SHAPE PARAMETER ESTIMATION IN RBF FUNCTION APPROXIMATION

A. KARAGEORGHIS[1] & P. TRYFONOS[1]
[1] Department of Mathematics and Statistics, University of Cyprus, Cyprus.

ABSTRACT

The radial basis function (RBF) collocation method is applied for the approximation of functions in two variables. When the RBFs employed include a shape parameter, the determination of an appropriate value for it is a major issue. In this work, this is addressed by including the value of the shape parameter in the unknowns along with the coefficients of the RBFs in the approximation. The variable shape parameter case when a different shape parameter is associated with each RBF in the approximation is also considered. Both approaches yield nonlinear systems of equations, which are solved by a standard non-linear solver. The results of several numerical experiments are presented.

Keywords: collocation, function approximation, radial basis functions.

1 INTRODUCTION

One of the major issues in the approximation of functions by linear combinations of radial basis functions (RBFs), which includes a shape parameter [2, 3] is the determination of an optimal, or at least an acceptable, value of this shape parameter and to this end several techniques have been proposed in the past [4, 8, 10, 18].

In this work the problem of approximating functions in two-dimensional domains using an RBF collocation method is considered and an attempt to address the issue of estimating an appropriate value of the RBF shape parameter is made. The basic idea was introduced in [12, 13] where the Kansa-RBF method [14] was applied to two-dimensional second and fourth order nonlinear boundary value problems. The shape parameter determination issue is still present in such applications and to overcome it, in [12, 13] it (the shape parameter) was taken to be one of the unknowns to be determined as part of the solution. Since the problems in question were nonlinear anyway, this inclusion did not affect the discretization of the problem much as this added nonlinearity was absorbed, in a natural way, by the resulting nonlinear system. In the current work, the same idea is employed in function approximation and the collocation of the approximation at various points in the domain now yields a system of nonlinear equations, which is solved using standard software. Thus, as well as the unknown coefficients in the RBF expansion, a value for the shape parameter is obtained. Here, the results of [15] are extended, see also [20], and, in addition, the case where a different value of the shape parameter is used for each RBF in the function approximation, is considered. This idea of using variable shape parameters in RBF approximations is not new and has been previously used in [1, 11, 19, 22].

The paper is organized as follows. In Section 2, the RBF formulation for the problem in question is presented. Some implementational details about the solution of the resulting nonlinear system are given in Section 3. In Section 4, the results of several numerical experiments are provided. The extension of the method to the variable RBF shape parameter formulation as well as some preliminary numerical results are presented in Section 5. Some remarks and ideas for future work are given in Section 6.

ISSN: 2046-0546 (paper format), ISSN: 2046-0554 (online), http://www.witpress.com/journals
DOI: 10.2495/CMEM-V7-N3-246-259

2 RADIAL BASIS FUNCTION INTERPOLATION

2.1 The problem

The interpolation of a bivariate function $f : \overline{\Omega} \to \mathrm{R}$, where $\Omega \to \mathrm{R}^2$, where, from a set of sample values $\left\{f\left(x_{\mathrm{m}}, y_{\mathrm{m}}\right)\right\}_{\mathrm{m}=1}^{\mathrm{M}}$ on a discrete set $X = \left\{\left(x_{\mathrm{m}}, y_{\mathrm{m}}\right)\right\}_{\mathrm{m}=1}^{\mathrm{M}} \subset \overline{\Omega}$ is considered.

2.2 RBF approximation

The function *f* is approximated by the linear combination

$$f_{\mathrm{N}}\left(x, y\right) = \sum_{n=1}^{N} a_n \phi_n\left(x, y\right), \qquad \left(x, y\right) \in \overline{\Omega}, \tag{1}$$

where the RBFs $\phi_{\mathrm{n}}(x, y)$, n=1,…, *N*, are expressed in the form

$$\phi_{\mathrm{n}}\left(x, y\right) = \Phi\left(r_n\right), \qquad r_{\mathrm{n}}^2 = \left(x - \mathrm{x}_{\mathrm{n}}\right)^2 + \left(y - \mathrm{y}_{\mathrm{n}}\right)^2, \tag{2}$$

Each ϕ_{n} is associated with one of the points $\left\{\left(\mathrm{x}_{\mathrm{n}}, \mathrm{y}_{\mathrm{n}}\right)\right\}_{\mathrm{n}=1}^{\mathrm{N}}$ which are known as centres. A number $\mathrm{N}_{\mathrm{int}}$ of interior centres $\left\{\left(\mathrm{x}_{\mathrm{n}}, \mathrm{y}_{\mathrm{n}}\right)\right\}_{\mathrm{n}=1}^{\mathrm{N}_{\mathrm{int}}}$ is taken as well as $\mathrm{N}_{\mathrm{bry}}$ boundary centres $\left\{\left(\mathrm{x}_{\mathrm{n}}, \mathrm{y}_{\mathrm{n}}\right)\right\}_{\mathrm{n}=\mathrm{N}_{\mathrm{int}}+1}^{\mathrm{N}_{\mathrm{int}}+\mathrm{N}_{\mathrm{bry}}}$ yielding a total of $\mathrm{N} = \mathrm{N}_{\mathrm{int}} + \mathrm{N}_{\mathrm{bry}}$ centres. RBFs often include shape parameters which shall be denoted by *c* and, as already mentioned, the determination of an optimal value for these remains a major challenge.

