

Approximation on non-tensor domains including squircles, Part III: Polynomial hyperinterpolation and radial basis function interpolation on Chebyshev-like grids and truncated uniform grids

Shan Li^a, John P. Boyd^{b,*}

^a College of Science, University of Shanghai for Science and Technology, Shanghai 200093, PR China

^b Department of Atmospheric, Oceanic & Space Science, University of Michigan, 2455 Hayward Avenue, Ann Arbor, MI 48109, United States

ARTICLE INFO

Article history:

Received 2 June 2014

Received in revised form 7 September 2014

Accepted 14 October 2014

Available online 23 October 2014

Keywords:

Pseudospectral

Hyperinterpolation

Chebyshev polynomials

Zernike polynomials

Radial basis functions

Group theory

D_4

Dihedral group

ABSTRACT

Single-domain spectral methods have been largely restricted to tensor product bases on a tensor product grid. To break the “tensor barrier”, we study approximation in a domain bounded by a “squircle”, the zero isoline of $B(x, y) = x^{2\nu} + y^{2\nu} - 1$. The boundary varies smoothly from a circle [$\nu = 1$] to the square [$\nu = \infty$]. Polynomial least-squares hyperinterpolation converges geometrically as long as the number of points P is (at least) double the number of basis functions N . The polynomial grid was made denser near the boundaries (“Chebyshev-like”) by depositing grid points along wisely chosen contours of B . Gaussian radial basis functions (RBFs) were more robust in the sense that they, too, converged geometrically, but hyperinterpolation ($P > N$) and a Chebyshevized grid were unnecessary. A uniform grid, truncated to include only those points within the squircle, was satisfactory even without interpolation points on the boundary (although boundary points are a cost-effective improvement). For a given number of points P , however, RBF interpolation was only slightly more accurate than polynomial hyperinterpolation, and needed twice as many basis functions. Interpolation costs can be greatly reduced by exploiting the invariance of the squircle-bounded domain to the eight elements D_4 dihedral group.

© 2014 Elsevier Inc. All rights reserved.

1. Introduction

Spectral methods have been hitherto limited to tensor product domains such as rectangles, disks and triangles. A major research frontier is to extend single domain spectral methods to more complicated geometries. Because this is a very difficult problem, we begin with two simple, simply connected geometries: the domain bounded by a squircle and a quadrifolium perturbed by a disk. (Both are defined and illustrated in Section 2.)

In Part II, we compared two polynomial basis sets. The Chebyshev tensor product is optimum for the square; how good is it for a disk? The Zernike polynomials are optimum for the disk; will they fail for the square? Our conclusion was that, rather to our surprise, there was little difference between the two, although the Chebyshev is less sensitive to the ratio

* Corresponding author.

E-mail address: jboyd@umich.edu (J.P. Boyd).

Table 1
Notation.

$B(x, y)$	Boundary function: its zero isoline is the boundary
D	Total degree of a polynomial (for x^{m+n} , $D = m + n$)
h	average grid spacing
M	Numbers of points in x (or y) for a tensor product grid on the square
N	Total number of basis functions
P	Total number of interpolation points
q	Parameterization for the squircle: $r = q(\theta; \nu) \equiv \{\cos^{2\nu}(\theta) + \sin^{2\nu}(\theta)\}^{-1/(2\nu)}$
r	Radial coordinate in polar coordinates
R	Radial coordinate in the transformed polar coordinates
T	Computational polar coordinate, the angle
$T_n(x)$	Chebyshev polynomial of degree n
$Z_n^m(x, y)$	Zernike polynomial of angular wavenumber m and total degree n
$Z_n^m(r, \theta)$	Zernike disk polynomial expressed in polar coordinates
α	RBF “shape parameter”; the inverse width relative to the grid spacing h
γ	P/N
$B_j(x)$	Generic basis functions
δ	shape parameter for perturbed quadrifolium boundary curve
$\phi(r)$	RBF Kernel; $\phi(r) \equiv \exp(-[\alpha/h]^2 r^2)$ here
α	relative shape parameter, ϵ divided by average grid spacing
ϵ	absolute shape parameter (inverse width)
θ	Angular coordinate in polar coordinates

of points P to basis functions N . A total degree truncation was used for both where the “total degree” is the sum of the degrees in x and y for reasons explained in Part I [24]. Since the Zernike polynomials are more complicated and much less familiar, all polynomial calculations here will use tensor products of Chebyshev polynomials, $T_m(x)T_n(y)$.

However, radial basis functions have become the “go-to” tool for irregular grid interpolation. A major theme here is to compare polynomial and RBF algorithms: Do RBFs deserve their high reputation?

We will compare the following grids:

1. Truncated uniform square grid without boundary points.
2. Same as the previous but including boundary points.
3. Grids with Chebyshev-like boundary-biasing.

Besides interpolation, we shall also employ “hyperinterpolation” or more precisely, “least-squares hyperinterpolation”. Ian Sloan coined “hyperinterpolation” to describe approximations where the coefficients of the basis functions are computed by numerical quadratures using a set of points whose cardinality exceeds that of the basis set [34]. Here, we do not use quadratures, but rather least-squares approximation with more interpolation conditions than basis coefficients.

If the number of points P equals N , the number of basis functions, the result is “interpolation”. If $P < N$ so that the matrix system is underdetermined, the approximation is “hypointerpolation”. If $P > N$, the result is “hyperinterpolation”.

There are useful strategies that discussed in Part II, but not applied here either because of incompatibility with the D_4 group symmetry (hexagonal grid) [17,2] or to avoid prolonging this article to excessive length (Fourier Extension) [1, 21,7,8], RBF Extension [10], mapping the domain to a square or disk through an analytic change of coordinates [28,26]. The Gram–Schmidt process can be applied to construct polynomials orthogonal on the squircle, similar to [27,13] and our experiments will be reported in due course.

Thus, our analysis will be incomplete. Nevertheless, we shall present a variety of new ideas that do break the “rectangularity barrier”.

In Part I [24], we explained how to exploit the D_4 symmetry group. A symmetry-respecting grid can be generated by first creating a grid in a triangular sector bounded by the x -axis and the diagonal line $y = x$. By applying the group reflections, one can populate the rest of the grid. We further showed how to decompose an arbitrary function into six parts which provide the input to six smaller interpolation problems. Lastly, we created linear combinations of the basis functions that are eigenfunctions of the reflections of the symmetry group; we provided the appropriate formulas for radial basis functions, tensor products of Chebyshev polynomials and Zernike polynomials. We will continue to exploit symmetry here, but it will not be a major theme. (Notation is catalogued in Table 1.)

2. Squircles and the perturbed quadrifolium: two simply-connected domains bounded by plane algebraic curves

Definition 1 (Squircle). A squircle is the plane algebraic curve defined implicitly as the set of all points satisfying the equation

$$x^{2\nu} + y^{2\nu} = 1 \quad (1)$$

As ν increases, the curve becomes more and more like a square with slightly rounded corners; the limit as $\nu \rightarrow \infty$ is a square. The word “squircle” is a portmanteau of the words, “square” and “circle”. The squircle is also known as a “Lamé curve” or “Lamé oval”, and also as a “Fermat curve”. It is a special case of the superellipse.

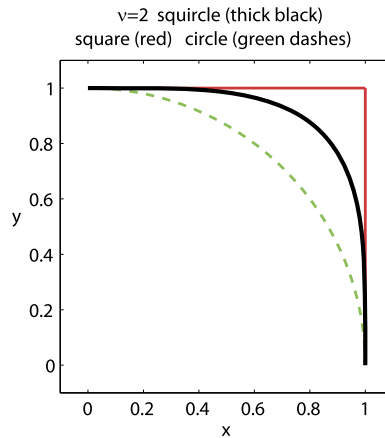


Fig. 1. Squircles in the upper right quadrant for $\nu = 1$ [circle], $\nu = 2$ [a particular instance of the squircle], and $\nu = \infty$ [square]. Only one-quarter of each curve is illustrated because squircles are symmetric with respect to reflection about both coordinates axes (and also the diagonal line, $y = x$). (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

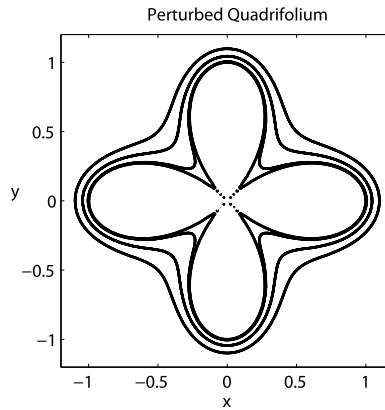


Fig. 2. Four realizations of the quadrifolium perturbed by a circle, an algebraic curve defined as the zero isoline of $B(x, y) = \delta(x^2 + y^2) - ((x^2 + y^2)^3 - (x^2 - y^2)^2)$ for $\delta = 1/4$ [outermost contour], $\delta = 1/10$, $\delta = 1/100$ and $\delta = 10^{-9}$ [innermost contour].

