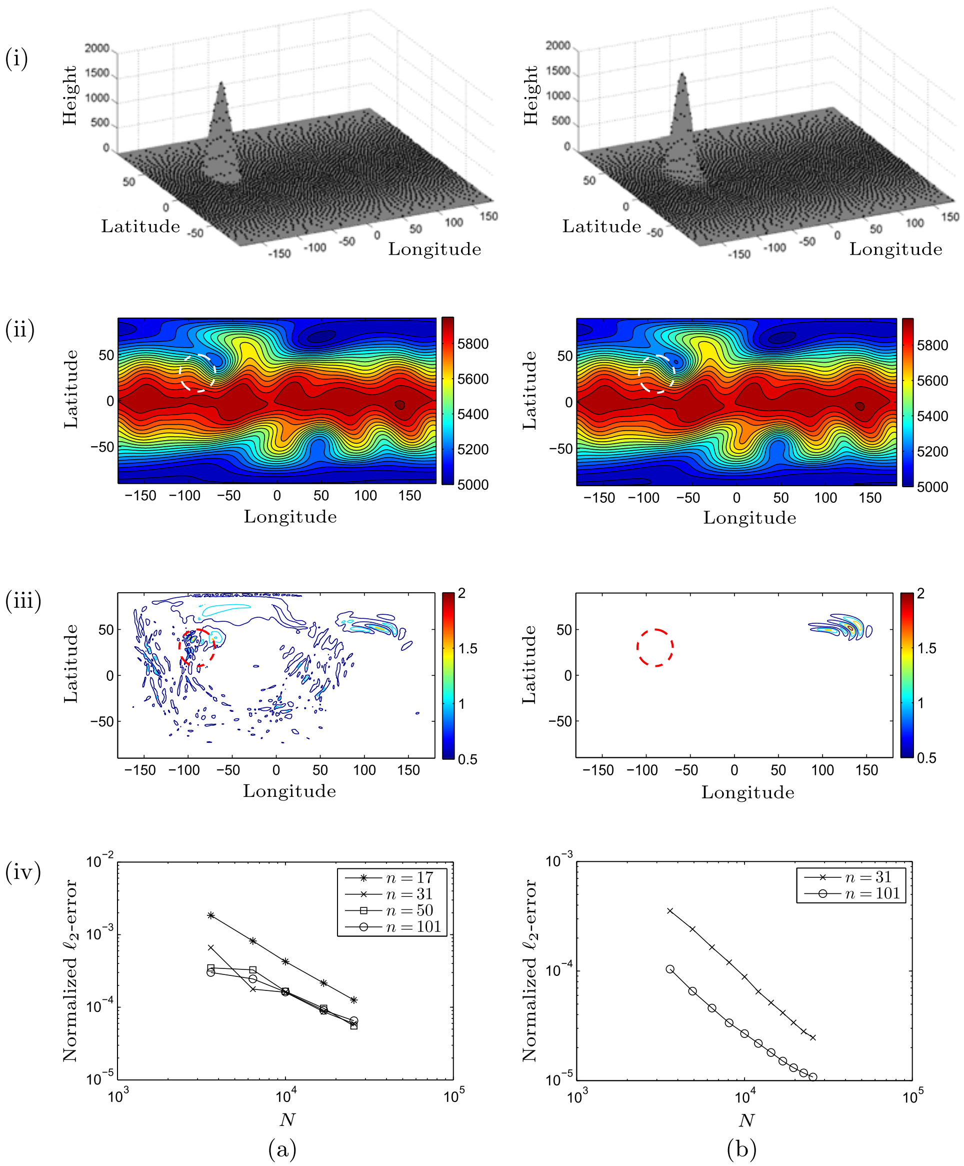
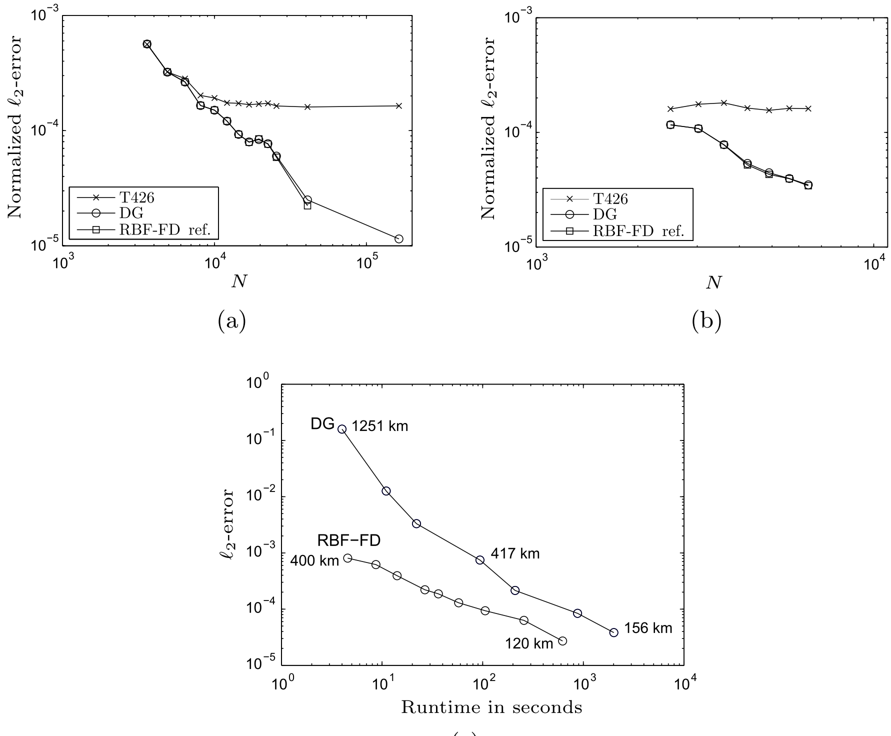


Figure 6.1. (a) Cone mountain results. (i) Profile of mountain. (ii) Plotted in metres, RBF-FD solution for h at day 15, $N = 25\,600$ and $n = 31$ with contour intervals at 50 m. (iii) Magnitude of the error between the RBF-FD solution and DG reference solution in metres. The contour interval is 0.5 m, with white denoting errors less than 0.1 m. (iv) ℓ_2-error as a function of the resolution N for varying stencil sizes. (b) The same as (a) but for Gaussian mountain forcing. The dashed circle in all plots is the base of the mountain.