Two popular RBFs, which include a shape parameter are:

- The normalized multiquadric (MQ) basis function

$$\phi_{\mathrm{n}}\left(x, y\right) = \Phi\left(r_{\mathrm{n}}\right) = \sqrt{\left(cr_{\mathrm{n}}\right)^2 + 1},\; r_{\mathrm{n}}^2 = \left(x - \mathrm{x}_{\mathrm{n}}\right)^2 + \left(y - \mathrm{y}_{\mathrm{n}}\right)^2, \tag{3}$$

and

- The inverse multiquadric (IMQ) basis function

$$\phi_{\mathrm{n}}\left(x, y\right) = \Phi\left(r_{\mathrm{n}}\right) = \frac{1}{\sqrt{\left(cr_{\mathrm{n}}\right)^2 + 1}},\; r_{\mathrm{n}}^2 = \left(x - \mathrm{x}_{\mathrm{n}}\right)^2 + \left(y - \mathrm{y}_{\mathrm{n}}\right)^2. \tag{4}$$

To emphasize the dependence of RBFs on the shape parameter, $\Phi(r_{\mathrm{n}})$ in (2) will be denoted by $\Phi(c, r_{\mathrm{n}})$ and thus approximation (1) will be replaced by

$$f_{\mathrm{N}}\left(x, y\right) = \sum_{n=1}^{N} a_n\, \Phi\left(c, r_{\mathrm{n}}\right), \qquad \left(x, y\right) \in \overline{\Omega}, \tag{5}$$

The unknown coefficients $\left\{a_{\mathrm{n}}\right\}_{n=1}^{\mathrm{N}}\{a_{\mathrm{n}}\}$ in eqn. (5) and the unknown shape parameter *c* are determined by collocation at the points $\left\{\left(x_{\mathrm{m}}, y_{\mathrm{m}}\right)\right\}_{\mathrm{m}=1}^{\mathrm{M}} \in \overline{\Omega}$

$$f_{\mathrm{N}}(x_{\mathrm{m}}, y_{\mathrm{m}}) = f\,(x_{\mathrm{m}}, y_{\mathrm{m}}), \quad \mathrm{m} = 1, \ldots, \mathrm{M}. \tag{6}$$

For the collocation points, $\mathrm{M}_{\mathrm{int}}$ interior points $\left\{\left(x_{\mathrm{m}}, y_{\mathrm{m}}\right)\right\}_{\mathrm{m}=1}^{\mathrm{M}_{\mathrm{int}}}$ and $\mathrm{M}_{\mathrm{bry}}$ boundary points $\left\{\left(x_{\mathrm{m}}, y_{\mathrm{m}}\right)\right\}_{\mathrm{m}=\mathrm{M}_{\mathrm{int}}+1}^{\mathrm{M}_{\mathrm{int}}+\mathrm{M}_{\mathrm{bry}}}$ are taken yielding a total of $\mathrm{M} = \mathrm{M}_{\mathrm{int}} + \mathrm{M}_{\mathrm{bry}}$ points. In general, the collocation points are not the same as the centres and, in order to have at least as many equations as unknowns, one needs to take $M \geq N + 1$.

2.3 Augmented polynomial basis

Often in RBF approximation the approximation (5) is replaced by

$$f_{\mathrm{N}}(x,y) = \sum_{\mathrm{n}=1}^{\mathrm{N}} a_n \Phi(c, r_n) + \sum_{\mathrm{k}=1}^{\mathrm{K}} a_{\mathrm{N+k}} p_{\mathrm{k}}(x,y), \qquad (x,y) \in \overline{\Omega} \tag{7}$$

where $\{p_{\mathrm{k}}\}_{\mathrm{k}=1}^{\mathrm{K}}$ is a basis of the set P_{p} of bivariate polynomials of total degree up to p. The polynomial basis is

$$p_{\mathrm{k}}(x,y) = x^{\mathrm{i-j}} y^{\mathrm{j}}, \; 0 \leq j \leq i, \; 0 \leq i \leq \mathrm{p}, \text{ for k} = 1, \ldots, \mathrm{K}, \tag{8}$$

where $\mathrm{K} = (\mathrm{p} + 1)(\mathrm{p} + 2)/2$. When using approximation (7), in addition to eqn. (6), the insolvency conditions

$$\sum_{\mathrm{n}=1}^{\mathrm{N}} a_{\mathrm{n}} \, p_{\mathrm{k}}(\mathrm{x_n}, \mathrm{y_n}) = 0, \; \mathrm{k} = 1, \ldots, \mathrm{K}. \tag{9}$$

are applied. In order to have at least as many equations as unknowns, one still needs to take $M \geq N + 1$.

3 IMPLEMENTATIONAL DETAILS

The presence of the shape parameter c in the unknowns renders the system of M + N eqns. (6) and (9)

$$\boldsymbol{F}(\boldsymbol{a}, c) := \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_{\mathrm{M+K}} \end{bmatrix} = \begin{bmatrix} f_{\mathrm{N}}(x_1, y_1) - f(x_1, y_1) \\ \vdots \\ f_{\mathrm{N}}(x_M, y_M) - f(x_M, y_M) \\ \sum_{\mathrm{n}=1}^{\mathrm{N}} a_{\mathrm{n}} \, p_1(x_{\mathrm{n}}, y_{\mathrm{n}}) \\ \vdots \\ \sum_{\mathrm{n}=1}^{\mathrm{N}} a_{\mathrm{n}} \, p_{\mathrm{K}}(\mathrm{x_n}, \mathrm{y_n}) \end{bmatrix} = 0, \tag{10}$$

nonlinear. In (10), the vector $\boldsymbol{a} = [a_1, \ldots, a_{\mathrm{N+K}}]^T$ contains the unknown coefficients in approximation (7).