In polar coordinates, the parametric equation is

$$r = q(\theta; \nu) \equiv \{\cos^{2\nu}(\theta) + \sin^{2\nu}(\theta)\}^{-1/(2\nu)} \quad (2)$$

Note that the terminology is not uniform. Some authors reserve the term “squircle” for the special case $\nu = 2$, and others prefer “supercircle”.

Gabriele Lamé began the study of squircles and its generalization in 1818 [22,19]. The Danish architect and poet Piet Hien incorporated superellipses (unequal axes) into some of his major projects [20]. Olympic Stadium in Mexico City is a famous example. Superellipses have been used for shape fitting [37] in computer graphics. Liew, Kitipornchai and Lim [25] compute vibrations of plates bounded by superellipses. Allen, Kundtz, Roberts, Cummer and Smith transformed electromagnetic sources using superellipses [3]. Although we shall restrict attention to equal axes, polynomial hyperinterpolation and RBF interpolation apply equally well (and badly) to superellipses, and not merely to squircles. Squircles are illustrated in Fig. 1.

Another useful one-parameter family of D_4 invariant boundaries was obtained by adding a circle [more precisely, a radially symmetric quadratic] to the quadrifolium:

$$B(x, y) = \delta(x^2 + y^2) - ((x^2 + y^2)^3 - (x^2 - y^2)^2) \quad (3)$$

where quadrifolium is the limit $\delta \rightarrow 0$ and a circle of radius $\delta^{1/4}$ is the limit $\delta \rightarrow \infty$. The curve can also be parameterized as $r(\theta)$ where

$$r = (1/2)\sqrt{1 + \cos(4\theta) + \sqrt{16\delta + 1 + 2\cos(4\theta) + \cos^2(4\theta)}} \quad (4)$$

Several curves in this family are illustrated in Fig. 2.

3. Uniform grids

3.1. Truncated uniform square grid

The uniform square grid is constructed in two stages. First, we construct a uniform grid for the *unit square* by choosing an integer M and then defining

$$x_j^s = -1 + 2(j-1)/(M-1); \quad y_k^s = -1 + 2(k-1)/(M-1) \quad (5)$$

Second, we test each point on the uniform square grid for membership in the interior of the domain by evaluating the function that implicitly defines the squircle,

$$B(x, y) \equiv 1 - x^{2\nu} - y^{2\nu} \quad (6)$$

A point is on the interior if and only if $B(x, y) > 0$. A counter is increased by one every time a point passes the test so that at the end of the double loop the counter is equal to N_I , the number of interior points.

Gridgeman shows that the area of a domain bounded by a squircle is [20]

$$A = 4 \int_0^1 (1 - x^{2\nu})^{1/(2\nu)} dx = \frac{4^{1-1/(2\nu)} \sqrt{\pi} \Gamma(1 + 1/(2\nu))}{\Gamma(1/2 + 1/(2\nu))} = \frac{(\Gamma(1/2 \nu^{-1}))^2}{\nu (\Gamma(\nu^{-1}))} \quad (7)$$

In the limit that the number of interpolation points is large, the number of included points will asymptote to $[A(\nu)/4]N$. (The area of the embedding square is 4.)

3.2. Boundary points

When there are no points on the boundary itself, all interpolation methods become *extrapolation* methods in the sense that the approximation must be evaluated outside the convex hull of the interpolation points. Obviously, as one moves farther and farther outside the region of the samples of $f(x, y)$, the error in approximating $f(x, y)$ will grow rapidly.

Therefore is highly desirable to place interpolation points on the boundary itself. This entails some difficulties:

1. One needs either an explicit parametric representation of the boundary or a spectrally-accurate method for calculating the zero contour line of the implicit boundary function $B(x, y)$
2. Boundary-gridding is a second, separate stage; it is additional work superimposed on the task of defining grid points for the interior.
3. The choice of N_B , the number of boundary points, is another numerical free parameter.

For the squircle, both implicit and explicit parametric representations are available, so the first difficulty disappears. For the square, extending a tensor product $M \times M$ grid to the boundaries automatically yields $4M - 4$ boundary points, or in other words, using $N_I = (M-2)^2$ as the total number of interior grid points,

$$N_B \sim 4\sqrt{M}, \quad N_I \gg 1 \quad (8)$$

This provides a useful guideline for the squircle. For more complicated geometries, however, extensive experimentation may be necessary to optimize the number of boundary points.

4. Chebyshev-like grids for domains with implicitly-specified boundaries

“The residual is typically largest near the boundary (by one or two orders) larger than in the domain far away from the boundary.”

– Fedoseyev, Friedman and Kansa (2002), pg. 440. [15]; see also [16].

This observation motivates the Chebyshev-like grid described in this section.

4.1. Motivation: the superiority of boundary-biased grids

In one dimension, it has been known since the work of Runge more than a century ago that uniform grid polynomial interpolation on a *uniform* grid is a really bad idea. Runge showed that the interpolant *diverges* on part of the interval spanned by the interpolation points if $f(x)$ were singular anywhere within a lens-shaped region in the complex plane, the “Runge Zone”. This divergence will occur even if $f(x)$ is analytic everywhere on the interpolation interval itself. Furthermore, the interpolation matrix is very ill conditioned. In floating-point arithmetic, polynomial interpolation on a uniform grid diverges as $N \rightarrow \infty$ even for entire functions (for which theory demands convergence in infinite precision arithmetic).

The remedy is to use as interpolation points the roots of the N -th Chebyshev polynomial or alternatively, the endpoints plus the zeros of the first derivative of a Chebyshev polynomial. The Chebyshev grids are very non-uniform with a grid spacing near the endpoints which is proportional to $1/N^2$ in contrast to the $2/N$ separation between neighboring grid points when the grid is uniform.

Fortunately, radial basis functions are much more grid-tolerant than polynomials, and we shall present successful applications of RBFs on two-dimensional uniform grids. However, there is a fundamental difficulty with the uniform grid which cannot be resolved merely by changing the basis. This difficulty is that at a grid point near the center of an interval, information from both sides of the point is utilized. At an endpoint, information in the form of samples of $f(x)$ is available only from one side of the endpoint. For every sample $f(x + jh)$ a distance jh from the endpoint, the centered approximation has two samples at $f(x \pm jh)$.

The result is the well-known fact that one-sided finite differences are much less accurate than centered differences of the same stencil width. Although, for the first derivative on a uniform grid, a centered seven-point formula and a one-sided seven-point formula are both formally sixth order with errors proportional to $c_m h^6$, the proportionality constant c_m is twenty times larger for the one-sided stencil, and the difference grows very rapidly with increasing order.

Unsurprisingly, it is well-known in the RBF community that accuracy can be greatly improved by making the grid *non-uniform* and *denser near the boundaries*. But how is one to create a boundary-concentrated grid when the boundary geometry is complicated?

4.2. Exploiting the isolines of the boundary-defining function

When the boundary of the domain is defined as an isoline of a function $B(x, y)$, we can use other isolines of the function, at least close to the boundary, to lay down a computational grid which is dense near the boundary. As indicated by the quote from Fedoseyev, Friedman and Kansa (2002) above [15], the errors in RBF interpolation are concentrated near the boundary when the grid is uniform. Accuracy is often significantly improved by making a grid which is “Chebyshev-like” in the sense of having a high concentration of points near the boundary and a lower density of interpolation points away from the boundary.