(a)

(b)

(c)

Figure 6.2. (a,b) Convergence plots in the ℓ_2-norm with regard to the geopotential height fields h for the RBF-FD (a) and global RBF (b), against the three reference solutions. Note that the RBF-FD and DG solutions agree perfectly. (c) The error as a function of computer runtime for the RBF-FD and DG methods.

truncation of T426, that is, it uses 182 329 spherical harmonic bases, and (iii) an RBF-FD based on $N = 163\,824$ icosahedral-type nodes on the sphere, representing a 60 km resolution, and a stencil size of $n = 31$. Figure 6.2(a) shows that the ℓ_2-errors for the RBF-FD method, whether using the RBF-FD reference solution or a DG one, are almost identical. This same trend is also seen in Figure 6.2(b) with global RBFs. In contrast, the error from the SH reference solution is an order of magnitude larger. Given that DG, RBF-FD, and global RBFs are vastly different numerical methods, this strongly indicates that the SH spectral model is providing a less accurate solution, while DG and RBF are in line with one another.

The next consideration is time benchmarking of RBF-FD against DG. The present benchmarking was done on a MacBook Pro laptop with an Intel i7 2.2 GHz quad-core processor, using only a single core, and 8 GB of memory. The RBF-FD code was written in MATLAB and the DG code in C++. The RBF-FD reference solution of $N = 163\,842$ and $n = 31$ (*i.e.*, 60 km resolution) was used to calculate the ℓ_2-error versus runtime (*i.e.*, wall-clock time) for both methods as shown in Figure 6.2(c). The RBF-FD method was computationally faster than the DG method, from about an order and a half of magnitude for coarser resolutions to four times faster for the finest resolutions.

6.2. The compressible Navier–Stokes equations on a limited area domain

The compressible Navier–Stokes equations in a two-dimensional Cartesian coordinate system, $\{x, z\}$, for stratified fluid flow (important in atmospheric processes) are as follows:

$$\begin{aligned} &\text{momentum} \quad && \frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u}\cdot\nabla)\mathbf{u} - c_p\theta\nabla P - g\mathbf{k} + \mu\Delta\mathbf{u}, \\ &\text{energy} && \frac{\partial \theta}{\partial t} = -(\mathbf{u}\cdot\nabla)\theta + \mu\Delta\theta, \\ &\text{mass} && \frac{\partial P}{\partial t} = -(\mathbf{u}\cdot\nabla)P - \frac{R}{c_v}(\nabla\cdot\mathbf{u})P, \end{aligned} \tag{6.4}$$

where $P = (p/P_0)^{R/c_p}$ is the non-dimensional Exner pressure ($P_0 = 1 \times 10^5$ Pa), and $\theta = T/P$ is the potential temperature. The constants $c_p = 1004$ and $c_v = 717$ are the specific heat at constant pressure and constant volume, respectively, $R = c_p - c_v = 287$, and μ, the dynamic viscosity. These equations are often used for testing novel numerical methods in atmospheric modelling, as will be done here.

A commonly used test case is known as the Straka density current (Straka *et al.* 1993). A bubble of cold air falls to the ground and develops three smooth and distinct rotors due to shear instability, as it spreads sideways. The computational domain is $[-25.6, 25.6]$ km in x with periodic boundary conditions, and $[0, 6.4]$ km in z with no-flux and free-slip boundary conditions on the velocity and Neumann on the temperature and pressure. The dynamic viscosity is $\mu = 75$ m^2 s^{-1}. PHS RBFs, r^7, together with polynomials up to third degree are used to approximate all spatial derivatives locally by the RBF-FD approach with a stencil size of $n = 37$. The remaining system of first-order ODEs is time-stepped with RK4. Figure 6.3 shows the behaviour of the numerical solution in time from $t = 0$ s until the final time $t = 900$ s.

The RBF-FD approach makes it particularly easy to test how different node distributions (all with the same total number of nodes) influence the extent to which the physics is captured. In Figure 6.4, three different node layouts are examined: Cartesian, hexagonal, and quasi-uniformly scattered.

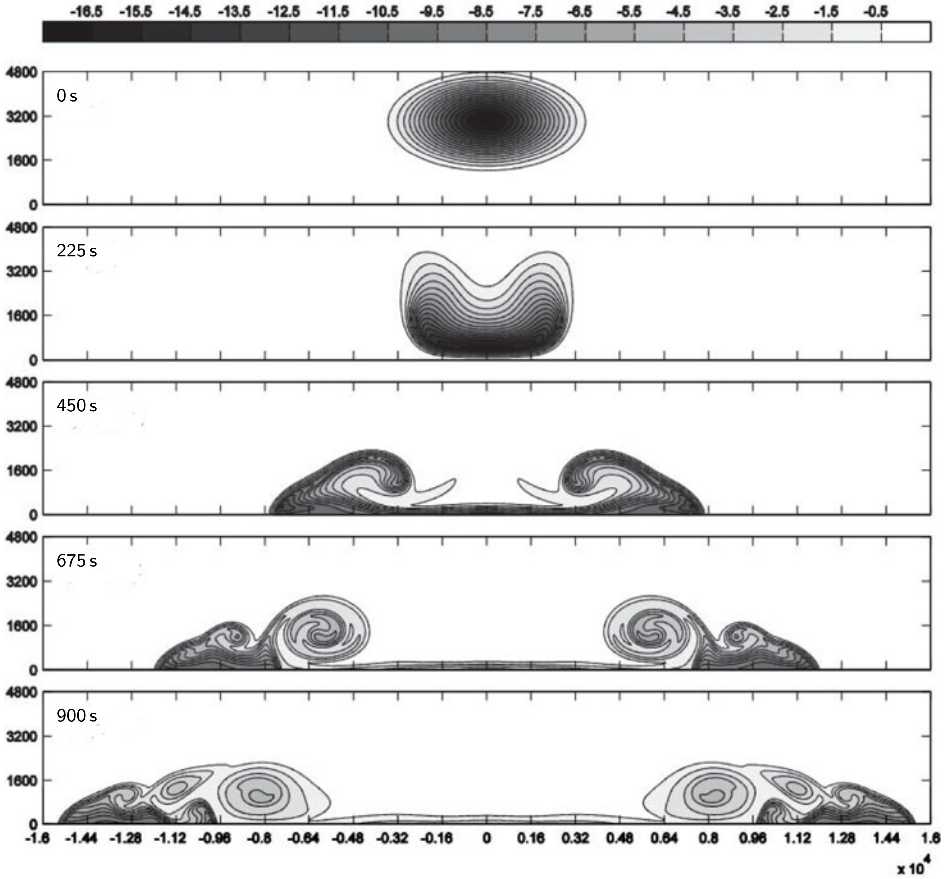


Figure 6.3. The time evolution of the potential temperature θ using a hexagonal node layout at 100 m resolution.

