

A Numerical Study of some Radial Basis Function based Solution Methods for Elliptic PDEs

Elisabeth Larsson*

Department of Scientific Computing
Information Technology, Uppsala University
Box 337, SE-751 05 Uppsala, Sweden (bette@tdb.uu.se)

Bengt Fornberg**

Department of Applied Mathematics
University of Colorado
526 UCB, Boulder, CO 80309, USA (fornberg@colorado.edu)

Abstract

During the last decade, three main variations have been proposed for solving elliptic PDEs by means of collocation with radial basis functions (RBFs). In this study, we have implemented them for infinitely smooth RBFs, and then compared them across the full range of values for the shape parameter of the RBFs. This was made possible by a recently discovered numerical procedure that bypasses the ill-conditioning, which has previously limited the range that could be used for this parameter. We find that the best values for it often fall outside the range that was previously available. We have also looked at piecewise smooth versus infinitely smooth RBFs, and found that for PDE applications with smooth solutions, the infinitely smooth RBFs are preferable, mainly because they lead to higher accuracy. In a comparison of RBF-based methods against two standard techniques (a second-order finite difference method and a pseudospectral method), the former gave a much superior accuracy.

Keywords Radial basis functions, RBF, Poisson's equation, collocation

1 Introduction

The ideal numerical method for PDE problems should be high-order accurate, flexible with respect to the geometry, computationally efficient, and easy to implement. The methods that are commonly used, usually fulfill one or two of the criteria, but not all. Finite difference methods can be made high-order accurate, but require a structured grid (or a collection of structured grids). Spectral methods are even more accurate, but have severe restrictions

*The work was supported by a postdoctoral grant from STINT, The Swedish Foundation for International Cooperation in Research and Higher Education.

**The work was supported by NSF grants DMS-9810751 (VIGRE), DMS-0073048, and a Faculty Fellowship from University of Colorado at Boulder.

on the geometry and, in the Fourier case, also require periodic boundary conditions. Finite element methods are highly flexible, but it is hard to achieve high-order accuracy, and both coding and mesh generation become increasingly difficult when the number of space dimensions increases.

A fairly new approach to solving PDEs is through radial basis functions (RBFs). An RBF depends only on the distance to a center point \underline{x}_j and is of the form $\phi(\|\underline{x} - \underline{x}_j\|)$. The RBF may also have a shape parameter ε , in which case $\phi(r)$ is replaced with $\phi(r, \varepsilon)$. Some of the most popular RBFs are given in Table 1.

Table 1: Some commonly used radial basis functions.

Piecewise smooth RBFs	$\phi(r)$
Piecewise polynomial (R_n)	$ r ^n, \quad n \text{ odd}$
Thin Plate Spline (TPS_n)	$ r ^n \ln r , \quad n \text{ even}$
Infinitely smooth RBFs	$\phi(r, \varepsilon)$
Multiquadric (MQ)	$\sqrt{1 + (\varepsilon r)^2}$
Inverse multiquadric (IMQ)	$\frac{1}{\sqrt{1 + (\varepsilon r)^2}}$
Inverse quadratic (IQ)	$\frac{1}{1 + (\varepsilon r)^2}$
Gaussian (GS)	$e^{-(\varepsilon r)^2}$

A key feature of an RBF method is that it does not require a grid. The only geometric properties that are used in an RBF approximation are the pairwise distances between points. Distances are easy to compute in any number of space dimensions, so working in higher dimensions does not increase the difficulty. The method works with points scattered throughout the domain of interest, and the RBF interpolant is a linear combination of RBFs centered at the scattered points \underline{x}_j ,

$$s(\underline{x}, \varepsilon) = \sum_{j=1}^N \lambda_j \phi(\|\underline{x} - \underline{x}_j\|, \varepsilon),$$

where the coefficients λ_j are usually determined by collocation with given discrete data, such as function values or derivative information. When infinitely smooth RBFs are used, the approximations feature spectral convergence as the points get denser. This has been proven strictly only for some special cases [1, 2], although numerical evidence strongly suggests that it is true in much more general settings. Furthermore, implementation of an RBF method is straightforward. However, there are some remaining issues such as computational efficiency, and stability if applied to time-dependent problems without viscosity.

For elliptic problems, there are three main categories of RBF based methods. The first category only uses the RBFs to find a particular solution of the inhomogeneous PDE. Then homogeneous fundamental solutions are added in such a way that the boundary conditions are met [3]. We are going to focus on the second category, which consists of pure collocation methods. In these, both the PDE and the boundary conditions are satisfied by collocation. We are going to investigate three slightly differing variations that were developed during the last decade [4, 5, 6, 7, 8]. We will see that the accuracy of the solutions is a function of

the shape parameter, and that very small values of ε often give the best results. A problem that has been common to these RBF methods is the severe ill-conditioning that occur for small values of ε . This has, until recently, made it impossible to use more than a limited range of ε -values. In this paper, we are going to use a novel numerical approach [9] that largely overcomes the numerical ill-conditioning as $\varepsilon \rightarrow 0$. We are going to compare the three collocation methods for solving Poisson's equation across the full range $[0, \infty]$ of the parameter ε .