For the squircle-bounded domain, it is convenient to add a constant to our boundary function $B(x, y)$ to shift the value of the boundary isoline from zero to one

$$\Psi(x, y; \nu) \equiv x^{2\nu} + y^{2\nu} = B(x, y; \nu) + 1 \quad (9)$$

where the boundary is defined as the contour $\Psi = 1$; the grid points will all lie on contours of Ψ for contour values c less than one.

To choose a good distribution of contour levels c , specialize to the x -axis where Ψ simplifies to $x^{2\nu}$. Along this axis, the contours of Ψ satisfy

$$x^{2\nu} = c \quad (10)$$

where c is the contour level. The points of a Chebyshev–Lobatto grid are given by

$$x_j = \cos(\pi(j-1)/(2M-1)), \quad j = 1, \dots, M \quad (11)$$

where M is the number of points on $x \in [0, 1]$ and all the $x_j \in [0, 1]$. It follows that the Ψ levels that will give the contours a Chebyshev-like spacing along the x -axis are

$$c_j = (x_j)^{2\nu} = \Psi(x_j, 0), \quad j = 1, \dots, M \quad (12)$$

Let us now choose to distribute M_j grid points uniformly in polar angle θ along each contour of Ψ . In polar coordinates,

$$x_{jk} = r_{jk} \cos(2\pi k/M_j), \quad y_{jk} = r_{jk} \sin(2\pi k/M_j), \quad j = 1, \dots, M; \quad k = 1, \dots, M_j \quad (13)$$

where r_{jk} is chosen so

$$\Psi(x_{jk}, y_{jk}; \nu) = c_j(\nu) \quad (14)$$

This gives

$$r_{jk} = \left(\frac{c_j(\nu)}{\cos^{2\nu}(2\pi k/M_j) + \sin^{2\nu}(2\pi k/M_j)} \right)^{1/(2\nu)} \quad (15)$$

Another issue is how to choose the numbers M_j of grid points to put on their j -th contour. A simple but not optimum choice is to choose $M_j = M$, a constant independent of j . The choice we actually made was

$$M_j = kj, \quad k = \text{positive integer} \quad (16)$$

Thus, longer contour curves (closer to the squircle boundary) have more points than the short isolines near the center of the domain.

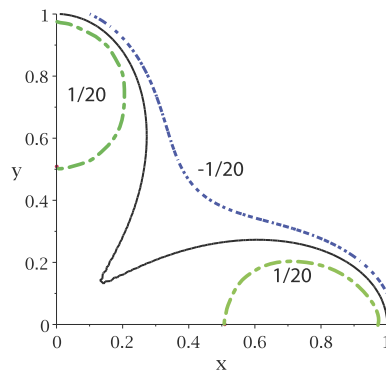


Fig. 3. The solid black curve is the zero isoline of $B(x, y)$, i.e., the domain boundary, for $\delta = 1/1000$. The dashed curves show two other isolines, $B = \pm 1/20$ as marked.

4.3. Generalizations: when the boundary is a parametric curve

The boundary-concentrating strategy using the isolines of the function that implicitly defines the boundary, $B(x, y)$, is very general, and not at all restricted to a squircle. However, curves are equally often specified by a pair of univariate functions:

$$x = X(t) \quad y = Y(t), \quad t \in [0, 1] \quad (17)$$

where t is the parameter. What do we do when the boundary is specified as an explicit, parametric curve?

The answer is that the implicit and parametric representations are both useful, and so a standard problem in computer graphics is to derive an implicit representation from a parametric representation (and vice-versa) [32,33]. If the parametric functions are smooth, they can always be approximated by polynomials. The resultant $B(x, y)$ of the two polynomials is zero only on the curve and thus furnishes the desired implicit representation:

$$B(x, y) = \text{Resultant}(x - X(t), y - Y(t), t) \quad (18)$$

where the third argument denotes the variable which is eliminated in the resultant computation, here t , and $B(x, y)$ so defined is a bivariate polynomial.

Resultants are ill-conditioned, so there are practical difficulties, and implicitization strategies are still a research activity. Nevertheless, it is usually possible with the present state of the art to generate useful $B(x, y)$ from a parametrically-specified boundary.

4.4. Failure for the perturbed quadrifolium

It is good that a truncated uniform grid plays nicely with polynomials, as shown in Section 6. Fig. 3 shows that using the contours of $B(x, y)$ to create a “Chebyshevized” grid is not a good option here for either interior or exterior problems for the quadrifolium for $\delta = 1/1000$. The contours of B are most useful for gridding when the near-boundary contours roughly parallel the boundary. Here, contours that are very close to the boundary near the x and y axes curve away from the boundary near the diagonal. This would leave huge voids in the grid near the “V” shape of the boundary. This sharp indentation is precisely where we want a *high* density of grid points to resolve high-gradients where the boundary is nearly discontinuous. The Chebyshev-like contours of B define a gridding scheme that is always simple, but useful only some of the time. However, generating a boundary-concentrated grid in complicated geometry is hard, and we should not discard a tool that works well frequently.

5. Numerical experiments

5.1. Introduction: cases

To compare methods, we will discuss four functions $f(x, y)$ cataloged in Table 2. Since one of our themes is to exploit symmetry to the maximum, all four functions are invariant under all eight operations of the D_4 symmetry group. The contour plot of $f_3(x, y)$ in Fig. 4 explicitly displays the invariance with respect to reflections about both coordinate axes and also both diagonals.

A common mistake is to choose exemplars that are combinations of sines and cosines and exponentials; these are very special by being entire functions, free of singularities except at infinity. In one dimension, it is well-known that spectral series converge at a “supergeometric” rate for entire functions. However, the generic case is that $f(x)$ has poles or branch

Table 2
Test functions.

Number	Explicit expression	Notes
f_1	$1/(2+x^2+y^2)$	Singular on a complex-valued surface
f_2	$1/(2-x^2-y^2)$	Singular at corners of unit square
f_3	$\sin(8(x^4+y^4))\cos(8(x^2+y^2))/(2+x^2+y^2)$	bidirectional oscillations & complex-valued singularities
f_4	$1/(1+9[x^2+y^2])$	polynomial interpolation of $f_4(x=0, y)$ diverges

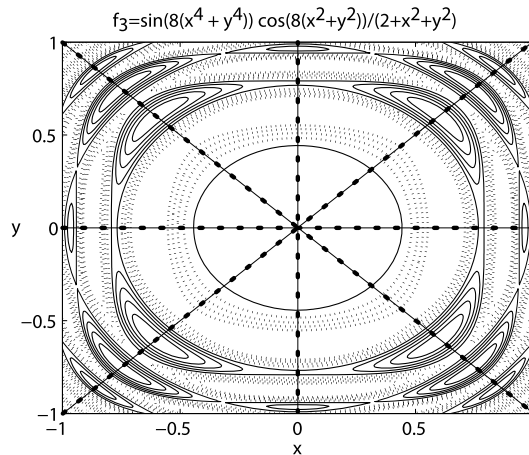


Fig. 4. Contour plot of $f_3(x, y) = \sin(8(x^4 + y^4)) \cos(8(x^2 + y^2)) / (2 + x^2 + y^2)$. Negative isolines are dashed. The contours are $-0.8:0.2:0.8$, scaled by the maximum of the function. This and all the other exemplary functions are invariant under all 8 operations of the D_4 symmetry group; invariant to reflection about the x and y axes and also invariant under reflection about the line $y = x$. The symmetry axes are the dotted lines. Note that reflection about $y = -x$ can be performed as the product of reflections about the other three axes, so the number of independent operations for the group is only eight.

points off the expansion interval but at a finite point in the complex plane. The result of such singularities is geometric convergence, that is, the error is proportional to $\exp(-qN)$ for some constant $q > 0$. So, all our examples contained singularities at either finite points or on finite curves outside the squircle.