The nonlinear system (10) can be solved with the MATLAB© [16] optimization toolbox routine `fsolve` which uses either a trust-region-dogleg/reflective algorithm or the Levenberg-Marquardt algorithm [17]. In `fsolve` the user may or may not provide the Jacobian. Alternatively, the nonlinear system (10) may be solved with the MATLAB© optimization toolbox routine `lsqnonlin` which solves nonlinear least squares problems using, as `fsolve`, either a subspace trust region method or the Levenberg-Marquardt algorithm. In this case, (10) is recast as a nonlinear least squares minimization problem by setting

$$S(\boldsymbol{a}, c) := \sum_{\mathrm{m}=1}^{\mathrm{M+K}} F_{\mathrm{m}}^2. \tag{11}$$

As with `fsolve`, in `lsqnonlin` the user may or may not provide the Jacobian and, in addition, the routine offers the option of imposing lower and upper bounds on the elements of the vector of unknowns $\boldsymbol{x} = [\boldsymbol{a}, c]^T$. In this way, constraints on the values of the coefficients and in particular on the values of the shape parameter may be imposed.

4 NUMERICAL EXAMPLES

In all the numerical examples considered in this paper, the approximate solution f_{N} was evaluated on a set of L test points in Ω. The maximum relative error

$$E = \frac{\left\| f - f_{\mathrm{N}} \right\|_{\infty,\Omega}}{\left\| f \right\|_{\infty,\Omega}} \tag{12}$$

and the root mean square error (RMSE)

$$\mathrm{E} = \sqrt{\frac{1}{\mathrm{L}} \sum_{l=1}^{\mathrm{L}} \left[f\left(x_l, y_l\right) - f_{\mathrm{N}}\left(x_l, y_l\right) \right]^2}. \tag{13}$$

were also calculated. In Example 1, both MQ (3) and IMQ (4) were used while in all the other numerical examples only MQ were used. In each case, once the value of the shape parameter c was determined from the solution of the nonlinear system, the solution was re-calculated for this value of c. The interior collocation points are taken to be the Halton points [6, Appendix A.1] while the boundary points are uniformly distributed and the centres are obtained in a similar way. For the test points L = 101 interior Halton points were taken. As the addition of the bivariate polynomials (8) did not improve the accuracy of the approximation, in all examples, in (7), p = 0 was taken. In all the numerical examples considered in this section, the domain Ω is the unit square (0, 1) × (0, 1) and M = 336, N = 278.

4.1 Example 1

The first example considered is the approximation of Franke's function [6, Section 2.2] (see Fig. 1)

$$\begin{aligned} f_1\left(x, y\right) = {} & \frac{3}{4} \mathrm{e}^{-\left(\frac{(9x-2)^2 + (9y-2)^2}{4}\right)} + \frac{3}{4} \mathrm{e}^{-\left(\frac{(9x+1)^2}{49} + \frac{(9y+1)^2}{10}\right)} + \frac{1}{2} \mathrm{e}^{-\left(\frac{(9x-7)^2 + (9y-3)^2}{4}\right)} \\ & - \frac{1}{5} \mathrm{e}^{-\left((9x-4)^2 + (9y-7)^2\right)}, \end{aligned} \tag{14}$$

which is a standard test function for data fitting.

4.1.1 Experiments with MQ

In Fig. 2(a) the results obtained by solving the linear approximation problem for a range of different (pre-assigned) values of the shape parameter are presented. This is done, in this and all subsequent examples of this section, in order to observe the behaviour of the errors with respect to the shape parameter and, in particular, to determine the optimal value of the latter. In Table 1 the corresponding results using the proposed technique (in the case the value of the shape parameter c is unknown), after 2000 iterations with different initial values c_0, are

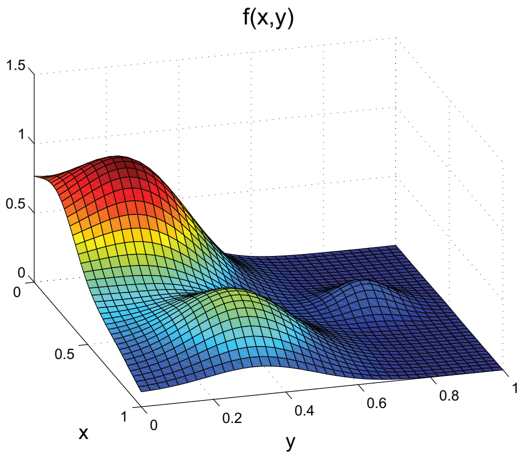
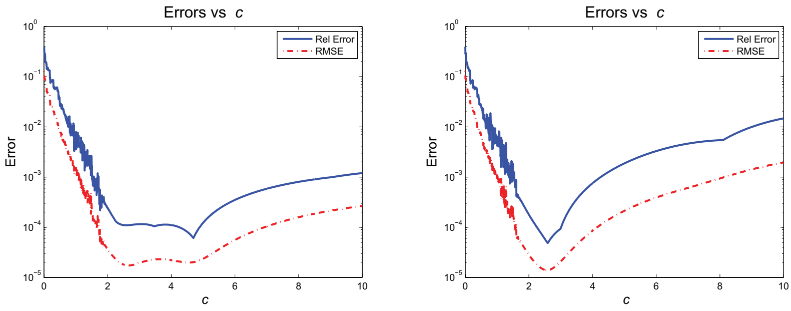


Figure 1: Example 1: Franke's function f_1.



(a) MQ

(b) IMQ

Figure 2: Example 1: Results with MQ and IMQ for varying shape parameter.

Table 1: Example 1: Results for various c_0 using MQ after 2000 iterations.

c_0	c	E	E
1.0	4.8689	9.4509(-5)	1.9220(-5)
2.0	4.8629	9.4089(-5)	1.9352(-5)
3.0	4.8663	9.4283(-5)	1.9025(-5)
4.0	4.8902	9.9538(-5)	1.9664(-5)
5.0	4.8768	9.7329(-5)	1.9331(-5)

presented. The obtained results are remarkably consistent and agree with the value of c giving the apparent minima in Fig. 2(a). The nonlinear system was solved with `fsolve`.