Convergence under refinement leads in all cases to the same solution, as seen in the highest-resolution displays (bottom row). However, in numerical weather prediction, the ability to work at such fine resolutions as 100 m is a luxury rather than a reality. The fact that observational data that initialize models are observed on the order of kilometres, makes the degree to which the physics is captured at coarser resolutions more important. Three key features to be noticed are: (i) formation of the rotors, (ii) how much cold air they have entrenched (larger negative values of θ, black) and (iii) where the front location is. In the coarsest case shown, using only 720 nodes in the domain (about 700 m resolution), the hexagonal and scattered-node calculations give more clear evidence of the first rotor being formed. At

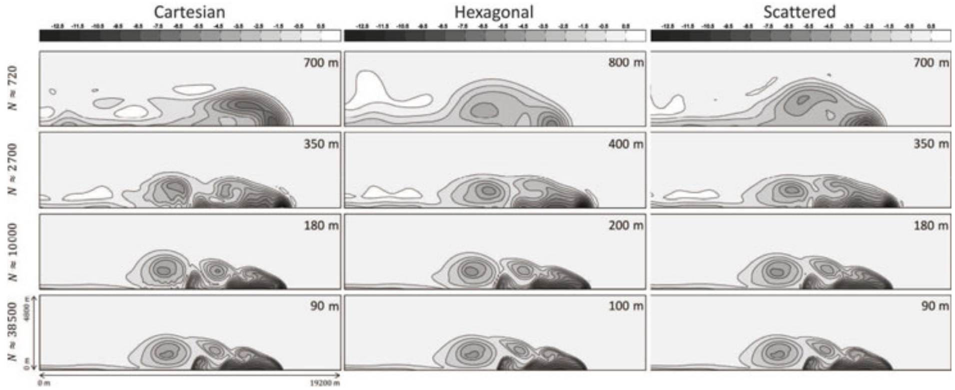
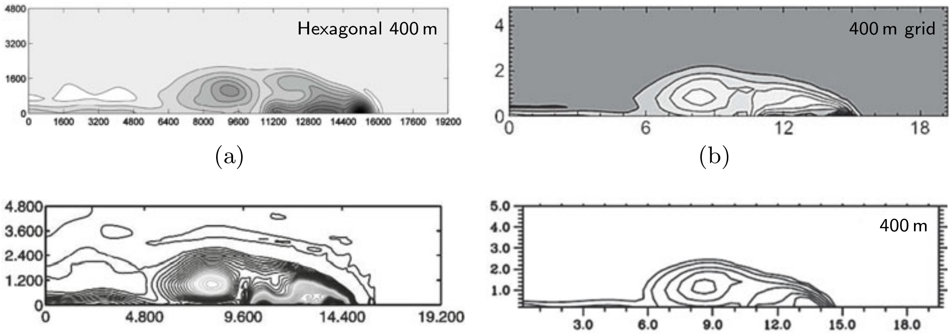


Figure 6.4. The potential temperature θ for the Straka density current test case at the final time $t = 900$ s, shown as a function of the total number of nodes when using the RBF-FD method on different node sets. For plot clarity, only half of the solution is displayed.

the next-higher resolution (2 700 nodes in the entire domain, about 350 m resolution), they provide a better picture of the formation of subsequent rotors, as well as more accurate entrenchment of cold air (black) near the front, looking more similar to the high-resolution 90 m case. Cartesian nodes furthermore give solutions more prone to Gibbs phenomenon oscillations (overshoots in white of 2.4 K as opposed to 1 K).

The calculations for Figure 6.4 all used RBF-FD stencils of size $n = 37$, generated from $\phi(r) = r^7$, supported with polynomials up through degree 3. The differences between the columns of subplots reflect only the intrinsic resolution capabilities of the different node layouts. The traditional Cartesian choice is the least effective one. If using a fixed node separation, a hexagonal layout can 'pack' more nodes into a fixed region than a Cartesian one. Conversely, in the present case with fixed node numbers, their separation becomes somewhat larger. Even so, at every resolution level, the hexagonal choice gives better accuracy than the Cartesian one. The big advantage of generalizing further, from hexagonal to quasi-uniformly scattered nodes, is that it then becomes trivial to implement spatially variable node densities, that is, 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.

To place Figure 6.4 in context with the results of other numerical methods, a comparison is done with DG, spectral element (SE), finite volume



(a) (b)

(c) (d)

Figure 6.5. Comparison at 400 m between four different numerical methods: (a) RBF-FD (Barnett, Flyer and Wicker 2015), (b) fifth-order upwind advection, (c) discontinuous Galerkin and spectral element, eighth-order, (d) finite volume solver. Contour intervals: (a) 1 K, (b) 1 K, (c) 0.25 K, (d) 1 K.

Figures 6.5(b), 6.5(c) and 6.5(d) are reproduced with the kind permission of Elsevier. All three plots are from the *Journal of Computational Physics*: (b) is from Figure 4 of Skamarock and Klemp (2008, p. 3475); (c) is from Figure 7 of Giraldo and Restelli (2008, p. 3869); (d) is from Figure 5 of Norman, Nair and Semazzi (2011, p. 1578).

(FV), and upwinding schemes in Figure 6.5. As can be seen, when no filtering is used in the RBF-FD method, there is a trade-off between capturing features at low resolutions and preserving monotonicity. Only the FV and upwind schemes do not exhibit Gibbs' oscillations and have solutions with monotonic properties. However, the price to be paid is that the solution is smoothed out both with regard to rotor formation and the amount of cold air that has been entrenched. The DG and SE solutions have more structure, but the beginning formation of the second rotor is still not seen as well as in the RBF-FD model.