5.2. Polynomial hyperinterpolation on the squircle

Fig. 5 shows the maximum pointwise errors for polynomial hyperinterpolation with various combinations of points P and basis functions N . When N is near P , i.e., close to interpolation ($P = N$), the errors are $O(1)$. When $N \leq P/2$, we obtain the exponential rate of convergence shown by the dashed line, which is

$$E_{\text{fit}}(P) = (1/2) \exp(-3\sqrt{P}) \quad (19)$$

In one dimension, the spectral approximation converges geometrically, that is, errors falling as $\exp(-qN) = \exp(-qP)$ since $N = P$ for interpolation. Similarly, with total degree truncation at degree D in d dimensions, the error can usually be bounded by $\exp(-qD)$. The subtlety is that the number of basis functions is $O(D^d)$. For either interpolation or hyperinterpolation, the number of points $P \sim O(D^d)$ also. Thus, in two dimensions, $\exp(-qD) = \exp(-q'\sqrt{P})$. For this reason, the horizontal axis is not the number of hyperinterpolation points P , but rather $\sqrt{P}$. The fitted curve shows that with a sufficiently large ratio of P/N , the convergence is as fast as can be expected, and merits the label “geometric”.

Although f_1 is singular everywhere along a curve in the complex plane whereas f_2 is singular at four real points, the corners of the unit square, results were so similar that we shall not show pictures for f_2 . Instead, we show the scatterplot for the highly oscillatory function $f_3(x, y)$ as Fig. 6.

The scatterplots are useful for showing geometric convergence, but useless for choosing $N(P)$. The need for hyperinterpolation – the dependence of error on the ratio of points to basis functions, $\gamma = P/N$ – is shown more clearly by plotting error versus γ for a fixed number of basis functions as in Fig. 7. Interpolation ($P = N$) fails; hyperinterpolation with $\gamma > 2$ is wasteful. A good choice for the number of points is double the number of basis functions, $P \approx 2N$.

5.3. Grids

Fig. 8 shows on the left the Chebyshev-like, boundary-concentrated grid used for polynomial hyperinterpolation. The right shows the truncated uniform grid that will be used for RBF interpolation in the remainder of this section. It is unlikely that either grid is optimum for its associated basis on the squircle, but both are simple to construct and “good enough”.

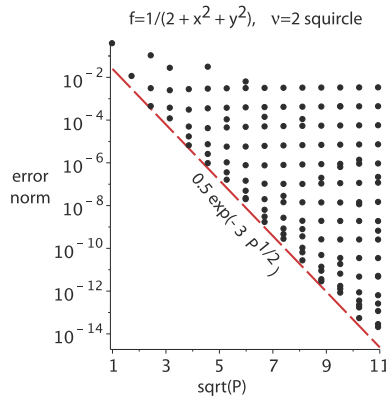


Fig. 5. Hyperinterpolation of $f_1(x, y) = 1/(2 + x^2 + y^2)$ using a tensor product Chebyshev basis with total degree truncation, symmetrized to be invariant to all reflections and rotations of the full 8-fold D_4 symmetry group. Each vertical row of dots represents a fixed number P of interpolation points, but varying numbers N of basis functions. The error is the usual L_∞ error norm, the maximum pointwise error in and on the squircle. The dashed line connecting the most accurate (lowest points) results for each P shows that the error is converging exponentially as $P \rightarrow \infty$. Nonoptimum N have been included to motivate further studies of the relationship between accuracy and the number of points P and basis size N . Optimizing $N(P)$ is the motive for later figures in this article. The motive of this scatterplot is to (i) show that exponential convergence is possible and (ii) there are large variations in error as N is varied. This is unfortunate because there is no theory for choosing $N_{\text{optimum}}(P)$, that is, there is no rule one can look up in a reference book, or even a research paper, that specifies N for each P so as to minimize the error for a given number of points P .

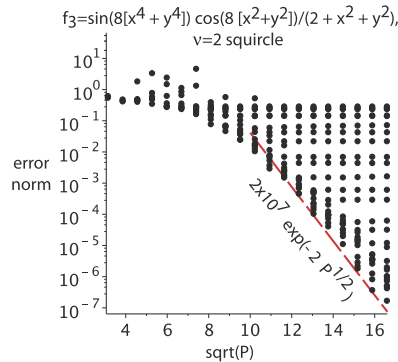


Fig. 6. Same as previous figure, but for a different $f(x, y)$. Maximum pointwise errors for polynomial approximations to $f_3(x, y) = \sin(8(x^4 + y^4)) \cos(8(x^2 + y^2)) / (2 + x^2 + y^2)$ in a $v = 2$ squircle. The polynomials were represented as a tensor product Chebyshev basis with total degree truncation and full 8-fold D_4 symmetry for both grid and basis functions using the Chebyshev-squircle grid. For each choice of the number of interpolation points, the vertical stack of dots displays the errors for different number of basis functions N for that number of points P .

5.4. RBF interpolation without boundary points

Radial basis functions (RBFs) have been described in standard reviews and textbooks [12,31,36] and in Parts I and II of this series, [24,23], so our discussion of RBF fundamentals is brief. We choose Gaussian functions with the *univariate* kernel

$$\phi(r; \alpha, h) \equiv \exp(-[\alpha/h]^2 r^2) \quad (20)$$

where h is the grid spacing between nearest neighbor grid points on a uniform grid or the average grid spacing if the grid is non-uniform. The constant α is the “shape parameter”, which is actually an inverse width relative to the average grid spacing. The basis functions ϕ_j in two dimensions are translated copies of the kernel function, $\phi(\sqrt{(x - x_j)^2 + (y - y_j)^2})$ where the “center” (x_j, y_j) is always one of the interpolation points.

Fig. 9 shows the isolines of the base-10 logarithm of the L_∞ error norm for the approximation of $f_4 = 1/(1 + 9[x^2 + y^2])$. One-dimensional equispaced polynomial interpolation along either the x or y axis is known to fail from theory [35]. It is encouraging that RBF interpolation is successful [for moderate shape parameter α] even with the crudest grid: a uniform grid pruned of all points outside the boundary squircle with no points on the squircle itself. Note that the method fails for small α , the near-polynomial regime, in spite of hundred-digit arithmetic. The axis of optimum accuracy for a given N veers to smaller RBF inverse width α as N increases.

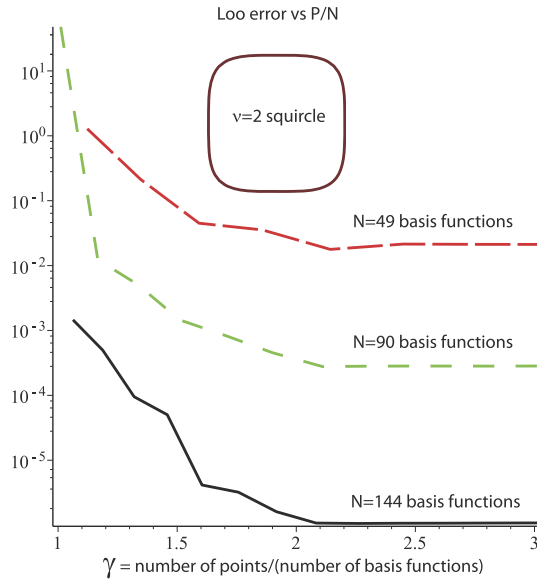


Fig. 7. Same case (Three) as the previous figure, $f_3(x, y) = \sin(8(x^4 + y^4)) \cos(8(x^2 + y^2)) / (2 + x^2 + y^2)$ in a $v = 2$ squircle, whose shape is depicted as the inset. Maximum pointwise error versus the number of interpolation points is shown for three different numbers of basis functions. $N = 49$: long red dashes. $N = 90$: green dashed. $N = 144$: black thick curve. Tensor product Chebyshev basis with total degree truncation, symmetrized to be invariant under the full eight-element D_4 symmetry group. There are $3j$ points on the j -th contour of B where the j -th contour is the curve $x^{2v} + y^{2v} = \cos^{2v}(\pi(j - 1)/(2N - 1))$, $j = 1, \dots, N$. All points of the Chebyshev-like/ Ψ -contours grid lie in one-eighth of the squircle-bounded domain. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

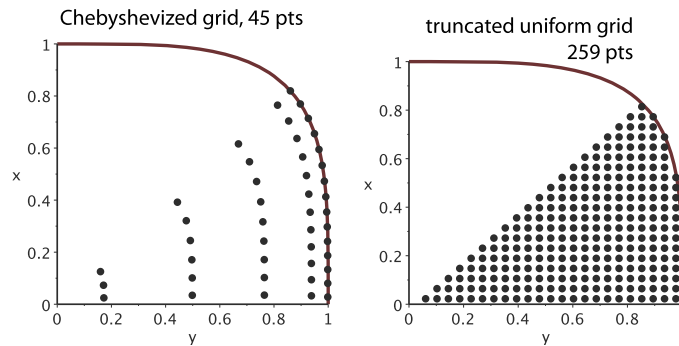


Fig. 8. Left: a Chebyshev-like grid for the $v = 2$ squircle. The grid points lie on contours of the function whose zero isoline implicitly defines the squircle. The grid is “Chebyshevized” in the sense that the density of points is much higher near the exterior boundary than elsewhere. Right: a truncated uniform grid. Only $1/8$ of the domain is gridded because we exploited the invariance of the squircle to the eight-element D_4 symmetry group.