4.1.2 Experiments with IMQ

In Fig. 2(b) the results corresponding to those in Fig. 2(a) when using IMQ are presented. In Table 2 the results obtained using the proposed technique after 2000 iterations for different

Table 2: Example 1: Results for various c_0 using IMQ after 2000 iterations.

c_0	c	E	E
1.0	3.5573	4.0088(-4)	4.8095(-5)
2.0	3.5709	4.1049(-4)	4.8774(-5)
3.0	3.5687	4.0895(-4)	4.8864(-5)
4.0	3.5812	4.1749(-4)	4.9525(-5)
5.0	3.5785	4.1542(-4)	4.9293(-5)

initial values c_0 are listed. These again agree well with the value of c giving the apparent minima in Fig. 2(b). The nonlinear system was solved with `fsolve`.

4.2 Example 2

In this example the function [5, 7, 13]

$$f_2(x, y) = (1 + e^{-1/\varepsilon} - e^{-x/\varepsilon} - e^{(x-1)/\varepsilon})(1 + e^{-1/\varepsilon} - e^{-y/\varepsilon} - e^{(y-1)/\varepsilon}),$$

is considered. This function exhibits a boundary layer which becomes more severe as the value of $\varepsilon > 0$ decreases (see Fig. 3) and, as a result, the problem of approximating f_2 becomes more challenging as ε decreases. In Fig. 4(a)–(d) the results obtained by solving the linear approximation problem for a range of (pre-assigned) values of the shape parameter for $\varepsilon = 1$,0.5, 0.25 and 0.1, are presented. In Table 3 the corresponding results obtained using the proposed technique (after 2000 iterations) with different initial values c_0 are listed.

4.2.1 Case $\varepsilon = 1$

The solution in this case is relatively flat and the convergence of the proposed technique is quite rapid, which leads to high accuracy. Typical results obtained using MQ with different initial values for c_0 are presented in Table 3. These indicate that the obtained value of c is close to the one giving the minima in Fig. 4(a). In this case the nonlinear system was solved with `fsolve`.

4.2.2 Case $\varepsilon = 0.5$

In this case the solution is not as flat and, as a consequence, convergence is not as rapid as for $\varepsilon = 1$, and therefore the accuracy is not as high. In Table 3 typical results obtained using MQ for different initial values c_0 are presented. In this case the routine `lsqnonlin` was used and, again, the obtained value of the shape parameter c is close to the optimal such value from Fig. 4(b).

4.2.3 Case $\varepsilon = 0.25$

Now the solution is even less flat and the accuracy not as high as in the cases $\varepsilon = 1$ and $\varepsilon = 0.5$. Typical results obtained using MQ are presented in Table 3 for different initial values c_0. The routine `lsqnonlin` was used and the obtained value of the shape parameter c is close to the optimal such value from Fig. 4(c).

4.2.4 Case $\varepsilon = 0.1$

This is the most challenging case and the accuracy is much lower. In Table 3 typical results obtained using MQ for different initial values c_0 are presented. The routine `lsqnonlin` was

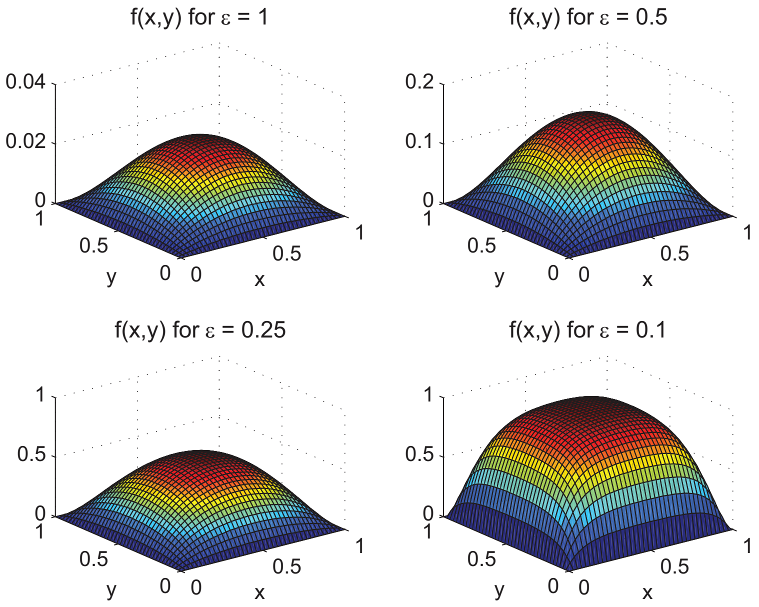
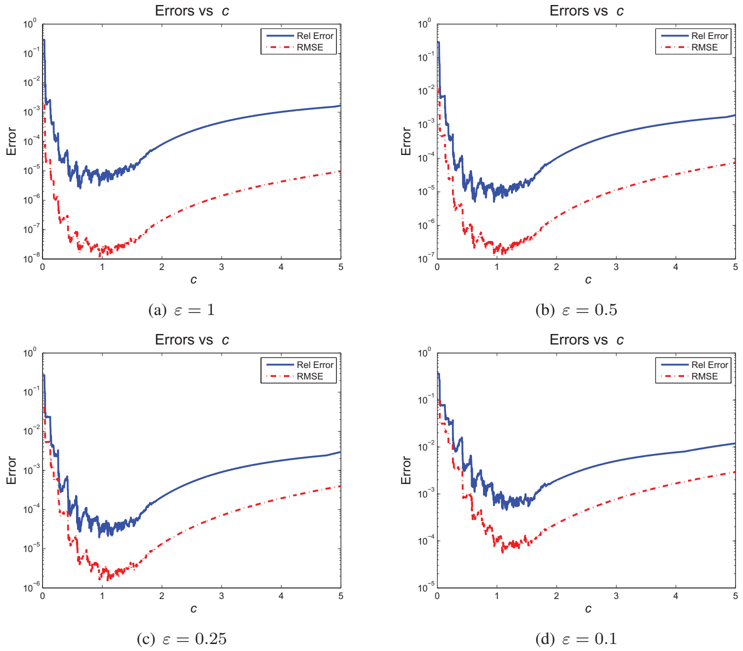


Figure 3: Example 2: The function $f_2(x, y)$.

employed and the obtained value of the shape parameter c is close to the optimal such value from Fig. 4(d).