Without any explicit viscosity, the solution enters the turbulent regime with the dynamics now modelled by the Euler equations. In such regimes, there is no convergence to any solution as energy cascades to smaller and smaller scales, eventually entering the subgrid-scale domain. Nevertheless, it is interesting to observe whether the model remains stable in this regime. Figure 6.6 shows the solution at 50 m and 25 m resolutions on a hexagonal layout (optimal in two dimensions and easily implemented with RBFs). The fact that now there is no explicit viscosity, that is, $\mu = 0$ in (6.4), does not affect the time stability, and the time step did not have to be altered between the two cases. Stability is governed solely by the fact that the time step could not exceed the speed of sound in air.

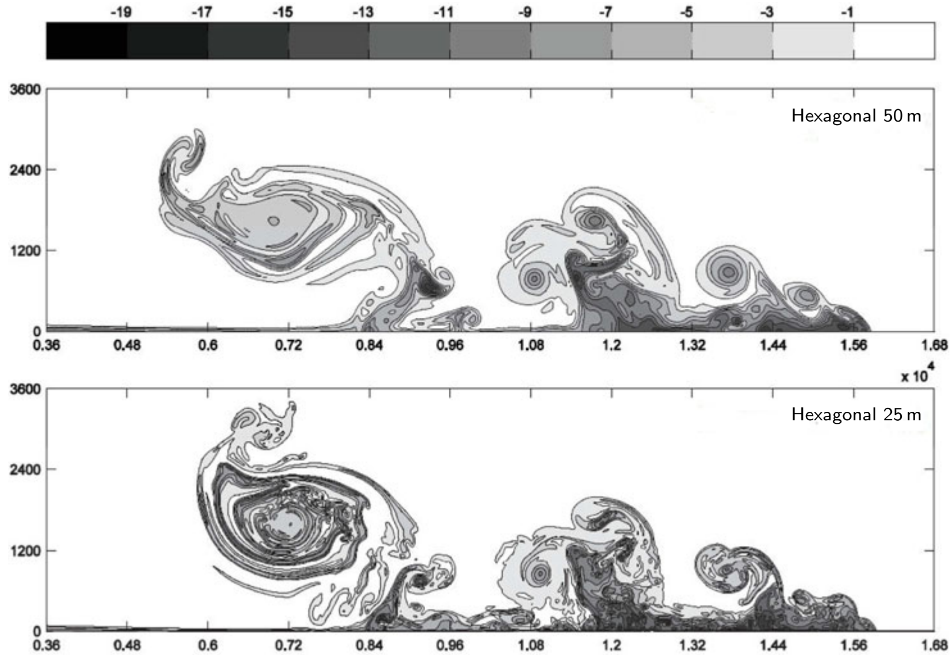


Figure 6.6. The potential temperature θ for the Straka density current test case with the dynamic viscosity $\mu = 0$ on a hexagonal node layout of 50 m and 25 m.

6.3. Forward seismic modelling

6.3.1. Background

Seismic exploration is the primary tool used for finding and then mapping out hydrocarbon deposits. In forward modelling, subsurface structures are assumed to be known, and the task is to simulate elastic wave propagation through the medium. Inversion programs then update subsurface assumptions to reconcile the model response with actual measurements. There are typically hundreds of irregularly curved interfaces present, often interrupted by fracture lines with associated translations between the strata on the two sides. During the history of the earth, the vast majority of all hydrocarbons (such as natural gas and oil), being lighter than water, have migrated up to the surface and then biodegraded. What is left are mostly small pockets where hard layers have somehow formed traps for this upward migration, due to their curvature or the presence of corners resulting from fractures. With drilling being far more expensive (and environmentally damaging) than seismic exploration, the latter is constantly pushed to its limits, leading to some of the largest computational tasks in any field.

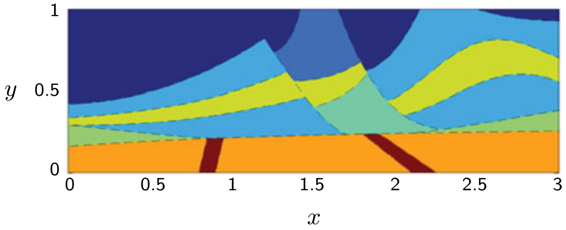


Figure 6.7. Subsurface acoustic velocities in a ‘micro-Marmousi’ test case.

Figure 6.7 shows a extremely simplified model for the Marmousi test case, itself a highly simplified two-dimensional vertical slice off the coast of Madagascar (as shown in Figure 2 of Martin, Wiley and Marfurt 2006). The governing elastic wave equations in two dimensions are

$$\begin{aligned} \rho u_t &= f_x + g_y, \\ \rho v_t &= g_x + h_y, \\ f_t &= (\lambda + 2\mu)\, u_x + \lambda v_y, \\ g_t &= \mu\,(u_x + v_y), \\ h_t &= (\lambda + 2\mu)\, v_y + \lambda u_x. \end{aligned} \tag{6.5}$$

The dependent variables are u, v (horizontal and vertical velocities) and f, g, h (components of the symmetric stress tensor), and the material is specified by ρ (density) and λ, μ (Lamé parameters for compression and shear). Away from interfaces, these equations support two types of waves: P-waves (pressure or primary) with speed $c_p = \sqrt{(\lambda + 2\mu)/\rho}$, and S-waves (shear or secondary) with speed $c_s = \sqrt{\lambda/\rho}$. Each incoming wave to an interface generally results in four main outgoing waves: reflected and transmitted P-waves and S-waves (and possibly also waves following interfaces). With typically hundreds of interfaces, wave patterns become extremely complicated. Simulated return signals at the surface need to accurately represent wave propagation for long distances through regions with smoothly varying material properties, as well as reflection-transmissions (with respect to amplitudes, phase angles, and directions).