A third category of collocation methods was recently proposed [10]. Instead of introducing an RBF expansion with unknown coefficients for the solution to the PDE, (which is then differentiated and collocated), the inhomogeneous term is interpolated by RBFs, and then integrated and collocated.

The following sections of this paper are structured as follows: In Section 2, we describe the test problem. Section 3, briefly describes the three collocation methods from the second category above. The new Cauchy integral approach that overcomes the ill-conditioning is summarized very briefly in Section 4. Section 5 contains a variety of numerical comparisons. While the performance of the three different collocation methods proved comparable, we observe significant differences in accuracy between different types of RBFs, and different values of ε . We also compare the accuracy of the RBF method against that of a second-order finite difference method and that of a Fourier–Chebyshev pseudospectral method. Section 6 contains some concluding remarks.

2 The elliptic model problem

Let $\Omega \subset \mathbb{R}^d$ be a d -dimensional domain and let $\partial\Omega$ be the boundary of the domain. We want to solve the following Poisson problem:

$$\begin{aligned} u(\underline{x}) &= g(\underline{x}) \text{ on } \partial\Omega, \\ \Delta u(\underline{x}) &= f(\underline{x}) \text{ in } \Omega. \end{aligned}$$

For collocation, we use node points distributed both along the boundary (\underline{x}_j , $j = 1, \dots, N_B$), and over the interior (\underline{x}_j , $j = N_B + 1, \dots, N_B + N_I = N$).

In the experiments we use cases where the solution $u(x)$ is known and normalized so that $\max_{\Omega} |u(\underline{x})| = 1$. Let the RBF interpolant be denoted by $s(\underline{x}, \varepsilon)$. We measure the error, (which depends on ε), in max norm as

$$E(\varepsilon) = \max_{\Omega} |s(\underline{x}, \varepsilon) - u(\underline{x})|.$$

3 Three collocation methods

In this section, the three collocation methods for elliptic PDEs that we are comparing are presented in the chronological order of their introductions. Note that the basic solution approach is not limited to elliptic problems. Applications to other types of problems are found, e.g., in [5], [11], [12].

Method 1. Straight collocation

A straightforward RBF-based collocation method for elliptic problems was introduced by Kansa, 1990 [4], [5]. Let the RBF approximation to the solution $u(\underline{x})$ be

$$s(\underline{x}, \varepsilon) = \sum_{j=1}^N \lambda_j \phi(\|\underline{x} - \underline{x}_j\|, \varepsilon).$$

Collocation with the boundary data at the boundary points and with the PDE at the interior points leads to the equations

$$\begin{aligned} s(\underline{x}_i, \varepsilon) &\equiv \sum_{j=1}^N \lambda_j \phi(\|\underline{x}_i - \underline{x}_j\|, \varepsilon) = g(\underline{x}_i), & i = 1, \dots, N_B, \\ \Delta s(\underline{x}_i, \varepsilon) &\equiv \sum_{j=1}^N \lambda_j \Delta\phi(\|\underline{x}_i - \underline{x}_j\|, \varepsilon) = f(\underline{x}_i), & i = N_B + 1, \dots, N. \end{aligned}$$

This corresponds to a system of equations with an unsymmetric coefficient matrix, schematically structured as

$$\begin{bmatrix} \phi \\ \Delta\phi \end{bmatrix} \begin{bmatrix} \lambda \end{bmatrix} = \begin{bmatrix} g \\ f \end{bmatrix}.$$

It has been shown (by example) that, in rare cases, the coefficient matrix may become singular [13]. However, practical experience shows that in general the method works well [14].

Method 2. Symmetric collocation

A variation that leads to a symmetric coefficient matrix was derived by Wu, 1992 [6]; see also Fasshauer, 1996 [7]. It has been shown for this method that the symmetry assures a non-singular system of equations [6].

The idea is to modify the basis functions in the interpolant by using the operator in the partial differential equation that is being studied. For each node point, we look at what the operator is and then apply it to the basis function centered in that point. Here, that means that the basis function is just ϕ for boundary points, and $\Delta\phi$ for interior points, leading to

$$s(\underline{x}, \varepsilon) = \sum_{j=1}^{N_B} \lambda_j \phi(\|\underline{x} - \underline{x}_j\|, \varepsilon) + \sum_{j=N_B+1}^N \lambda_j \Delta\phi(\|\underline{x} - \underline{x}_j\|, \varepsilon).$$

Collocation at boundary and interior points yields the equations

$$\begin{aligned} s(\underline{x}_i, \varepsilon) &\equiv \sum_{j=1}^{N_B} \lambda_j \phi(\|\underline{x}_i - \underline{x}_j\|, \varepsilon) + \sum_{j=N_B+1}^N \lambda_j \Delta\phi(\|\underline{x}_i - \underline{x}_j\|, \varepsilon) \\ &= g(\underline{x}_i), \quad i = 1, \dots, N_B, \\ \Delta s(\underline{x}_i, \varepsilon) &\equiv \sum_{j=1}^{N_B} \lambda_j \Delta\phi(\|\underline{x}_i - \underline{x}_j\|, \varepsilon) + \sum_{j=N_B+1}^N \lambda_j \Delta^2\phi(\|\underline{x}_i - \underline{x}_j\|, \varepsilon) \\ &= f(\underline{x}_i), \quad i = N_B + 1, \dots, N. \end{aligned}$$

The block structure of the system of equations becomes

$$\left[\begin{array}{c|c} \phi & \Delta\phi \\ \hline \Delta\phi & \Delta^2\phi \end{array} \right] \begin{bmatrix} \lambda \end{bmatrix} = \begin{bmatrix} g \\ f \end{bmatrix}.$$

Method 3. Direct collocation, using the PDE also on the boundary

As for most interpolation methods, the errors in RBF approximations tend to be largest near boundaries [15]. It therefore makes sense to impose more information there. Fedoseyev, Friedman and Kansa (2000) [8] formulated a method that collocates both with the boundary condition and the PDE at the boundary points.