(a) $\varepsilon = 1$

(b) $\varepsilon = 0.5$

(c) $\varepsilon = 0.25$

(d) $\varepsilon = 0.1$

Figure 4: Example 2: Results for varying shape parameter for ε = 1, 0.5, 0.25 and 0.1.

Table 3: Example 2: Results for various c_0 using MQ after 2000 iterations for $\varepsilon = 1$, 0.5, 0.25 and 0.1.

	c_0	c	E	E
$\varepsilon = 1$	0.4	1.4312	1.0344(-5)	3.5067(-8)
	0.5	1.4384	1.0296(-5)	3.5667(-8)
	0.6	1.4277	9.7104(-6)	3.3584(-8)
	0.7	1.4171	1.1609(-5)	3.3984(-8)
	0.8	1.4102	1.1596(-5)	3.4626(-8)
$\varepsilon = 0.5$				
	0.3	0.6222	4.1773(-6)	2.2671(-7)
	0.4	0.6301	4.9118(-6)	2.4898(-7)
	0.5	0.6477	8.2606(-6)	2.8472(-7)
	0.6	0.6310	4.7927(-6)	2.3990(-7)
	0.7	0.6575	1.1300(-5)	3.1951(-7)
$\varepsilon = 0.25$				
	0.3	0.8754	5.2289(-5)	3.7164(-6)
	0.4	0.9115	5.5683(-5)	3.8877(-6)
	0.5	0.9189	6.4466(-5)	4.2627(-6)
	0.6	0.9159	6.9501(-5)	4.6381(-6)
	0.7	0.9347	4.7612(-5)	3.0524(-6)
$\varepsilon = 0.1$				
	0.4	0.9668	8.4964(-4)	9.5993(-5)
	0.5	0.9857	7.2374(-4)	9.2681(-5)
	0.6	0.9955	9.2337(-4)	1.1115(-4)
	0.7	1.0219	7.3247(-4)	1.0153(-4)
	0.8	1.0111	9.5063(-4)	1.2014(-4)

4.3 Example 3

The function [3, Section 2.1.1] (see Fig. 5)

$$f_3(x, y) = \sin\left(\frac{\pi x}{6}\right) \sin\left(\frac{7\pi x}{4}\right) \cos\left(\frac{3\pi y}{4}\right) \cos\left(\frac{5\pi y}{4}\right)$$

is approximated. This function is different from f_1 and f_2 in that it is an oscillatory trigonometric function. In Fig. 6 the results obtained by solving the linear approximation problem for a range of (pre-assigned) values of the shape parameter are presented, while in Table 4 the corresponding results obtained with the proposed method for different initial values c_0 after 2000 iterations are listed. In this case, the routine `lsqnonlin` was used. The obtained results agree well with the apparent optimal value of c in Fig. 6.

4.4 Example 4

Finally, the function [9] (see Fig. 7)

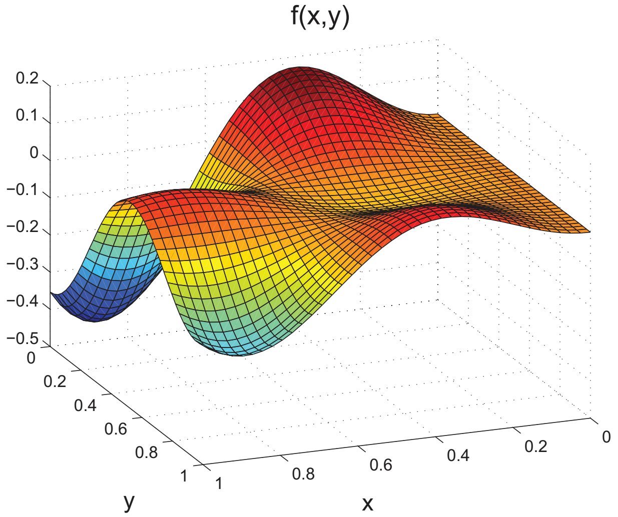


Figure 5: Example 3: The function $f_3(x, y)$.

Table 4: Example 3: Results for various c_0 using MQ after 2000 iterations.

c_0	c	E	E
0.2	1.3327	2.5836(-6)	1.8364(-7)
0.3	1.3566	1.9701(-6)	1.7476(-7)
0.4	1.3580	1.9872(-6)	1.7955(-7)
0.5	1.3659	5.1221(-6)	2.9474(-7)
0.6	1.3844	3.8214(-6)	2.8858(-7)

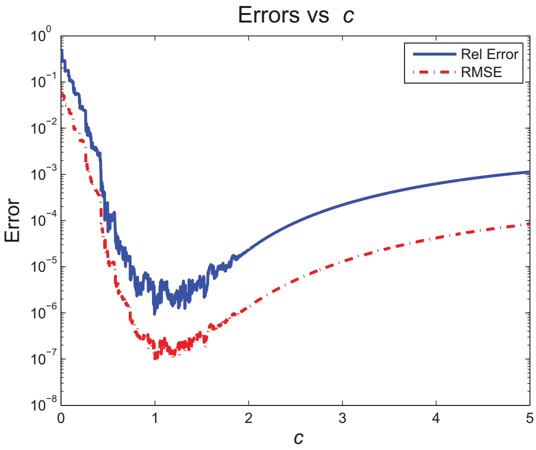


Figure 6: Example 3: Results for varying shape parameter.