In the smoothly varying regions, the dominant error source is numerical dispersion. The only practical remedy for this is to use high-order approximations (Fornberg 1987). The industry standard moved from second to fourth order in the 1980s, and FD approximations of extremely high (around 20th) order are now in common use. It has proved much more difficult to achieve accurate interface treatments (Lombard and Piraux 2004, Lombard, Piraux, Gelis and Virieux 2008, Symes and Vdovina 2009). While closed-form expressions are available in simplified cases (such as straight interfaces

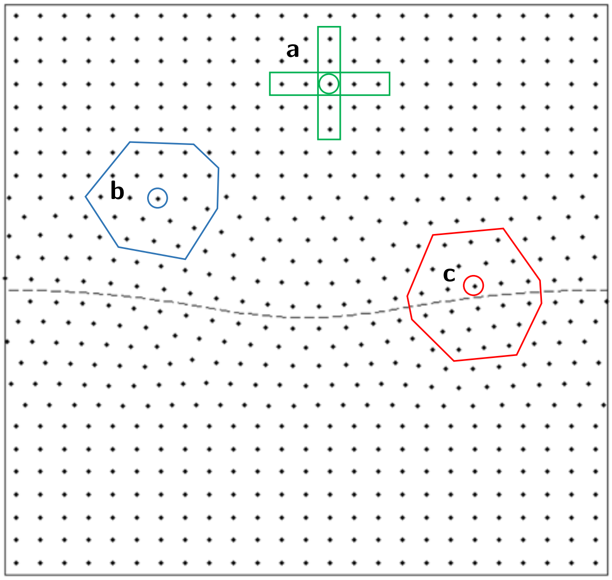


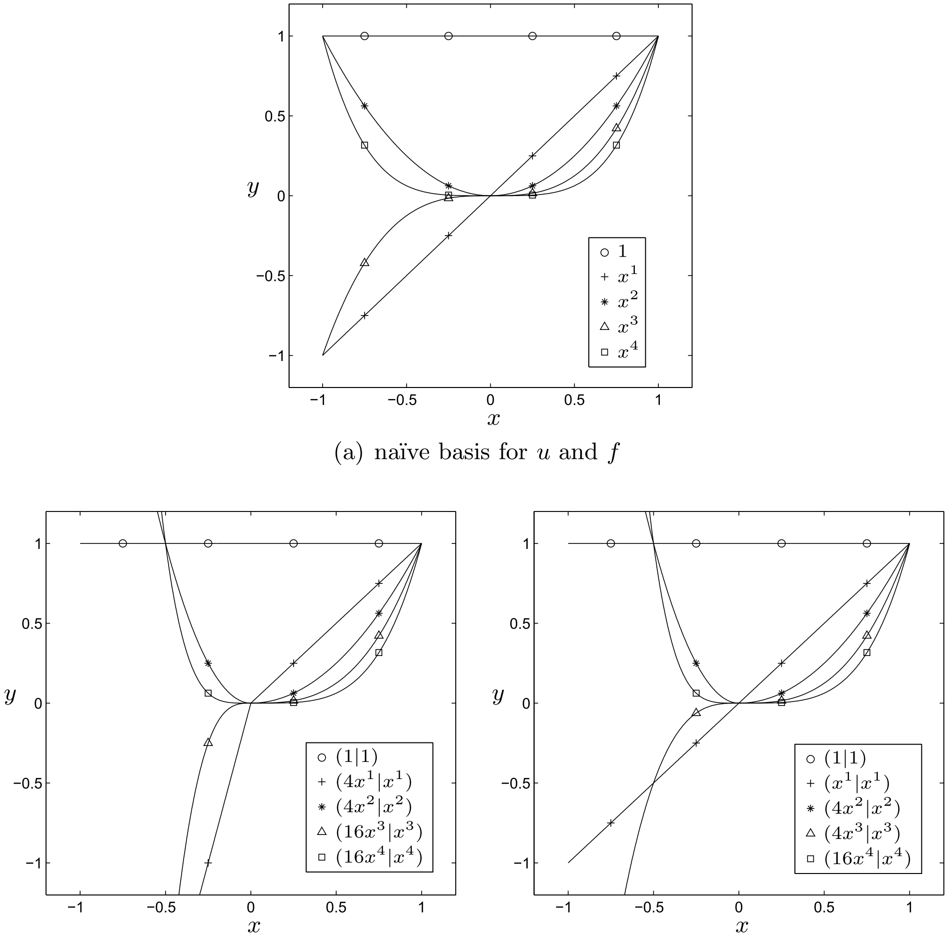
Figure 6.8. The three stencil types (a), (b), (c) used in a hybrid approach combining FD with RBF-FD/AC.

between constant media), incorporating these in full production codes has so far not been cost-effective. Surprisingly, it has become standard procedure in the industry to omit special treatment (beyond the mild smoothing of interfaces), accepting, typically, first-order convergence for reflected waves. The present RBF-FD/AC method (with AC standing for analytic correction) achieves third-order accuracy both in smooth regions and across smoothly curved interfaces, making it very competitive.

6.3.2. RBF-FD/AC approach

Figure 6.8 illustrates a typical node layout and the different stencil types used in an even more simplified test case, with just one curved interface in two dimensions. The nodes are distributed to straddle the interface but then smoothly transition to become lattice-based a short distance away from it. Three stencil types are used: (a) regular FD when the whole stencil is lattice-based, (b) standard RBF-FD where some nodes are irregularly placed, while away from the interface, and (c) RBF-FD/AC when an interface intersects a stencil.

Although the RBF-FD/AC discretization has been successfully tested in one, two and three dimensions, we limit ourselves here to describing it in one dimension (for a two-dimensional description: see Martin, Fornberg and



(a) naïve basis for u and f

(b) AC basis for u

(c) AC basis for f

Figure 6.9. The naïve supporting monomials up through degree 4 compared to the interface-specific ones in the special case of $c_L = 1, c_R = 2, \rho_L = \rho_R = 1$.

St-Cyr 2015). The governing equation reduces in one dimension to

$$\frac{\partial}{\partial t}\begin{bmatrix} u \\ f \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{\rho}\frac{\partial}{\partial x} \\ \rho c^2 \frac{\partial}{\partial x} & 0 \end{bmatrix} \begin{bmatrix} u \\ f \end{bmatrix}. \tag{6.6}$$

While ρ and c typically both jump at an interface, continuity of motion and traction requires u and f to be continuous (in the two-dimensional case, there will similarly be four continuity relations linking the five variables in (6.5)). Denoting left and right sides of the interface by subscripts L and R

respectively, it will then hold that

$$\begin{bmatrix} u_L \\ f_L \end{bmatrix} - \begin{bmatrix} u_R \\ f_R \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

which implies

$$\frac{\partial^k}{\partial t^k}\left\{\begin{bmatrix} u_L \\ f_L \end{bmatrix} - \begin{bmatrix} u_R \\ f_R \end{bmatrix}\right\} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad k = 0, 1, 2, \ldots .$$