In order to have a matching number of unknowns and equations, additional expansion functions are added. The centers of these are placed outside the boundary. Let the center points be denoted by \underline{z}_j , where

$$\underline{z}_j = \begin{cases} \underline{x}_j, & j = 1, \dots, N, \\ \text{a point outside } \Omega, & j = N + 1, \dots, N + N_B. \end{cases}$$

The RBF interpolant now takes the form

$$s(\underline{x}, \varepsilon) = \sum_{j=1}^{N+N_B} \lambda_j \phi(\|\underline{x} - \underline{z}_j\|, \varepsilon).$$

The collocation equations are very similar to those of Method 1. The difference lies in the extra collocation at the boundary and the added centers.

$$\begin{aligned} s(\underline{x}_i, \varepsilon) &\equiv \sum_{j=1}^{N+N_B} \lambda_j \phi(\|\underline{x}_i - \underline{z}_j\|, \varepsilon) = g(\underline{x}_i), \quad i = 1, \dots, N_B, \\ \Delta s(\underline{x}_i, \varepsilon) &\equiv \sum_{j=1}^{N+N_B} \lambda_j \Delta\phi(\|\underline{x}_i - \underline{z}_j\|, \varepsilon) = f(\underline{x}_i), \quad i = 1, \dots, N. \end{aligned}$$

The structure of the system of equations is the same as for Method 1, but the sizes of the blocks in the matrix are different.

$$\left[\begin{array}{c} \phi \\ \Delta\phi \end{array} \right] \begin{bmatrix} \lambda \end{bmatrix} = \begin{bmatrix} g \\ f \end{bmatrix}$$

Some notes on the three methods

1. In the original versions of Methods 1 and 3, the ε parameter is varied with the center location. This can improve the condition number of the coefficient matrix and it also allows for adaptivity in the method. We have chosen to use a fixed ε for all the methods in order to simplify comparison.

2. For pure interpolation, adding a polynomial term of a certain degree to the RBF interpolant, and introducing corresponding constraints on the expansion coefficients will in many cases lead to a guaranteed non-singular coefficient matrix [16]. However, for the elliptic problem, the non-singularity cannot be guaranteed when unsymmetric collocation is employed, even if the correct polynomial term is added [13]. We have found that such polynomial terms make little or no difference in the resulting accuracy, so we have not considered them further.
3. For Method 3, it is not necessary to put the extra centers outside Ω , but it does not seem to be especially beneficial to put them inside.

4 A Cauchy integral technique for stable computation at all values of the shape parameter ε

Computing the interpolant for large ε is usually not a problem, since the matrix then has a reasonable condition number. However, for ε decreasing towards zero, the condition number grows rapidly. Recently, Fornberg and Wright developed a method to compute the RBF interpolant for small values of ε [9].

Very briefly, the main ideas are the following: First note that the interpolant $s(\underline{x}, \varepsilon)$, for any fixed \underline{x} , is an analytical function of ε with, in general, $\varepsilon = 0$ as a removable singularity. We can compute $s(\underline{x}, \varepsilon)$ around a circle in the complex ε -plane, where the radius is chosen large enough that the standard method is stable. Then we take the inverse Fourier transform of the values around the circle. Assuming that the function $s(\underline{x}, \varepsilon)$ has no poles inside the circle, this results in the Taylor coefficients in an expansion $s(\underline{x}, \varepsilon) = s_0(\underline{x}) + \varepsilon^2 s_2(\underline{x}) + \varepsilon^4 s_4(\underline{x}) + \dots$. Using this expansion, the interpolant can be computed for any given value of ε inside the circle. If the function $s(\underline{x}, \varepsilon)$ has poles inside the circle, some extra steps are called for in the algorithm. For the complete description, see [9].

5 Numerical results

For the numerical experiments, the computational domain Ω is the unit disk. The following normalized test functions are chosen so that they have quite different optimal values of ε .

$$\begin{aligned}
 u_1 &= \frac{65}{65 + (x - 0.2)^2 + (y + 0.1)^2}, \\
 u_2 &= \frac{25}{25 + (x - 0.2)^2 + 2y^2}, \\
 u_3 &= \exp\left(-\left((x - 0.1)^2 + 0.5y^2\right)\right), \\
 u_4 &= \frac{\exp\left((x - 0.1)^2 + 0.5y^2\right)}{\exp(1.21)}, \\
 u_5 &= \sin\left(\pi(x^2 + y^2)\right), \\
 u_6 &= \frac{\arctan(2(x + 3y - 1))}{\arctan(2(\sqrt{10} + 1))}.
 \end{aligned}$$

The right hand side functions f and g are computed from the known solution u . All computations are performed using double precision (64 bit) arithmetic. When the Cauchy technique is used, the interpolant is evaluated at 256 or 512 points around the circle. However, due to the fourfold symmetry, we only need to perform the computation at 65 or 129 points.