5.5. RBF interpolation with boundary points

Each of the six thick lines in Fig. 10 connects the error norms for a group of experiments with an *identical* number of interior points but *different* numbers of boundary points. (All computations in a group except the first also used $(0, 0)$ as an interpolation point.) Each group shows that adding a few boundary points significantly reduces error. However, for a given number of interior points N_I , a plateau is quickly reached such that adding more boundary points produces no further decrease in error.

Fig. 11 shows that more points is better, but the addition of boundary points has not merely reduced the maximum pointwise error. Adding points near the boundary has rendered the error much more *uniform* over the spatial domain.

5.6. Comparisons between polynomial hyperinterpolation and RBF interpolation

Fig. 12 compares RBF interpolation and polynomial hyperinterpolation for a different example. For RBF interpolation, the Chebyshev-like grid leads to a faster rate of convergence. The Chebyshev scheme converges no faster, but its error is an order-of-magnitude below that of the RBF interpolation with the same number of points.

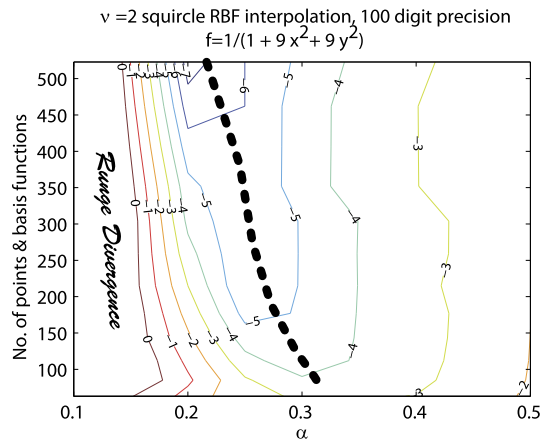


Fig. 9. Isolines of the base 10 logarithm of the L_∞ error norm for very high precision (Digits = 100) Gaussian RBF interpolation [not hyperinterpolation] on the squircle. Truncated uniform square grid with no interpolation points on the boundary. Centers and interpolation points coincide. The thick dashed curve is the “axis of optimality”.

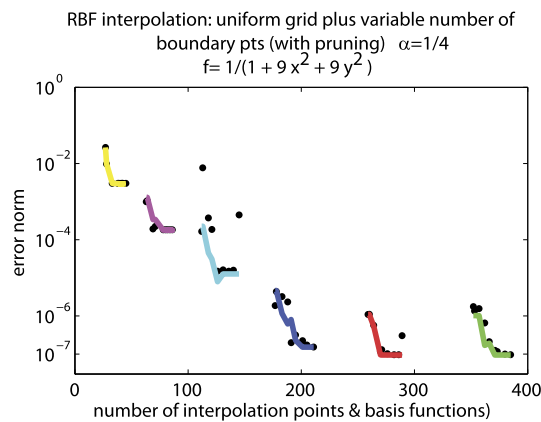


Fig. 10. Error norms for RBF interpolation of $f(x, y) = 1/(1 + 9x^2 + 9y^2)$, plotted versus the total number of interpolation points including both interior and boundary points. Each solid curve denotes a group that employs the same number of interior points, but the number of boundary points increases to the right. The curves were computed in 32-digit precision; the black dots show 16-digit computations where these differed (always larger error) from the higher precision computations.

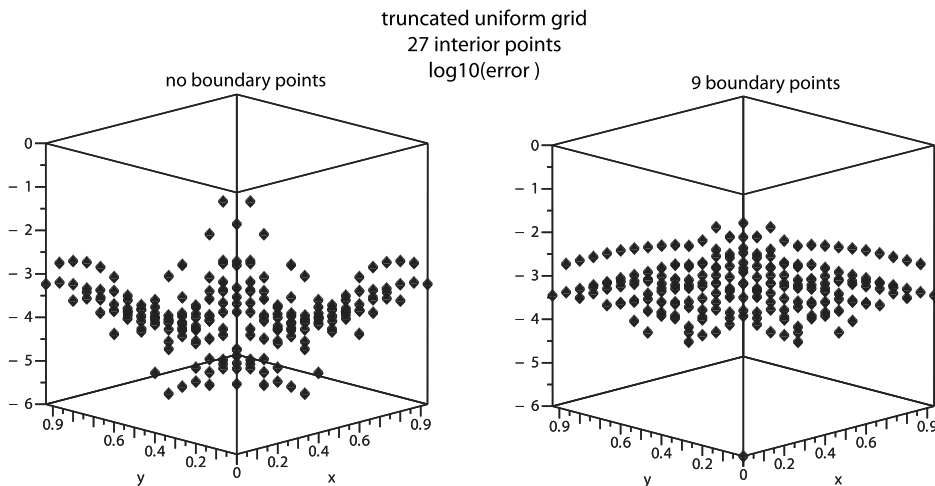


Fig. 11. Errors for RBF interpolation of $f_4 = 1/(1 + 9x^2 + 9y^2)$. Both used 27 interior grid points, derived from the truncated uniform grid, on one-eighth of the domain bounded by an $\nu = 2$ squircle. The left plot used no boundary points; the right plot used 9 boundary points. The errors were calculated on a high resolution graphing grid with many more points than used for interpolation.

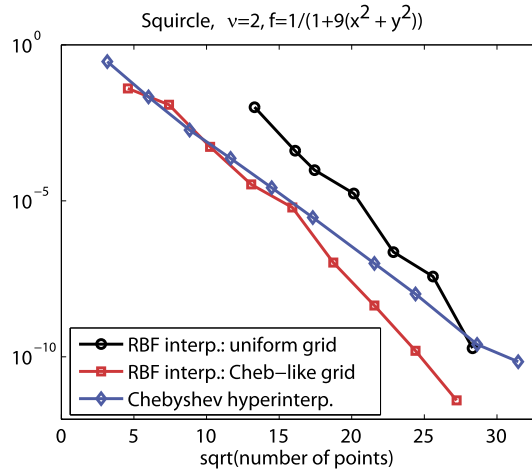


Fig. 12. Maximum pointwise errors (L_∞ norm of the errors) for three different approximations to $f_4(x, y) = 1/(1 + 9(x^2 + y^2))$. The red dashed curve [RBF interpolation on a Chebyshev grid] is superior to the black curve, which is RBF interpolation with a uniform grid. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

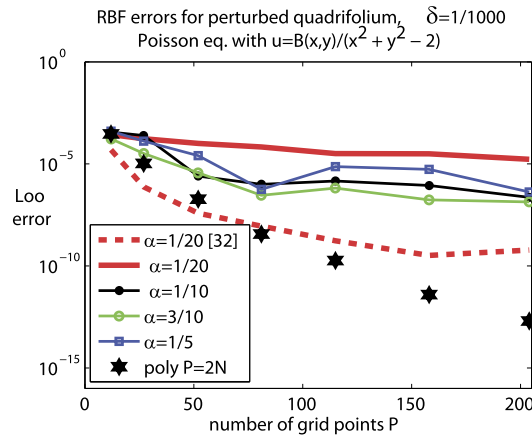


Fig. 13. Maximum pointwise errors (L_∞ norm of the errors) for RBF and polynomial approximations to the solution of a Poisson equation where the exact solution is $u = B(x, y; \delta)f(x, y)$ where $f(x, y) = 1/(x^2 + y^2 - 2)$ and $\delta = 1/1000$. The polynomials used a number of basis functions roughly half the number of interpolation points P . Both RBF and polynomials used the same truncated uniform grids of various densities.