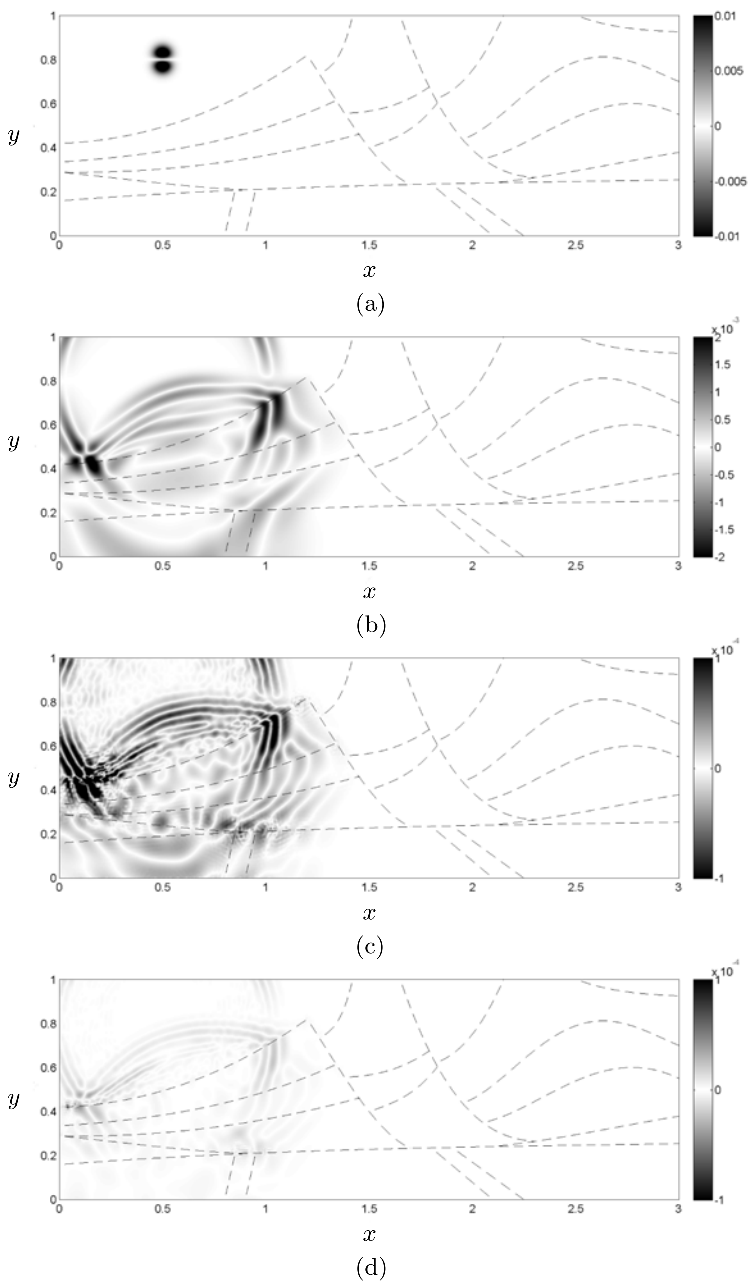
With use of (6.6), these time identities will translate to relations between spatial derivatives for u and f on the two sides. The idea is to embed these relations in the supplementing polynomials for the RBF-FD approximation (but not in the RBFs themselves: see Yu and Chen 2011, where a similar approach was considered in the context of Maxwell's equations). Figure 6.9 illustrates how one thus arrives at 'interface-aware' supplementary polynomials (with their changes across the interface dependent on the material properties on the two sides).

6.3.3. Two-dimensional test case

In the geometry shown in Figure 6.7, Figure 6.10(a) (overleaf) shows the vertical velocity v associated with an underground explosive source, and Figure 6.10(b) a very accurate calculation of the solution at a certain later time. Figures 6.10(c) and 6.10(d) display the error at this same later time for RBF-FD/AC solutions when using $N = 38\,400$ and $N = 153\,600$ nodes, respectively (using IMQ-type RBFs: stencils of type (b), as shown in Figure 6.8, $n = 19$, polynomials degree 3; stencils of type (c), $n = 38$, polynomials degree 2). Since the colour bars for these latter two cases (Figures 6.10(c) and 6.10(d)) are identical, one can readily note that halving the typical node separation h has reduced the error by more than a factor of ten. For a scheme that is third-order accurate everywhere, the expected error reduction would have been a factor of eight.

7. Conclusions

Ever since RBFs were first introduced for multivariate interpolation, their range of applications has grown tremendously. In some sense, computational experiences with RBFs for PDEs are now well ahead of the more strict analysis of these methods. With so many free parameters associated with irregularly scattered nodes, strict numerical analysis obviously becomes far more difficult than for lattice-based methods, such as FD or PS. This lack of rigorous theory, for example with regard to stability during time-stepping, might have somewhat delayed the broad adoption of RBF-based methods for large applications. However, the number of successful large-scale benchmark comparisons against alternative PDE approaches is now steadily increasing.



(a)

(b)

(c)

(d)

Figure 6.10. Test calculation for the ‘micro-Marmousi’ example using the RBF-FD/AC approach, showing better than a factor of 10 reduction in error when the number of nodes is doubled. (a) Initial condition for v at $t = 0$, (b) solution for v at $t = 0.3$, (c) RBF-FD/AC error, $N = 38\,400$ nodes, (d) RBF-FD/AC error, $N = 153\,600$ nodes.

The present authors hope that this will stimulate new advances on all fronts of the topic of RBFs for PDEs: theoretical, computational, as well as still further extending the range of application areas.

Acknowledgements

This research was supported by NSF grants DMS-0914647, DMS-0934317, OCI-0904599 and by Shell International Exploration and Production, Inc.

REFERENCES

G. A. Barnett, N. Flyer and L. J. Wicker (2015), An RBF-FD polynomial method for nonhydrostatic atmospheric modeling on different node layouts. Submitted.

V. Bayona and M. Kindelan (2013), 'Propagation of premixed laminar flames in 3D narrow open ducts using RBF-generated finite differences', *Combust. Theory Model.* **17**, 789–803.

D. Bercovici, G. Schubert, G. A. Glatzmaier and A. Zebib (1989), 'Three-dimensional thermal convection in a spherical shell', *J. Fluid Mech.* **206**, 75–104.