5.1 A comparison of a straightforward and the new implementation

In our first experiment, we compare, for all three methods, the results from using a standard implementation and from using the Cauchy integral method for small values of ε . Figure 1 shows the node distributions used. The distribution for Methods 1 and 2 has 50 node points and centers. There are two distributions for Method 3. The first one makes the size of the coefficient matrix the same as for the other two methods, and the second one has the same number of collocation points (but a larger matrix size). It should be noted that the results are not very sensitive to the choice of node distribution and that the ones we use are not optimized in any way.

Figure 2 shows the results of the computations for the function u_1 using the MQ radial basis function. Considering only the standard implementation, Method 2 seems to be the most accurate by almost two orders of magnitude. The reason for this is that the conditioning of the system for Method 2 is better, so that we can accurately compute the solution for smaller values of ε . The Cauchy version of the implementation suffers no numerical ill-conditioning, and the results that are shown are accurate at all values of ε . A theoretical study of the errors for small values of the shape parameter will be included in [17]. Some results for approximations in the limit $\varepsilon \rightarrow 0$ are also given in [18]. We see a major change in the trend of the error around $\varepsilon = 0.1$. It is by coincidence only that the change in error trend in this example occurs around the same time as the standard implementation fails. The Cauchy results show that the best accuracy is obtained for values of ε in the interval 0.11–0.15, where the standard implementation fails to produce reliable results. Furthermore it shows, that around the optimum, there is very little difference between Methods 1 and 2. Method 3 performs slightly better when the same number of collocation points are used, but that is not a fair comparison since the amount of work is larger with the larger matrix size. Method 3 with the same matrix size performs worse than the other two method in this case.

5.2 A more thorough comparison of the three methods

Here we are going to use the Cauchy version of the implementation. We use the same node distributions, the MQ RBF, and all the test functions. The optimal ε and errors are shown in Table 2. Again, Methods 1 and 2 give very similar results, with Method 2 marginally better. Method 3 is in most cases worse with the fair comparison, but slightly better with the same number of collocation points.

5.3 A comparison between different basis functions

So far, all experiments have been performed with the infinitely smooth MQ RBF. Next, we will try to see if the interrelationship between the methods change if we use piecewise smooth (shape parameter free) basis functions, and we will also compare MQ with other infinitely

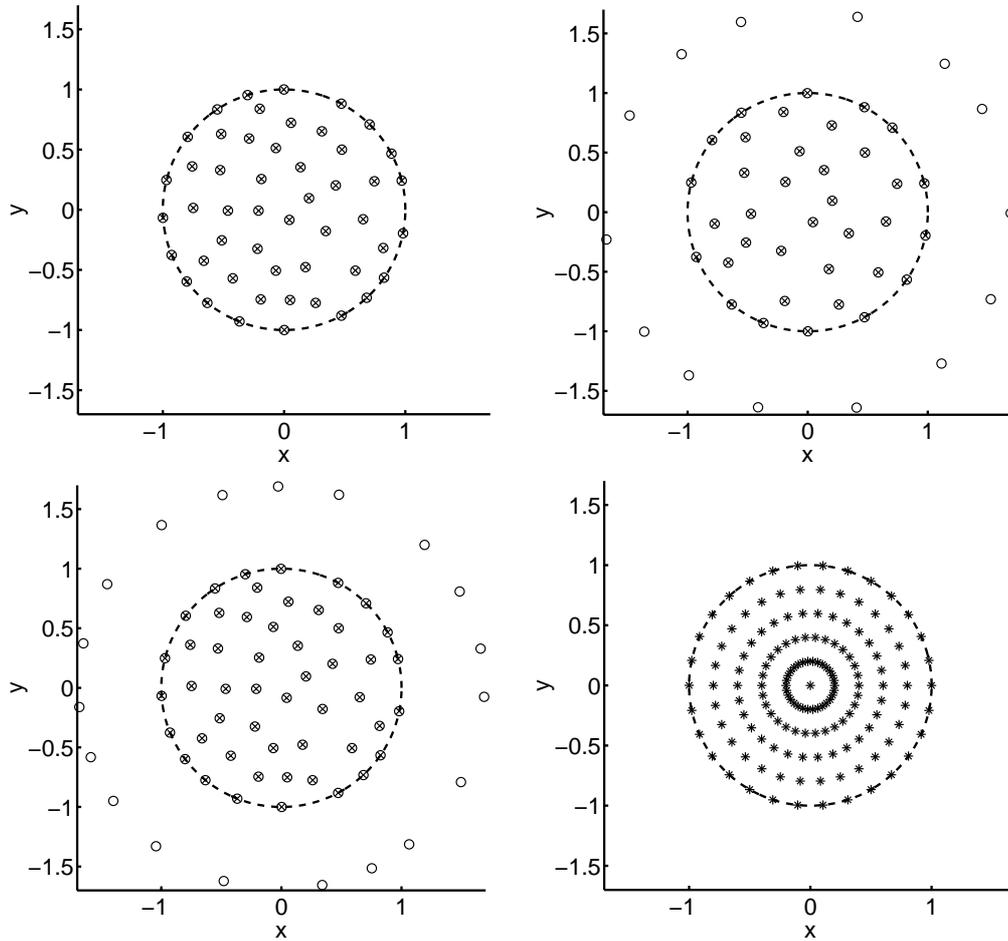


Figure 1: The distribution of nodes (x) and centers (o) for Methods 1 and 2 (top left), and Method 3 (top right), with a total of 50 centers. The distribution of nodes and centers for Method 3 with a total of 50 nodes (bottom left). The points at which the error is evaluated (bottom right).

smooth RBFs. This time we will only include results for Method 3 that corresponds to equally sized matrices.