6. Perturbed quadrifolium

We also solved PDEs such as the Poisson equation in various geometries with various basis sets. For the PDE solutions reported here, the homogeneous boundary conditions were imposed by Krylov's method in which the approximation with polynomial or RBF basis functions ϕ_j is written as

$$u(x, y) = B(x, y) \sum_{j=1}^N a_j \phi_j(x, y) \quad (21)$$

Fig. 13 shows that RBF interpolation and polynomial hyperinterpolation are also successful for solving PDEs, here the Poisson equation, when the boundary is the plane algebraic curve defined implicitly by

$$B(x, y) = \delta(x^2 + y^2) - ((x^2 + y^2)^3 - (x^2 - y^2)^2) \quad (22)$$

This is true even though $\delta = 1/1000$ so that the curve bends sharply towards the origin along the rays $x = \pm y$.

Remarkably, polynomial interpolation is more accurate for a given number of points than RBFs.

We also solve the Poisson equation when the exact solution was

$$u = B(x, y; \delta) \frac{\sin(8(x^4 + y^4)) \cos(8(x^2 + y^2))}{x^2 + y^2 + 2} \quad (23)$$

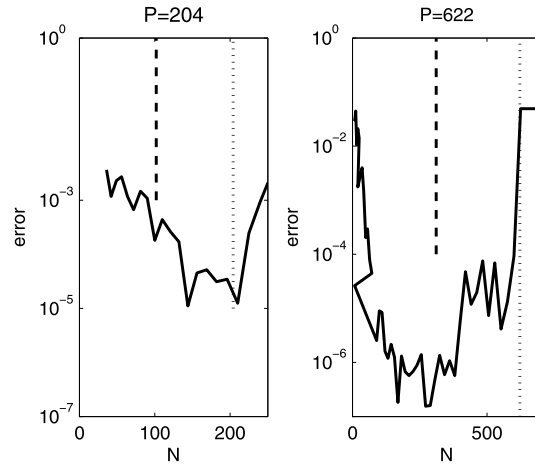


Fig. 14. Perturbed quadrifolium domain with $\delta = 1/1000$. Maximum pointwise errors (L_∞ norm of the errors) for polynomial approximations to the solution of a Poisson equation where the exact solution is $u = B(x, y; \delta)f(x, y)$ where $f(x, y) = \sin(8(x^4 + y^4)) \cos(8(x^2 + y^2))/(x^2 + y^2 + 2)$, plotted versus the number of basis functions. The maximum pointwise value of u on the domain is 0.049. P is the number of points on a truncated uniform grid – 204 points in the octant on the left and on the right, 622 points in the octant, equivalent by symmetry to 4976 points over the entire interior of the curve. The divided vertical lines mark $N = P/2$; the dotted lines are at $N = P$. All calculations in 16 decimal digit precision.

Table 3

Maximum pointwise errors in RBF solutions to the Poisson equation on a domain bounded by a perturbed quadrifolium.

P	Digits	$\alpha = 0.05$	$\alpha = 0.10$	$\alpha = 0.15$	$\alpha = 0.20$	$\alpha = 0.25$	$\alpha = 0.30$
260	16	0.84	0.011	6.6E–5	0.15	6.09E–5	1.1E–4
"	24	4.68E–02	5.93E–05	3.34E–05	5.00E–06	1.49E–04	1.53E–04
"	32	2.3E–3	2.1E–4	5.4E–6	1.4E–5	1.5E–4	1.5E–4
458	16	0.44	3.6E–4	8.6E–6	2.2E–6	2.5E–5	3.4E–5
"	32	2.03E–4	1.27E–6	5.43E–7	1.49E–6	1.72E–5	2.03E–4
"	64	2.50E–8	2.93E–11				
"	96	1.53E–10	7.35E–10	6.85E–8			

Fig. 14 shows, anomalously, that polynomial interpolation ($P = N$) is more accurate than hyperinterpolation for small P . At the higher resolution [right panel], $P \approx 2N$ is again optimum.

The RBF computations are shown in Table 3. Hyperinterpolation was unnecessary. High precision is essential to achieve optimal results; with $\alpha = 1/10$, errors as small as 3×10^{-11} are possible, but only in sixty-four decimal digits of precision. However, a larger value of α – narrower RBFs – gives 10^{-5} to 10^{-6} error norms in Matlab/IEEE double precision of sixteen digits. The error norm shows considerable fluctuations as α , the precision and the number of points changes.

6.1. Helmholtz eigenfunctions on the perturbed quadrifolium

The Helmholtz eigenproblem is to solve

$$\nabla^2 u + \lambda u = 0, \quad u = 0 \quad \forall B(x, y; \delta) = 0 \quad (24)$$

The smallest eigenmode is the “ground state” in physics jargon; this is associated with the smallest eigenvalue and is usually the smoothest, the eigenfunction approximated with highest accuracy for a given numerical method at a given truncation.

We shall solve this by RBF methods first and then, in the next subsection, by a tensor product spectral method.

The truncated uniform grid gave mediocre results. It was well-nigh impossible to get accurate eigenvalues for $\delta = 1/1000$, so we retreated all the way to $\delta = 1$, which is only a modest perturbation of the unit disk. Even so, RBF methods were not impressively accurate with this grid as shown in Fig. 15.

It is well-known that for large RBF shape parameter α [RBFs narrow compared to the spacing between grid points], Gaussian approximation error “saturates”, that is, asymptotes to a non-zero constant. Here, the RBF calculations for $\alpha = 1/2$ and $\alpha = 0.4$ saturate at absolute errors of 0.18 [for $\alpha = 0.5$] and 0.0038 [for $\alpha = 0.4$]. The most accurate result is an absolute error of 2×10^{-4} for $\alpha = 0.3$ and a thousand basis functions. Ill-conditioning degrades accuracy for smaller α .

Ill-conditioning for small α for RBFs can be overcome by the algorithms described in [14,18], but these are expensive and blow up the simplicity. There is much to be said for multiple precision as in [30].

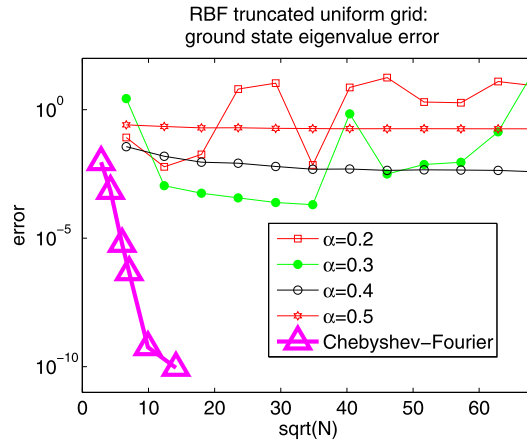


Fig. 15. Errors in approximating the ground state eigenvalue, $\lambda \approx 7.68$, of the Helmholtz equation in a domain bounded by the perturbed quadrifolium with $\delta = 1$. The horizontal axis is the square root of the number of basis functions. All computations were performed in sixteen digit precision.

Table 4

Absolute errors in the ground state of the Helmholtz equation Chebyshev–Fourier tensor product basis with basis recombination combined with a radial-stretch map; the radial factors are $T_{2n} - 1$. All calculations in 16 decimal precision. The first column lists the number of basis functions in radius $\times$ the number of basis functions in θ ; their product is the total number of tensor product basis functions, which is shown inside square brackets.

2×1 [2]	2.01
4×2 [8]	0.0091
6×3 [18]	0.00069
9×4 [36]	6.0×10^{-6}
8×4 [32]	6.9×10^{-6}
7×5 [35]	7.8×10^{-6}
6×6 [36]	5.0×10^{-5}
5×7 [35]	2.9×10^{-3}
10×5 [50]	4.7×10^{-7}
7×7 [49]	1.0×10^{-5}
14×7 [98]	5.7×10^{-10}
20×10 [200]	9.0×10^{-11} [Digits = 16],
"	3.8×10^{-14} [Digits = 24]

6.2. Helmholtz eigenproblem in the perturbed quadrifolium: comparisons with a tensor product spectral method

The RBF literature is full of sweeping claims that RBF methods are superior to conventional spectral methods most of the time. To test this claim, the same Helmholtz eigenproblem was also solved by a change-of-coordinates that mapped the boundary into the unit circle by stretching the radial coordinate only; the angle in polar coordinates was left unchanged by the transformation. A tensor product Chebyshev–Fourier pseudospectral method was then applied. To exploit the D_4 symmetry, only basis functions with the angular factors $\cos(4j\theta)$ with $j = 0, 1, 2, \dots$ were employed. To satisfy the homogeneous boundary conditions, the radial basis functions were

$$T_{2k}(r) - 1, \quad k = 1, 2, \dots \quad (25)$$

Table 4 shows the errors in the Chebyshev–Fourier basis for various resolutions.