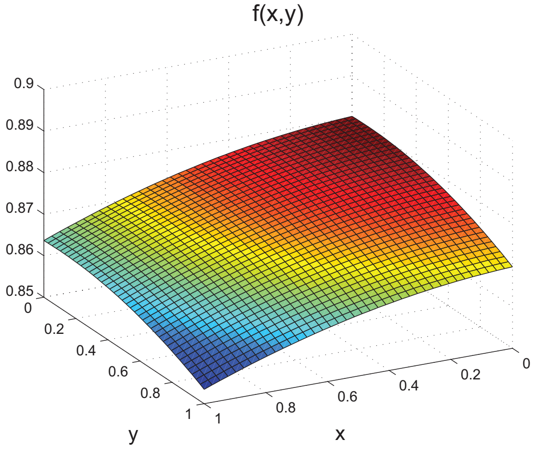


Figure 7: Example 4: The function $f_4(x, y)$.

$$f_4(x,y) = \frac{59}{67 + \left(x + \frac{1}{7}\right)^2 + \left(y - \frac{1}{11}\right)^2}$$

is approximated. In contrast to the previous functions considered, f_4 is very flat. The results obtained by solving the linear approximation problem for a range of (pre-assigned) values of the shape parameter are presented in Fig. 8, while in Table 5 the corresponding results obtained with the proposed method for different initial values c_0 are listed. In this case the routine `fsolve` was used and it was observed that the obtained results (after 2000 iterations) agree well with the apparent optimal value of c in Fig. 8.

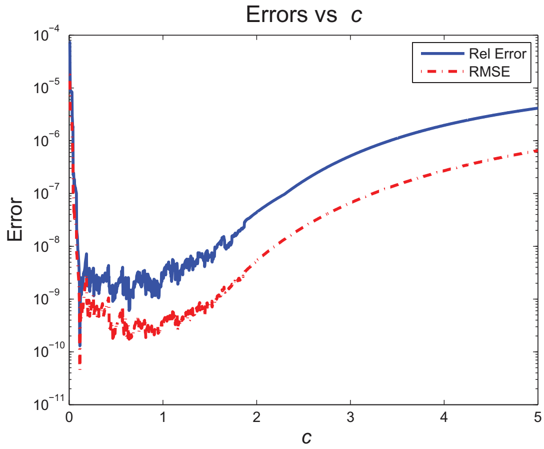


Figure 8: Example 4: Results for varying shape parameter.

Table 5: Example 4: Results for various c_0 using MQ after 2000 iterations.

c_0	c	E	E
0.1	0.2413	2.3610(-9)	7.8452(-10)
0.2	0.2437	2.9100(-9)	8.5599(-10)
0.3	0.2461	2.9079(-9)	9.0576(-10)
0.4	0.2430	2.7862(-9)	9.0576(-10)
0.5	0.2446	2.7667(-9)	8.7399(-10)

5 EXTENSION TO VARIABLE SHAPE PARAMETER APPROXIMATION

Now, a different shape parameter is associated with each centre and thus, instead of using approximation (5), the approximation

$$f_{\mathrm{N}}\left(x,y\right)=\sum_{\mathrm{n}=1}^{\mathrm{N}} a_n\,\Phi\left(c_n,r_n\right), \qquad \left(x,y\right)\in\overline{\Omega}, \tag{15}$$

is used and instead of using (7),

$$f_{\mathrm{N}}\left(x,y\right)=\sum_{\mathrm{n}=1}^{\mathrm{N}} a_n\,\Phi\left(c_n,r_n\right)+\sum_{\mathrm{k}=1}^{\mathrm{K}} a_{\mathrm{N+k}}\,p_{\mathrm{k}}\left(x,y\right), \qquad \left(x,y\right)\in\overline{\Omega}, \tag{16}$$

is used. In this approach, in addition to the coefficients $\left\{a_{\mathrm{n}}\right\}_{\mathrm{n}=1}^{\mathrm{N+K}}$ in (16), one needs to determine the unknown shape parameters $\left\{c_{\mathrm{n}}\right\}_{\mathrm{n}=1}^{\mathrm{N}}$. This is achieved by satisfying the collocation equations (6) and the insolvency conditions (9). The resulting nonlinear system is of the form (10) with c in $\boldsymbol{F}(\boldsymbol{a}, c)$ replaced by the vector $\boldsymbol{c}$, where $\boldsymbol{c} = [c_1, \ldots, c_{\mathrm{N}}]^T$, and, in order to have at least as may equations as unknowns, one now needs to take M ≥ 2N.

Two ways of initializing the starting values of the shape parameters were adopted. In the first one (Approach a), $\boldsymbol{c}_0 = c_0\,[1,1, \ldots ,1]$ where c_0 denotes the starting value of all shape parameters. In the second one (Approach b) a uniform initial distribution of shape parameters given by (see [19])

$$\boldsymbol{c}_0\left(l\right)=d_{\min}+\left(d_{\max}-d_{\min}\right)\frac{\left(l-1\right)}{\left(\mathrm{N}-1\right)}, \qquad l=1,\ldots,\mathrm{N},$$

was taken. In addition to the errors (12) and (13), $c_{\min}$ and $c_{\max}$ were calculated, which are the minimum and maximum values, respectively, of the entries of the final vector c.