For piecewise polynomial (spline) interpolation, accuracy can be severely degraded close to the boundary unless a suitable boundary condition is imposed. In [15] the counterpart for RBFs of the Not-a-Knot (NaK) condition in spline interpolation is introduced. This condition is applied by moving the centers that are closest to the boundary, but not on the boundary, to the outside of the domain Ω . (Note that only the centers, not the node points, are moved.) Since the piecewise smooth RBFs are infinitely smooth everywhere except at the center point, this enforces smoothness of the interpolant in the region just inside the

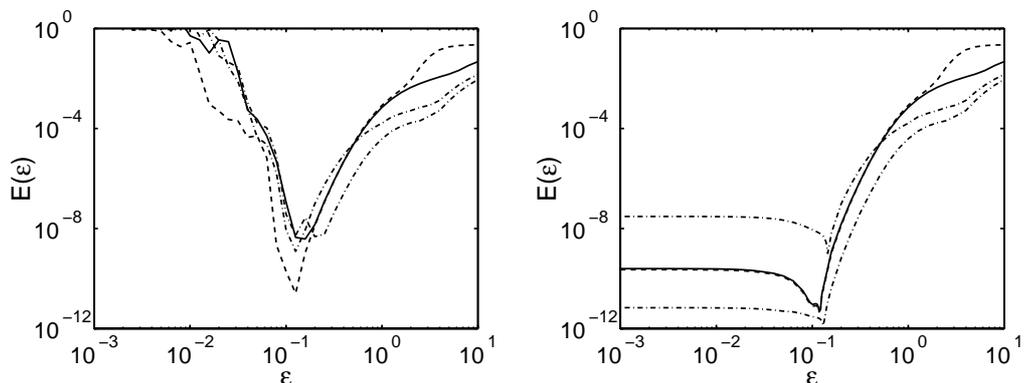


Figure 2: The errors with the standard implementation (left) and the Cauchy implementation (right) using the test function u_1 . Results for Method 1 (solid line), Method 2 (dashed line), and Method 3 (dash-dot lines).

Table 2: Optimal ε and errors for the three methods for different test functions. The bullets show the best result of the three with the same matrix size. The second column for Method 3 shows the result with the same number of collocation points for comparison.

	Method 1		Method 2		Method 3			
	ε^*	$E(\varepsilon^*)$	ε^*	$E(\varepsilon^*)$	ε^*	$E(\varepsilon^*)$	ε^*	$E(\varepsilon^*)$
u_1	0.12	5.0e-12	0.12	●4.7e-12	0.15	1.0e-9	0.13	1.2e-12
u_2	0.24	6.4e-9	0.23	●4.6e-9	0.25	5.3e-8	0.27	8.5e-9
u_3	0.37	9.7e-6	0.36	●8.3e-6	0.40	2.2e-5	0.30	3.7e-7
u_4	0	5.5e-4	0	●5.0e-4	0.31	1.7e-2	0	9.8e-5
u_5	0	●2.8e-2	0	3.3e-2	0.79	2.2e-1	0.41	1.9e-2
u_6	0.89	2.3e-1	0.79	2.3e-1	3.20	●1.7e-1	1.77	2.6e-1

boundary. Fornberg et al. [15] also went one step further to the “super NaK” condition, where the centers on the boundary as well as the first layer of centers inside the boundary are moved out of the domain. Here, we use an even more general version of the condition, where any number of centers can be moved outside (for examples, see Figure 3).

For the numerical experiments, we first solve the problem using the distributions from Figure 1. Then we modify the distributions by moving out centers and solve the problem again. For each combination of method, RBF, and test function, we pick the distribution that gives the best result. As a rule of thumb, the number of centers that are moved out needs to increase with the power of $|r|$ in the RBF. Table 3 shows the numerical results for each method. The first column is for the standard version and the second column shows the result when the generalized NaK boundary condition is used. Note that using the boundary condition with Method 2 actually defeats the purpose of the method, which was to get a symmetric matrix. Still, it improves the result, so we include the experiment anyway. Also note that Method 2 uses higher derivatives of the basis function. Therefore, R_3 and TPS_4 cannot be used with it. Basis functions with even lower powers of $|r|$, such as TPS_2 and R_1 ,

Table 3: For each method, the first column shows the error for the standard version and the second column shows the error when the generalized Not-a-Knot boundary condition is added. The bullets show the best result in each row. For the definition of the different RBFs, see Table 1.