For real-world applications where the exact eigenvalues are unknown, an essential safety procedure is to do every calculation *twice* at *different* resolutions. Here, we compare solutions in thirty-two decimal precision (i.e., the Maple variable “Digits” = 32) on a 30×15 grid [where the first integer is the number of radial degrees of freedom] with a 40×20 computation. If the errors falls geometrically with the square root of the total number of basis functions as is usual for smooth solutions (Chap. 2 of [6,9]), then denoting the errors in a given eigenvalue by E_1, E_2 ,

$$E_1 \sim aN_1^k \exp(-\mu\sqrt{N_1}), \quad E_2 \sim aN_2^k \exp(-\mu\sqrt{N_2}) \quad (26)$$

where the non-exponential factor N_1^k depends on the type of singularity; $k = 0$ for a simple pole, $k = -1$ for a logarithmic branch point and so. The constants a and μ are not known, which would seem to make these errors formulas almost useless, but they imply that the *separation* between two calculations of the same eigenvalue is of the form

Table 5

Eigenvalues and estimated errors for the ground state of the Helmholtz equation via Chebyshev–Fourier tensor product basis with basis recombination combined with a radial-stretch map; the radial factors are $T_{2n} - 1$. All calculations in 32 decimal precision. The first column is the mode number when the modes are ordered with smallest eigenvalue first where the eigenvalues are listed in the third column. The second column lists the difference (“separation”) between the eigenvalue as computed with forty basis functions in radius combined with twenty basis functions in angle and the same eigenvalue computed with a 30×15 resolution. These separations, if small, are good approximations to the error of the lower resolution calculation as explained in the text.

Mode number n	Separation	λ_n
1	0.663886e–20	7.686205475763327689527657
2	0.418694e–21	38.32805848007442721456331
3	0.402448e–18	70.97913795048395401621281
4	0.397036e–18	100.9746506812428448410525
5	0.991374e–18	137.5901469046520826390058
6	0.853487e–17	186.3942366785738161025437
7	0.553881e–17	202.4827138088717256855599
8	0.130549e–16	234.1303979784776391861648
9	0.112400e–15	295.1227936057382587592942
10	0.849775e–18	319.4577776164389782957925
11	0.131332e–15	349.1644789243148164179217
12	0.940910e–16	388.4172514927270676418426
13	0.871580e–15	436.5104100692903097673503
14	0.115465e–15	468.6873585148495205259377
15	0.260256e–14	493.2064535868689801188717
16	0.109545e–14	530.7264147405158005637296
17	0.357013e–14	589.9695217581549420809034
18	0.439360e–14	646.4356171870649174465085
19	0.140284e–14	634.6425557092395742312173
20	0.265120e–13	676.9608634027805005307136
21	0.239003e–13	722.5202284926238218475263
22	0.727533e–13	757.4756608298808093303245
23	0.388162e–13	806.0829213076128538096674
24	0.112242e–12	852.9892718015007028528237
25	0.184916e–11	890.6945024251483945875953
26	0.160660e–11	922.2429411150826069744504
27	0.187445e–11	948.1158893337261370049174
28	0.208239e–10	964.8062018554378067768950
29	0.167422e–10	1034.284729222475423125170
30	0.989814e–10	1088.674244677369154677464
31	0.139650e–9	1104.612536092546231028047
32	0.863359e–9	1141.681177977954840873482
33	0.648903e–9	1175.081890188335890611160
34	0.438977e–8	1209.563778974376963554548
35	0.527977e–8	1279.384934788311768009794
36	0.103442e–7	1295.511511341397952534471
37	0.747238e–7	1346.909667992594585617417
38	0.301903e–7	1369.262398921897819796541
39	0.230335e–6	1409.213064727081354007227
40	0.427825e–7	1442.704474799536439150649

$$\lambda(N_1) - \lambda(N_2) \sim aN_1^k \exp(-\mu\sqrt{N_1}) - aN_2^k \exp(-\mu\sqrt{N_2}) \sim aN_1^k \exp(-\mu\sqrt{N_1}) \quad (27)$$

since, if N, N_2 are sufficiently large, the higher resolution error is exponentially small compared to the lower resolution error.

The cautious approach to ensuring that the error is less than some specified tolerance τ is to continue increasing the resolution until the eigenvalue separation is less than τ . A less cautious approach is to estimate error by asserting that if the square root of the number of basis functions is doubled, then the logarithm of the error is doubled and therefore the error in the higher resolution computation is roughly the square of the separation between the high and low resolution computations. However, the actual high-resolution error is the square of the low resolution error multiplied by $(2/N)^k$. If $k = -3$, which is equivalent to a convergence-limiting singularity of the form $(x-a)^2 \log(x-a)$ [Chapter 2 of [6]], then squaring the separation underestimates the error by $N^3/8$, which will be huge if $N \gg 1$. Usually the convergence-limiting singularities are complex-valued in location and their type is unknown, so the “separation-squaring” estimate may sometimes be very inaccurate.

Table 5 shows that the Chebyshev/Fourier/radial stretch scheme in thirty-two digit arithmetic allows computation of the ground state eigenvalue to at least twenty correct digits.

Fig. 15 shows that the RBF method for this problem is greatly inferior to the mapped tensor product method. Unfortunately, coordinate mapping has its own limitations; for more complicated domains, the mapping may be singular or nearly-singular. The RBF and polynomial hyperinterpolation methods described here and mapped tensor product methods

are options that expand the range of single domain, exponentially accurate algorithms. None is a universal cure for a complicated geometry.

7. Summary

The Stone–Weierstrass Theorem asserts that a smooth function can always be approximated by a polynomial even in a convoluted domain. We have shown that for a squircle-bounded domain, least squares hyperinterpolation with $P \geq 2N$ yields a geometrically convergent approximation when $f(x, y)$ is smooth. Gaussian radial basis functions (RBFs) do not require hyperinterpolation, which would seem a large plus. However, RBFs need about as many points as polynomials do.

Even in one dimension, hyperinterpolation cannot guarantee convergence on a *uniform* grid unless the number of interpolation points P is $O(N^2)$ [29,11], so we did not apply a uniform grid with polynomials here (but see Part II [23]). If the boundary is defined implicitly as the zero contour of a bivariate function $B(x, y)$, we show that it is easy to use non-zero, interior contours of B to cluster grid points near the boundary in imitation of the usual Chebyshev grid. This Chebyshev-like squircle grid proved quite effective, but a more complicated strategy is needed for the quadrifolium domain with $\delta = 1/1000$. The methodology is *general*, however, and can be applied frequently when the boundary is defined by a function $B(x, y)$.

A much simpler construction is the “truncated uniform” grid. This is built by first embedding the domain in a square, constructing a uniformly-spaced, tensor product grid for the square, and then deleting all points that lie outside the domain Ω . If the sign of the boundary function is normalized so that B is negative on the interior, it is then trivial to test the inclusion of a point merely by examining the sign of B at that point.

Gaussian RBFs lived up to their reputation for grid tolerance by working fairly well on the truncated uniform domain even in the absence of boundary points. We show, however, that adding boundary points is a cost-effective option, and renders the errors more uniform in space. RBFs did not perform well on the Helmholtz eigenproblem.

Polynomials and RBFs required about the same number of interpolation conditions P to achieve a given accuracy, but RBFs (classic interpolation) thus need twice as many basis functions as the comparably-accurate $P \sim 2N$ polynomial hyperinterpolation. For a fixed number of basis functions, hyperinterpolation ($P > N$) produced little improvement for RBFs. Hyperinterpolation is therefore not recommended for RBFs, but proved essential for bivariate polynomials on the squircle.