S. Blaise and A. St-Cyr (2012), 'A dynamic *hp*-adaptive discontinuous Galerkin method for shallow water flows on the sphere with application to a global tsunami simulation', *Mon. Weather Rev.* **140**, 978–996.

S. Bochner (1933), 'Monotone Functionen, Stieltjes Integrale und harmonische Analyse', *Math. Ann.* **108**, 378–410.

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

J. P. Boyd (2000), *Chebyshev and Fourier Spectral Methods*, Dover.

M. D. Buhmann (2000), Radial basis functions. In *Acta Numerica*, Vol. 9, Cambridge University Press, pp. 1–38.

M. D. Buhmann (2003), *Radial Basis Functions: Theory and Implementations*, Vol. 12 of *Cambridge Monographs on Applied and Computational Mathematics*, Cambridge University Press.

W. Chen, Z.-J. Fu and C. S. Chen (2014), *Recent Advances in Radial Basis Function Collocation Methods*, Springer Briefs in Applied Sciences and Technology, Springer.

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

L. Collatz (1960), *The Numerical Treatment of Differential Equations*, Springer.

P. C. J. Curtis (1959), '*n*-parameter families and best approximation', *Pacific J. Math.* **93**, 1013–1027.

T. A. Driscoll and B. Fornberg (2002), 'Interpolation in the limit of increasingly flat radial basis functions', *Comput. Math. Appl.* **43**, 413–422.

J. Duchon (1977), Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In *Constructive Theory of Functions of Several Variables*, Vol. 571 of *Lecture Notes in Mathematics* (W. Schempp and K. Zeller, eds), Springer, pp. 85–100.

G. E. Fasshauer (1997), Solving partial differential equations by collocation with radial basis functions. In *Surface Fitting and Multiresolution Method*, Vol. 2, *Proc. 3rd International Conference on Curves and Surfaces* (A. Le Méhauté, C. Rabut and L. L. Schumaker, eds), Vanderbilt University Press, pp. 131–138.

G. E. Fasshauer (2007), *Meshfree Approximation Methods with MATLAB*, Vol. 6, *Interdisciplinary Mathematical Sciences*, World Scientific.

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

N. Flyer and E. Lehto (2010), 'Rotational transport on a sphere: Local node refinement with radial basis functions', *J. Comput. Phys.* **229**, 1954–1969.

N. Flyer and G. B. Wright (2007), 'Transport schemes on a sphere using radial basis functions', *J. Comput. Phys.* **226**, 1059–1084.

N. Flyer and G. B. Wright (2009), 'A radial basis function method for the shallow water equations on a sphere', *Proc. Roy. Soc.* A **465**, 1949–1976.

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

B. Fornberg (1987), 'The pseudospectral method: Comparisons with finite differences for the elastic wave equation', *Geophysics* **52**, 483–501.

B. Fornberg (1996), *A Practical Guide to Pseudospectral Methods*, Cambridge University Press.

B. Fornberg (1998), 'Calculations of weights in finite difference formulas', *SIAM Rev.* **40**, 685–691.

B. Fornberg and N. Flyer (2015*a*), 'Fast generation of 2-D node distributions for mesh-free PDE discretizations', *Comput. Math. Appl.* doi:10.1016/j.camwa.2015.01.009

B. Fornberg and N. Flyer (2015*b*), *A Primer on Radial Basis Functions with Applications to the Geosciences*, SIAM.

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

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

B. Fornberg and C. Piret (2008), 'On choosing a radial basis function and a shape parameter when solving a convective PDE on a sphere', *J. Comput. Phys.* **227**, 2758–2780.

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

B. Fornberg and J. Zuev (2007), 'The Runge phenomenon and spatially variable shape parameters in RBF interpolation', *Comput. Math. Appl.* **54**, 379–398.

B. Fornberg, T. A. Driscoll, G. Wright and R. Charles (2002), 'Observations on the behavior of radial basis functions near boundaries', *Comput. Math. Appl.* **43**, 473–490.

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

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

B. Fornberg, E. Lehto and C. Powell (2013), 'Stable calculation of Gaussian-based RBF-FD stencils', *Comput. Math. Appl.* **65**, 627–637.

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

L. Fox (1947), 'Some improvements in the use of relaxation methods for the solution of ordinary and partial differential equations', *Proc. Roy. Soc.* A **190**, 31–59.

E. J. Fuselier and G. B. Wright (2013), 'A high-order kernel method for diffusion and reaction–diffusion equations on surfaces', *J. Sci. Comput.* **56**, 535–565.

F. X. Giraldo and M. Restelli (2008), '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**, 3849–3877.

M. M. Gupta (1991), 'High accuracy solutions of incompressible Navier–Stokes equations', *J. Comput. Phys.* **93**, 343–359.

H. Harder and U. Hansen (2005), 'A finite-volume solution method for thermal convection and dynamo problems in spherical shells', *Geophys. J. Int.* **161**, 522–532.

R. L. Hardy (1971), 'Multiquadric equations of topography and other irregular surfaces', *J. Geophys. Res.* **76**, 1905–1915.

Y. C. Hon and R. Schaback (2001), 'On unsymmetric collocation by radial basis functions', *Appl. Math. Comput.* **119**, 177–186.

A. Iske (2004), *Multiresolution Methods in Scattered Data Modelling*, Vol. 37 of *Lecture Notes in Computational Science and Engineering*, Springer.

M. C. Kameyama, A. Kageyama and T. Sato (2008), 'Multigrid-based simulation code for mantle convection in spherical shell using Yin–Yang grid', *Phys. Earth Planet. Interiors* **171**, 19–32.

E. J. Kansa (1990*a*), 'Multiquadrics: A scattered data approximation scheme with applications to computational fluid-dynamics, part I: Surface approximations and parital derivative estimates', *Comput. Math. Appl.* **19**, 127–145.

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

B. B. T. Kee, G. R. Liu and C. Lu (2008), 'A least-square radial point collocation method for adaptive analysis in linear elasticity', *Eng. Anal. Bound. Elem.* **32**, 440–460.

M. Kindelan, F. Bernal, P. Gonzalez-Rodriguez and M. Moscoso (2010), 'Application of the RBF meshless method to the solution of the radiative transport equation', *J. Comput. Phys.* **229**, 1897–1908.

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

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

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

M. Li, T. Tang and B. Fornberg (1995), 'A compact fourth-order finite difference scheme for the steady incompressible Navier–Stokes equations', *Internat. J. Numer. Meth. Fluids* **20**, 1137–1151.