	Method 1		Method 2		Method 3	
ϕ	u_1					
R ₃	1.8e-2	5.5e-4	–	–	2.5e-3	• 1.4e-4
R ₅	2.1e-2	• 7.0e-5	4.0e-2	9.3e-5	1.4e-3	1.6e-4
R ₇	6.3e-2	1.6e-4	7.5e-2	• 5.0e-5	4.5e-3	1.6e-4
TPS ₄	6.7e-2	6.6e-5	–	–	4.5e-3	• 5.7e-5
TPS ₆	1.8e-1	• 5.4e-5	4.5e-1	5.7e-5	1.0e-2	2.4e-4
ϕ	u_3					
R ₃	2.8e-2	• 6.9e-3	–	–	1.3e-2	1.3e-2
R ₅	2.6e-2	1.7e-3	4.8e-2	3.5e-3	3.0e-3	• 1.4e-3
R ₇	7.0e-2	• 5.2e-4	8.4e-2	1.1e-3	5.6e-3	1.6e-3
TPS ₄	7.0e-2	3.4e-3	–	–	7.8e-3	• 2.7e-3
TPS ₆	1.8e-1	• 9.0e-4	4.8e-1	3.6e-3	1.1e-2	2.4e-3

can not be used with any of the methods.

The table clearly shows that Method 3 improves the accuracy for piecewise continuous RBFs, when no boundary enhancement is used. Adding the generalized NaK condition to any of the three methods gives a significant improvement. However, with the boundary enhancement, none of the methods are significantly better than the others. Also, there are no big differences between the different basis functions. With larger numbers of node points, we would expect to see improvements as the degree of the RBF is increased to a certain extent. When the degree gets too high, the beneficial effect is counteracted by increasing boundary errors.

An example of what the node distributions looks like with the generalized NaK condition is shown in Figure 3. For these problems, where the exact solutions are infinitely smooth, the best distributions are when all or almost all centers are moved to the outside to increase the smoothness of the interpolant. The optimal choice of distribution depends both on the solution and the basis function.

This is in sharp contrast to the case of infinitely smooth RBFs, where extra boundary conditions have little or no effect on the minimum value of the error, and there is little to be gained by moving the centers around.

Figure 4 shows how MQ, IMQ, IQ, and GS RBFs perform compared with each other using Method 2 and the node distribution from Figure 1. The best result for a piecewise smooth RBF is also included.

For very smooth functions (like u_1), the infinitely smooth RBFs very clearly outperform piecewise smooth ones (as to be expected due to their spectral rather than algebraic accuracy). For functions with more local variations (like u_3), this will again be the case as the resolution is increased but, when the sampling is relatively coarse, all the methods become more similar in accuracy. In all cases, the difference in accuracy between the four smooth RBFs methods is seen to be rather minor.

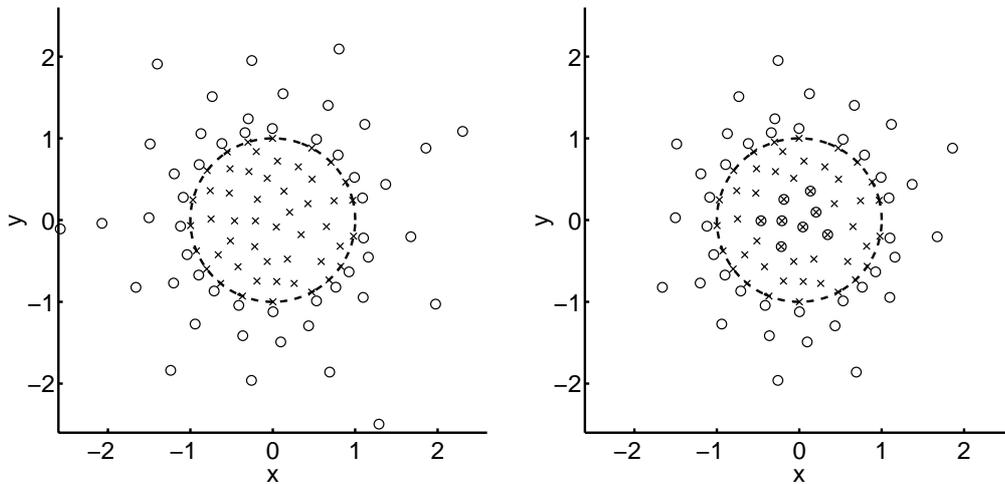


Figure 3: An optimized Not-a-Knot node distribution for Method 1 and $\phi = |r|^5$ for test functions u_1 (left) and u_3 (right).

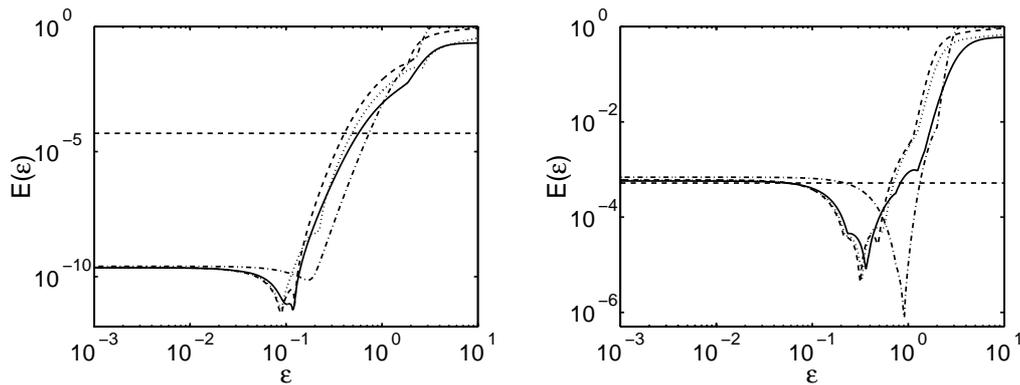


Figure 4: The errors for u_1 (left) and u_3 (right) using MQ (solid line), IMQ (dotted line), IQ (dashed line), and GS (dash-dot line). The best result for a piecewise smooth RBF is included for comparison.