The squircle-bounded domain and the perturbed quadrifolium domain are both invariant to the symmetry operations of the eight-element dihedral group D_4 : the domains are invariant with respect to reflection about the x -axis, the y -axis and the diagonal line $y = x$ and all products of these reflections. We showed in our previous article [24] (exploiting ideas explained earlier by Bossavit [4,5] and others he cites) that an arbitrary function can be decomposed into six parts. The interpolation problem can then be reduced from an $N \times N$ matrix equation to four matrix problems of size $N/8$ and two of size $N/4$ at a cost which is $3/128$ that of the original problem, assuming use of Gaussian elimination with costs proportional to N^3 . (Note that the two larger subproblems share an identical interpolation matrix, which only needs to be computed and factored once.) For simplicity, we here focused on the completely symmetric problem, but everything we have done can be applied equally well to the other five subproblems.

The squircle is simple because both an explicit parameterization and an implicit specification (through the zero contour of the polynomial $B(x, y, v)$) are available. Often, though, a boundary is defined only by a set of points. We are developing an “isogeometric” approach in which both the boundary and the interpolant/PDE solution are RBF series, but this is for the future.

Another frontier is to apply RBFs and polynomial interpolation to geometries with holes, complicated curvature and so on. With due respect to the non-constructive theory of Weierstrass and Stone, it seems likely that very complicated domains will “break” RBFs and polynomial interpolation, unless memory and runtime requirements beyond the capabilities of a desktop machine are deemed acceptable. However, we are a long way from knowing what these limits are.

Acknowledgements

This work was supported by the National Science Foundation through grant OCE 1059703. Shan Li was supported by the Huijiang Foundation of China (B14005) and National Natural Science Foundation of China (11471215). We thank Jianping Xiao for his careful proofreading. We thank the two reviewers for their detailed comments.

References

- [1] B. Adcock, D. Huybrechts, On the resolution power of Fourier extensions for oscillatory functions, *J. Comput. Appl. Math.* 260 (2014) 312–336.
- [2] M. Al-Lawatia, Solution of advection diffusion equations in two space dimensions by a rational Eulerian Lagrangian localized adjoint method over hexagonal grids, *Int. J. Numer. Anal. Model.* 9 (2012) 43–55.
- [3] J. Allen, N. Kundtz, D.A. Roberts, S.A. Cummer, D.R. Smith, Electromagnetic source transformations using superellipse equations, *Appl. Phys. Lett.* 94 (2009) 194101-1–194101-3.
- [4] A. Bossavit, Symmetry, groups and boundary value problems. A progressive introduction to noncommutative harmonic analyses of partial differential equations in domain with geometrical symmetry, *Comput. Methods Appl. Mech. Eng.* 56 (1986) 167–215.
- [5] A. Bossavit, Boundary value problems with symmetry and their approximation by finite elements, *SIAM J. Appl. Math.* 53 (1993) 1352–1380.
- [6] J.P. Boyd, *Chebyshev and Fourier Spectral Methods*, 2nd edn., Dover, Mineola, New York, 2001, 665 pp.
- [7] J.P. Boyd, A comparison of numerical algorithms for Fourier extension of the first, second and third kinds, *J. Comput. Phys.* 178 (2002) 118–160.

- [8] J.P. Boyd, Limited-area Fourier spectral models and data analysis schemes: windows, Fourier extension, Davies relaxation and all that, *Mon. Weather Rev.* 133 (2005) 2030–2042.
- [9] J.P. Boyd, Large-degree asymptotics and exponential asymptotics for Fourier coefficients and transforms, Chebyshev and other spectral coefficients, *J. Eng. Math.* 63 (2009) 355–399.
- [10] J.P. Boyd, Six strategies for defeating the Runge phenomenon in Gaussian radial basis functions on a finite interval, *Comput. Math. Appl.* 62 (2010) 3108–3122.
- [11] J.P. Boyd, F. Xu, Divergence (Runge phenomenon) for least-squares polynomial approximation on an equispaced grid and Mock–Chebyshev subset interpolation, *Appl. Math. Comput.* 210 (2009) 158–168.
- [12] M.D. Buhmann, *Radial Basis Functions: Theory and Implementations*, Cambridge Monographs on Applied and Computational Mathematics, vol. 12, Cambridge University Press, 2003.
- [13] C.F. Dunkl, Orthogonal polynomials on the hexagon, *SIAM J. Appl. Math.* 47 (1987) 343–351.
- [14] G.E. Fasshauer, M.J. McCourt, Stable evaluation of Gaussian basis function interpolants, *SIAM J. Sci. Comput.* 34 (2012) A737–A762.
- [15] A.I. Fedoseyev, M.J. Friedman, E.J. Kansa, Improved multiquadric method for elliptic partial differential equations via PDE collocation on the boundary, *Comput. Math.* 43 (2002) 439–455.
- [16] B. Fornberg, T.A. Driscoll, G. Wright, R. Charles, Observations on the behavior of radial basis function approximations near boundaries, *Comput. Math. Appl.* 43 (2002) 473–490.
- [17] B. Fornberg, N. Flyer, J. Russell, Comparisons between pseudospectral and radial basis function derivative approximations, *IMA J. Numer. Anal.* 30 (2010) 149–172.
- [18] B. Fornberg, C. Piret, A stable algorithm for flat radial basis functions on a sphere, *SIAM J. Sci. Comput.* 30 (2007) 60–80.
- [19] M. Gardner, The superellipse: a curve that lies between the ellipse and rectangle, *Sci. Am.* 21 (1965) 222–234.
- [20] N.T. Gridgeman, Lamé ovals, *Math. Gaz.* 54 (1970) 31–37.
- [21] D. Huybrechs, Stable high-order quadrature rules with equidistant points, *J. Comput. Appl. Math.* 231 (2010) 933–947.
- [22] G. Lamé, *Examen de différentes méthodes employées pour résoudre les problèmes de géométrie*, M. V. Courcier imprimeur Libraire, 1818.
- [23] S. Li, J.P. Boyd, Spectral methods in non-tensor geometry, Part II: Chebyshev versus Zernike polynomials, gridding strategies and spectral extension on squircle-bounded and perturbed-quadrifolium domains, *Appl. Math. Comput.* (2013), submitted for publication.
- [24] S. Li, J.P. Boyd, Symmetrizing grids, radial basis functions, and Chebyshev and Zernike polynomials for the d_4 symmetry group; interpolation within a squircle, Part I, *J. Comput. Phys.* 258 (2013) 931–947.
- [25] K.M. Liew, S. Kitipornchai, C.W. Lim, Free vibration analysis of thick superelliptical plates, *ASCE J. Eng. Mech.* 124 (1998) 137–145.
- [26] R.L. McCrory, S.A. Orszag, Spectral methods for multi-dimensional diffusion problems, *J. Comput. Phys.* 37 (1980) 93–112.
- [27] G.-m. Dai, V.N. Mahajan, Nonrecursive determination of orthonormal polynomials with matrix formulation, *Opt. Lett.* 32 (2007) 74–76.
- [28] S.A. Orszag, Spectral methods for problems in complex geometries, *J. Comput. Phys.* 37 (1980) 70–92.
- [29] E.A. Rakhmanov, Bounds for polynomials with a unit discrete norm, *Ann. Math.* 165 (2006) 55–88.
- [30] S.A. Sarra, Radial basis function approximation methods with extended precision floating point arithmetic, *Eng. Anal. Bound. Elem.* 35 (2011) 68–76.
- [31] R. Schaback, H. Wendland, Kernel techniques: from machine learning to meshless methods, *Acta Numer.* 15 (2006) 543–639.
- [32] T.W. Sederberg, D.C. Anderson, R.N. Goldman, Implicit representation of parametric curves and surfaces, *Comput. Vis. Graph. Image Process.* 28 (1984) 72–84.
- [33] T.W. Sederberg, R.N. Goldman, Algebraic geometry for computer-aided geometric design, *IEEE Comput. Graph. Appl.* 6 (1986) 52–59.
- [34] I.H. Sloan, Polynomial interpolation and hyperinterpolation over general regions, *J. Approx. Theory* 83 (1995) 238–254.
- [35] L.N. Trefethen, *Approximation Theory and Approximation Practice*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 2013.
- [36] H. Wendland, *Scattered Data Approximation*, Cambridge University Press, 2005.
- [37] X. Zhang, P.L. Rosin, Superellipse fitting to partial data, *Pattern Recognit.* 36 (2003) 743–752.