B. Lombard and J. Piraux (2004), 'Numerical treatment of two-dimensional interfaces for acoustic and elastic waves', *J. Comput. Phys.* **195**, 90–116.

B. Lombard, J. Piraux, C. Gelis and J. Virieux (2008), 'Free and smooth boundaries in 2-D finite-difference schemes for transient elastic waves', *Geophys. J. Int.* **172**, 252–261.

W. R. Madych and S. A. Nelson (1992), 'Bounds on multivariate polynomials and exponential error estimates for multiquadric interpolation', *J. Approx. Theory* **70**, 94–114.

J. C. Mairhuber (1956), 'On Haar's theorem concerning Chebyshev approximation problems having unique solutions', *Proc. Amer. Math. Soc* **7**, 609–615.

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

G. S. Martin, R. Wiley and K. J. Marfurt (2006), 'Marmousi2: An elastic upgrade for Marmousi', *The Leading Edge* **25**, 156–166.

C. A. Micchelli (1986), 'Interpolation of scattered data: Distance matrices and conditionally positive definite functions', *Constr. Approx.* **2**, 11–22.

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

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

C. Piret (2012), 'The orthogonal gradients method: A radial basis functions method for solving partial differential equations on arbitrary surfaces', *J. Comput. Phys.* **231**, 4662–4675.

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

M. J. D. Powell (2005), Five lectures on radial basis functions. Technical report, Technical University of Denmark, Lyngby.

H. Power and V. Barraco (2002), 'A comparison analysis between unsymmetric and symmetric radial basis function collocation methods for the numerical solution of partial differential equations', *Comput. Math. Appl.* **43**, 551–583.

L. F. Richardson (1911), 'The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam', *Phil. Trans. Royal Soc. London* **210**, 307–357.

J. D. Rodrigues, C. M. C. Roque and A. J. M. Ferreira (2013), 'An improved meshless method for the static and vibration analysis of plates', *Mechanics Based Design of Structures and Machines* **41**, 21–39.

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

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

R. Schaback and H. Wendland (2006), Kernel techniques: From machine learning to meshless methods. In *Acta Numerica*, Vol. 15, Cambridge University Press, pp. 543–639.

I. J. Schoenberg (1938), ‘Metric spaces and completely monotone functions’, *Ann. of Math.* **39**, 811–841.

V. Shankar, G. B. Wright, A. L. Fogelson and R. M. Kirby (2013), ‘A study of different modeling choices for simulating platelets within the immersed boundary method’, *Appl. Numer. Math.* **63**, 58–77.

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

W. C. Skamarock and J. B. Klemp (2008), ‘A time-split nonhydrostatic atmospheric model for weather research and forecasting applications’, *J. Comput. Phys.* **227**, 3465–3485.

K. Stemmer, H. Harder and U. Hansen (2006), ‘A new method to simulate convection with strongly temperature- and pressure-dependent viscosity in a spherical shell: Applications to the Earth’s mantle’, *Phys. Earth Planet. Inter.* **157**, 223–249.

J. Straka, R. Wilhelmson, L. Wicker, J. Anderson and K. Droegemeier (1993), ‘Numerical solutions of a nonlinear density current: A benchmark solution and comparisons’, *Internat. J. Numer. Meth. Fluids* **17**, 1–22.

W. W. Symes and T. Vdovina (2009), ‘Interface error analysis for numerical wave propagation’, *Comput. Geosci.* **13**, 363–370.

L. Takacs (1988), ‘Effects of using *a posteriori* methods for the conservation of integral invariants’, *Mon. Weather Rev.* **116**, 525–545.

A. E. Tarwater (1985), Parameter study of Hardy’s multiquadric method for scattered data interpolation. Technical report UCRL-54670, Lawrence Livermore National Laboratory.

A. I. Tolstykh (2000), ‘On using RBF-based differencing formulas for unstructured and mixed structured–unstructured grid calculations’, *Proc. 16th IMACS World Congress* **228**, 4606–4624.

A. I. Tolstykh and D. A. Shirobokov (2003), ‘On using radial basis functions in a “finite difference mode” with applications to elasticity problems’, *Comput. Mech.* **33**, 68–79.

L. N. Trefethen (2000), *Spectral Methods in MATLAB*, SIAM.

A. M. Turing (1952), ‘The chemical basis of morphogenesis’, *Phil. Trans Royal Soc. London* B **237**, 37–72.

J. G. Wang and G. R. Liu (2002), ‘A point interpolation meshless method based on radial basis functions’, *Internat. J. Numer. Meth. Engrg* **54**, 1623–1648.

H. Wendland (2005), Scattered Data Approximation. Vo. 17 of *Cambridge Monographs on Applied and Computational Mathematics*, Cambridge University Press.

D. L. Williamson, J. B. Drake, J. J. Hack, R. Jakob and P. N. Swarztrauber (1992), 'A standard test set for numerical approximations to the shallow water equations in spherical geometry', *J. Comput. Phys.* **102**, 211–224.

G. B. Wright (2003), Radial basis function interpolation: Numerical and analytical developments. PhD thesis, University of Colorado.

G. B. Wright and B. Fornberg (2006), 'Scattered node compact finite difference-type formulas generated from radial basis functions', *J. Comput. Phys.* **212**, 99–123.

G. B. Wright, N. Flyer and D. A. Yuen (2010), 'A hybrid radial basis function: Pseudospectral method for thermal convection in a 3D spherical shell', *Geochem. Geophys. Geosyst.* **11**, Q07003.

Z. Wu (1992), 'Hermite–Birkhoff interpolation of scattered data by radial basis functions', *Approx. Theory Appl.* **8**, 1–10.

Y. Yu and Z. Chen (2011), 'Implementation of material interface conditions in the radial point interpolation methless method', *IEEE Trans. Ant. Prop.* **59**, 2916–2923.

S. Zhai, X. Feng and Y. He (2013), 'A family of fourth-order and sixth-order compact difference schemes for the three-dimensional Poisson equation', *J. Sci. Comput.* **54**, 97–120.

S. Zhong, A. McNamara, E. Tan, L. Moresi and M. Gurnis (2008), 'A benchmark study on mantle convection in a 3-D spherical shell using CitcomS', *Geochem. Geophys. Geosyst.* **9**, Q10017.