5.1 Numerical example

As in Example 1, Franke's function (14) was approximated on the unit square. In Table 6 the results obtained with Approach a using `fsolve` for various numbers of degrees of freedom and different starting c_0 using MQ after 200 iterations are presented. The corresponding results obtained with Approach b for different $d_{\min}$, $d_{\max}$ are listed in Table 7. From these preliminary results it is observed that satisfactory accuracy may be obtained with relatively few degrees of freedom and a limited number of iterations.

Table 6: Example 1, Approach a: Results for various M, N and c_0 using MQ after 200 iterations.

M	N	c_0	$c_{\min}$	$c_{\max}$	E	E
452	148	2.0	1.1390	6.7495	2.2600(-3)	7.5785(-4)
452	148	2.5	1.6063	5.8086	1.7556(-3)	4.9926(-4)
452	148	3.0	2.1761	7.3573	2.5192(-3)	6.3261(-4)
492	188	2.0	1.1979	6.5165	8.2771(-4)	3.0365(-4)
492	188	2.5	1.6794	5.6618	7.6577(-4)	2.5493(-4)
492	188	3.0	2.5402	7.3493	1.2594(-3)	3.5584(-4)
536	248	2.0	1.2008	6.1611	3.2940(-4)	6.4558(-5)
536	248	2.5	1.6789	7.2949	3.2886(-4)	4.9834(-5)
536	248	3.0	2.2077	7.0799	3.3533(-4)	8.1232(-5)
561	278	2.0	1.1466	6.0327	1.7391(-4)	3.3371(-5)
561	278	2.5	1.8154	6.6718	9.0588(-5)	2.3758(-5)
561	278	3.0	2.4343	8.1001	1.6087(-4)	3.4088(-5)
656	278	2.0	1.2487	6.5684	1.4671(-4)	3.0017(-5)
656	278	2.5	1.6308	8.1263	5.8600(-5)	1.5845(-5)
656	278	3.0	2.2820	8.5574	8.7639(-5)	2.0150(-5)

Table 7: Example 1, Approach b: Results for various M, N and $d_{\min}$, $d_{\max}$ using MQ after 200 iterations.

M	N	$d_{\min}$	$d_{\max}$	$c_{\min}$	$c_{\max}$	E	E
452	148	2.5	3.5	2.3589	5.4495	2.2327(-3)	5.1564(-4)
452	148	3.0	4.0	2.4338	7.8863	2.3084(-3)	6.4032(-4)
452	148	2.0	4.0	1.9848	5.7519	2.1908(-3)	5.1638(-4)
492	188	2.5	3.5	2.3686	6.0079	8.8472(-4)	2.7468(-4)
492	188	3.0	4.0	2.8215	7.4603	1.4377(-3)	3.5393(-4)
492	188	2.0	4.0	1.9720	5.9779	6.9359(-4)	2.0169(-4)
536	248	2.5	3.5	2.4582	7.7133	1.7178(-4)	5.5196(-5)
536	248	3.0	4.0	2.9211	7.2200	2.6193(-4)	5.8612(-5)
536	248	2.0	4.0	1.9992	7.6053	5.9643(-4)	8.9178(-5)
561	278	2.5	3.5	2.4094	6.5063	1.5876(-4)	3.2555(-5)
561	278	3.0	4.0	2.9233	7.4067	2.2266(-4)	4.4236(-5)
561	278	2.0	4.0	2.0018	6.3572	2.5854(-4)	4.1789(-5)
656	278	2.5	3.5	2.4471	5.9029	4.9113(-5)	1.2664(-5)
656	278	3.0	4.0	2.9275	6.1518	5.9370(-5)	1.3736(-5)
656	278	2.0	4.0	1.9967	6.3194	1.3254(-4)	2.6770(-5)

6 CONCLUSIONS

The RBF collocation method was applied for the approximation of functions in two dimensions. Instead of pre-assigning the value of the shape parameter, this value is taken to be one of the unknowns along with the coefficients of the RBFs in the approximation and this leads to a nonlinear system of equations, which is solved using standard MATLAB© software. To demonstrate the robustness of the proposed approach, it was applied to several examples from the literature, each of which exhibits different features and hence presents different challenges. Provided the initial value of the shape parameter is taken within a certain range, it was observed that the obtained final value of the shape parameter was close to its optimal value. This idea was extended to the variable shape parameter case where each RBF used in the approximation is associated with a different value of the shape parameter. The results of some preliminary numerical experiments revealed that this approach is promising. A justified criticism of the proposed approaches is clearly the cost, as one is currently solving a nonlinear problem instead of a linear one in which case the value(s) of the shape parameter(s) is(are) pre-assigned. This however is also true of various other approaches proposed for the determination of a suitable shape parameter. The most accurate way to determine an optimal shape parameter is by brute force, i.e. by solving the (linear) problem for a sequence of values of (fixed) shape parameters and finding the one for which the approximation is most accurate. Clearly, the more shape parameter values taken, the more accurate the estimate. Since in most cases the exact solution is unknown, the accuracy of the approximation is assessed, for example, in terms of how well the boundary conditions are satisfied. This brute force approach, however, amounts to an (expensive) iterative method of solution of the problem. A good example of a similar iterative approach is the use of the golden section search algorithm for obtaining a good shape parameter [21]. Another expensive approach for obtaining a good shape parameter is the so-called leave-one-out cross validation proposed by Rippa [18]. One issue that requires further experimentation is the number of iterations required in the routines used, especially in the variable shape parameter case where the early indications are that high accuracy can be achieved with relatively few iterations. In addition, since both `fsolve` and `lsqnonlin` offer the user the option of providing the Jacobian of the nonlinear system under consideration, it would be worth investigating how this modification affects the speed of convergence and computational cost of the method.