5.4 A comparison with standard methods

With the unit disk as the computational domain, many standard numerical methods can be applied. We next show a comparison with a second-order finite difference method and a Fourier–Chebyshev pseudospectral method. For the pseudospectral method we use the procedure described in [19, Chapter 11]. For the comparison, we require this time that all methods use the same number of boundary, as well as interior, node points (16 and 32 respectively). The node distributions are shown in Figure 5.

The computational costs for the methods are of course different, even though the num-

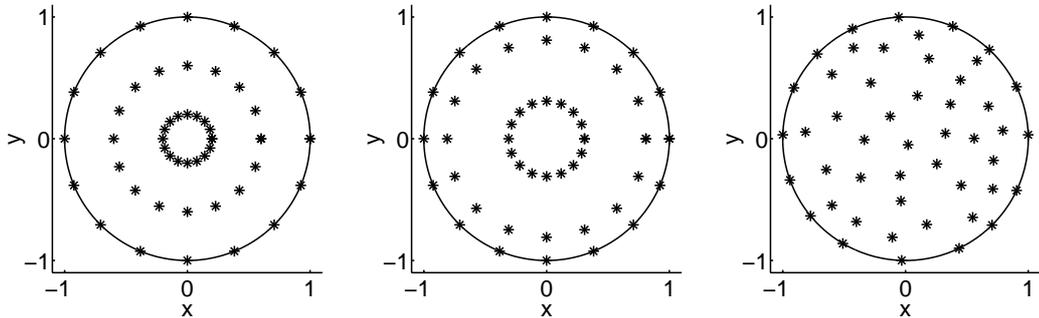


Figure 5: The distributions for the second order finite difference, pseudospectral, and RBF methods. All have 16 points on the boundary and 32 in the interior.

ber of nodes are the same. However, our main concern here is to find out what level of accuracy we can expect to obtain with an RBF method. For a problem on the unit disk, the pseudospectral approach would likely be the most efficient, but for an irregular geometry, only the RBF method is a viable alternative. The computational efficiency issue, and ways of dealing with large point sets have been investigated by other authors, see e.g., [20].

The results from the computations are shown in Figure 6. For both test functions, the RBF method is superior to the standard methods with respect to accuracy, no matter which smooth RBF is used. A great advantage with the RBF method is the fact that node points can be placed anywhere. Accordingly, we can use a fairly uniform distribution of the points over the domain, whereas for the standard methods, there has to be a certain amount of clustering.

6 Concluding remarks

By using the new Cauchy integral method we were able to explore the whole range of the shape parameter ε for infinitely smooth RBFs applied to elliptic PDEs. We have found that, in many cases, the most accurate results are achieved for values of ε that are inaccessible to the standard method of computing RBF approximations. We also note that the range of ε that can be reached with the standard method decreases when the number of node points increases (because this requires the solution of systems of equations with increasingly ill-conditioned coefficient matrices).

In this study, we have compared three different RBF methods from the literature and used them both with infinitely smooth and with piecewise smooth RBFs. For the infinitely smooth RBFs, Method 2 (symmetric collocation) is marginally better than Method 1 (straight collocation), whereas Method 3 (extra boundary collocation) usually is less effective.

The situation is the opposite for piecewise smooth RBFs. In that context, Method 3 is clearly better than Methods 1 and 2. However, when introducing the generalized Not-a-Knot boundary enhancement, all three methods are improved and perform very similarly. Method 2 is the least effective of the three with the piecewise smooth RBFs, because it requires more regularity from the RBFs, and the symmetry (which is the advantage of the

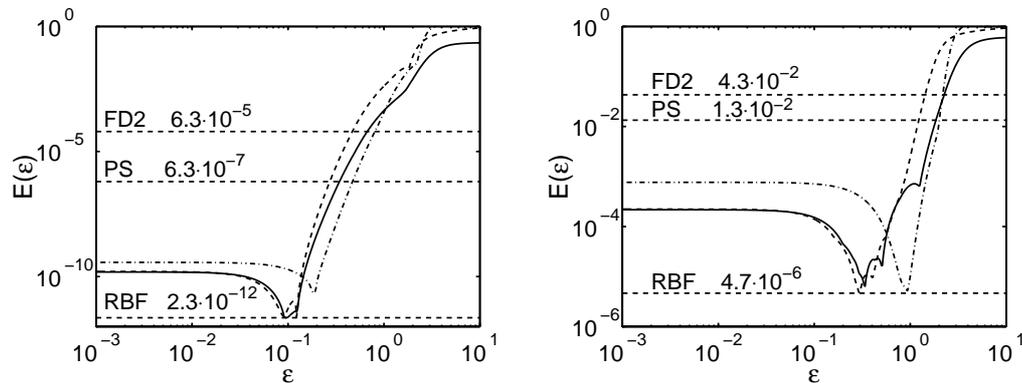


Figure 6: The errors as a function of ε using Method 2 with MQ (solid line), IQ (dashed line), and GS (dash-dot line) RBFs for test functions u_1 (left) and u_3 (right). The errors for the standard second-order finite difference method (FD2) and the Fourier–Chebyshev pseudospectral (PS) method are also included.

method) is destroyed when the Not-a-Knot condition is introduced.