REFERENCES

[1] Afiatdoust, F. & Esmaeilbeigi, M., Optimal variable shape parameters using genetic algorithm for radial basis function approximation. *Ain Shams Engineering Journal*, **6(2)**, pp. 639–647, 2015. https://doi.org/10.1016/j.asej.2014.10.019

[2] Buhmann, M.D., *Radial Basis Functions*, Cambridge University Press, Cambridge, 2003.

[3] Chen, W., Fu, Z.J. & Chen, C.S., *Recent Advances in Radial Basis Function Collocation Methods*, Springer Briefs in Applied Sciences and Technology, Springer, Heidelberg, 2014.

[4] Cheng, A.H.D., Multiquadric and its shape parameter - a numerical investigation of error estimate, condition number, and round-off error by arbitrary precision computation. *Engineering Analysis with Boundary Elements*, **36(2)**, pp. 220–239, 2012. https://doi.org/10.1016/j.enganabound.2011.07.008

[5] Fasshauer, G.E., Newton iteration with multiquadrics for the solution of nonlinear PDEs. *Computers & Mathematics with Applications*, **43(3–5)**, pp. 423–438, 2002. https://doi.org/10.1016/s0898-1221(01)00296-6

[6] Fasshauer, G.E., *Meshfree Approximation Methods with MATLAB*, Interdisciplinary Mathematical Sciences, Vol. 6, World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2007.

[7] Fasshauer, G.E., Gartland, E.C. & Jerome, J.W., Newton iteration for partial differential equations and the approximation of the identity. *Numerical Algorithms*, **25(1/4)**, pp. 181–195, 2000. https://doi.org/10.1023/a:1016609007255

[8] Fasshauer, G.E. & Zhang, J.G., On choosing optimal shape parameters for RBF approximation. *Numerical Algorithms,* **45**(**1–4**), pp. 345–368, 2007. https://doi.org/10.1007/s11075-007-9072-8

[9] Fornberg, B. & Flyer, N., Solving PDEs with radial basis functions. *Acta Numerica*, **24,** pp. 215–258, 2015. https://doi.org/10.1017/s0962492914000130

[10] Fornberg, B. & Wright, G., Stable computation of multiquadric interpolants for all values of the shape parameter. *Computers & Mathematics with Applications*, **48(5–6)**, pp. 853–867, 2004. https://doi.org/10.1016/j.camwa.2003.08.010

[11] Jankowska, M.A. & Karageorghis, A., Variable shape parameter Kansa RBF method for the solution of nonlinear boundary value problems. *Engineering Analysis with Boundary Elements*, **103**, pp. 32–40, 2019. https://doi.org/10.1016/j.enganabound.2019.02.005

[12] Jankowska, M.A., Karageorghis, A. & Chen, C.S., Kansa RBF method for nonlinear problems. *International Journal of Computational Methods and Experimental Measurements*, **6(6)**, pp. 1000–1007, 2018. https://doi.org/10.2495/cmem-v6-n5-1000-1007

[13] Jankowska, M.A., Karageorghis, A. & Chen, C.S., Improved Kansa RBF method for the solution of nonlinear boundary value problems. *Engineering Analysis with Boundary Elements*, **87**, pp. 173–183, 2018. https://doi.org/10.1016/j.enganabound.2017.11.012

[14] Kansa, E.J., Multiquadrics—a scattered data approximation scheme with applications to computational fluid-dynamics. II. Solutions to parabolic, hyperbolic and elliptic partial differential equations. *Computers & Mathematics with Applications*, **19**(**8–9**), pp. 147–161, 1990. https://doi.org/10.1016/0898-1221(90)90271-k

[15] Karageorghis, A. & Tryfonos, P., Determination of shape parameter in RBF approximation. *WIT Transactions on Engineering Sciences*, Vol. 122, WIT Press, pp. 153–162, 2019.

[16] The MathWorks, Inc., 3 Apple Hill Dr. *Matlab*, Natick, MA.

[17] Moré, J.J., The Levenberg-Marquardt algorithm: Implementation and theory. In *Numerical Analysis*, (Proc. 7th Biennial Conf., Univ. Dundee, Dundee, 1977), Lecture Notes in Math., Vol. 630, Springer, Berlin, pp. 105–116, 1978.

[18] Rippa, S., An algorithm for selecting a good value for the parameter c in radial basis function interpolation. *Advances in Computational Mathematics*, **11(2/3)**, pp. 193–210, 1999. https://doi.org/10.1023/a:1018975909870

[19] Sarra, S.A. & Sturgill, D., A random variable shape parameter strategy for radial basis function approximation methods. *Engineering Analysis with Boundary Elements*, **33(11)**, pp. 1239–1245, 2009. https://doi.org/10.1016/j.enganabound.2009.07.003

[20] Tryfonos, P., Some numerical experiments on radial basis function approximation. *MSc thesis, Department of Mathematics and Statistics*, University of Cyprus, 2018.

[21] Tsai, C.H., Kolibal, J. & Li, M., The golden section search algorithm for finding a good shape parameter for meshless collocation methods. *Engineering Analysis with Boundary Elements*, **34(8)**, pp. 738–746, 2010. https://doi.org/10.1016/j.enganabound.2010.03.003

[22] Xiang, S., Wang, K.M., Ai, Y.T., Sha, Y.D. & Shi, H., Thrigonometric variable shape parameter and exponent strategy for generalized multiquadric radial basis function approximation. *Applied Mathematical Modelling*, **36(5)**, pp. 1931–1938, 2012. https://doi.org/10.1016/j.apm.2011.07.076