For the type of PDE application that we have studied here, with smooth solutions, the infinitely smooth RBFs are preferable to the piecewise smooth ones. This is not only because they lead to significantly more accurate solutions, but also because they eliminate the need to optimize the node distribution. Of course there is the question of which ε to choose, but that is more of a possibility for improvement than an obstacle. While some specific value gives the best performance, a wide range of low ε -values give excellent accuracy.

When comparing the RBF method with standard methods (FD2 and PS) for a case where the geometry is simple enough to allow the latter to be implemented effectively, the RBF method still performed significantly better. This can be ascribed to the flexibility of the RBF methods, which all allow nodes to be distributed for a fairly uniform coverage, (in any geometry) and, of course, to their spectral accuracy.

References

1. M. Buhmann and N. Dyn, Spectral convergence of multiquadric interpolation. *Proc. Edinburgh Math. Soc. (2)* **36** (2) 319–333 (1993).
2. W. R. Madych and S. A. Nelson, Bounds on multivariate polynomials and exponential error estimates for multiquadric interpolation. *J. Approx. Theory* **70** (1) 94–114 (1992).
3. M. A. Golberg and C. S. Chen, *Discrete Projection Methods for Integral Equations*, Computational Mechanics Publications, Southampton (1997).
4. E. J. Kansa, Multiquadrics—A scattered data approximation scheme with applications to computational fluid-dynamics. I. Surface approximations and partial derivative estimates. *Comput. Math. Appl.* **19** (8/9) 127–145 (1990).

5. E. J. Kansa, Multiquadrics—A scattered data approximation scheme with applications to computational fluid-dynamics. II. Solutions to parabolic, hyperbolic and elliptic partial differential equations. *Comput. Math. Appl.* **19** (8/9) 147–161 (1990).
6. Z. M. Wu, Hermite-Birkhoff interpolation of scattered data by radial basis functions. *Approx. Theory Appl.* **8** (2) 1–10 (1992).
7. G. E. Fasshauer, Solving partial differential equations by collocation with radial basis functions, in *Surface fitting and multiresolution methods, Vol. 2 of the Proceedings of the 3rd International Conference on Curves and Surfaces held in Chamonix–Mont-Blanc, 1996* (Edited by A. Le Méhauté, C. Rabut, and L. L. Schumaker), vol. 2, pp. 131–138, Vanderbilt University Press, Tennessee (1997).
8. A. I. Fedoseyev, M. J. Friedman, and E. J. Kansa, Improved multiquadric method for elliptic partial differential equations via PDE collocation on the boundary. *Comput. Math. Appl.* (2001). To appear.
9. B. Fornberg and G. Wright, Stable computation of multiquadric interpolants for all values of the shape parameter. Submitted to *BIT* (2002).
10. N. Mai-Duy and T. Tran-Cong, Numerical solution of differential equations using multiquadric radial basis function networks. *Neural Networks* **14** (2) 185–199 (2001).
11. Y. C. Hon and X. Z. Mao, An efficient numerical scheme for Burgers’ equation. *Appl. Math. Comput.* **95** (1) 37–50 (1998).
12. Y. C. Hon, K. F. Cheung, X. Z. Mao, and E. J. Kansa, Multiquadric solution for shallow water equations. *ASCE J. Hydr. Engrg.* **125** (5) 524–533 (1999).
13. Y. C. Hon and R. Schaback, On unsymmetric collocation by radial basis functions. *Appl. Math. Comput.* **119** (2–3) 177–186 (2001).
14. E. J. Kansa and Y. C. Hon, Circumventing the ill-conditioning problem with multiquadric radial basis functions: applications to elliptic partial differential equations. *Comput. Math. Appl.* **39** (7–8) 123–137 (2000).
15. B. Fornberg, T. A. Driscoll, G. Wright, and R. Charles, Observations on the behaviour of radial basis function approximations near boundaries. *Comput. Math. Appl.* (2002). To appear.
16. C. A. Micchelli, Interpolation of scattered data: distance matrices and conditionally positive definite functions. *Constr. Approx.* **2** (1) 11–22 (1986).
17. E. Larsson and B. Fornberg, Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions. To be submitted.
18. T. A. Driscoll and B. Fornberg, Interpolation in the limit of increasingly flat radial basis functions. *Comput. Math. Appl.* (2002). To appear.
19. L. N. Trefethen, *Spectral methods in MATLAB*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (2000).

20. R. K. Beatson, W. A. Light, and S. Billings, Fast solution of the radial basis function interpolation equations: domain decomposition methods. *SIAM J. Sci. Comput.* **22** (5) 1717–1740 (electronic) (2000).