DETERMINATION OF SHAPE PARAMETER IN RBF APPROXIMATION

ANDREAS KARAGEORGHIS & PINELOPI TRYFONOS
Department of Mathematics and Statistics, University of Cyprus, Cyprus

ABSTRACT

We apply a radial basis function (RBF) collocation method for the approximation of functions in two dimensions. The solution is approximated by a linear combination of radial basis functions. The issue of determining the optimal value of the shape parameter is tackled by including it in the unknowns along with the coefficients of the RBFs in the approximation. The resulting nonlinear system of equations is solved by directly applying a standard non-linear solver. The results of some numerical experiments are presented and analysed.

Keywords: radial basis functions, function approximation, collocation.

1 INTRODUCTION

In the last few decades radial basis functions (RBFs) have been used extensively for solving problems in various fields of engineering and scientific computing. Initially, RBF methods were used primarily for multivariate approximation [2], [5], the theoretical foundations of which may be found in [11], [16]. RBF approximation methods are particularly attractive due to their accuracy, flexibility and ability to handle scattered data in complex domains. In addition, they are meshless and very easy to implement. However, they suffer from a number of unfavourable issues. One such issue is the illconditioning of the global matrix which means that as the number of collocation points increases, the condition number of the matrix increases and, eventually, the accuracy of the solution suffers. Another unresolved issue in RBF approximation is the determination of the optimal value of the shape parameter present in many RBFs, as will be explained later. In the past, various techniques have been proposed for the determination of an appropriate value of the shape parameter [4], [9], [10], [18]. More recently, RBF methods have also become popular for the solution of boundary and initial value problems for partial differential equations, in particular due to the method proposed by Kansa [14], which has become known as the Kansa method. The same advantages and disadvantages as those mentioned for RBF approximation still hold for the Kansa method. A recent review of Kansa RBF methods for the numerical solution of partial differential equations may be found in [3].

In the current work we attempt to find a way of resolving the shape parameter issue when RBF methods are used to approximate functions in two dimensional domains. In particular, we use an idea developed in [12], [13] for applications of the Kansa-RBF method [14] to nonlinear boundary value problems. In the RBF approximation of the function in question we let the shape parameter be one of the unknowns. Since the shape parameter appears nonlinearly in the RBF expansion, the collocation of the approximation at various points in the domain now yields a system of nonlinear equations. This system is then solved using standard software for the solution of nonlinear systems which yields, in addition to the unknown coefficients in the RBF expansion, a value for the shape parameter. It is shown, in several examples, that, provided the initial guess is taken within a certain range, this value is close to the optimal shape parameter value. The results of further numerical experiments may be found in [20].

The paper is organized as follows. In Section 2 we present the problem and describe the RBF formulation. The details regarding the solution of the resulting nonlinear system are

WIT Transactions on Engineering Sciences, Vol 122, © 2019 WIT Press
www.witpress.com, ISSN 1743-3533 (on-line)
doi:10.2495/BE410141

presented in Section 3. In Section 4 we present the results of two numerical examples. Finally, some concluding remarks and ideas for future work are provided in Section 5.

2 RADIAL BASIS FUNCTION INTERPOLATION

2.1 The problem

We consider the interpolation of a bivariate function $f : \overline{\Omega} \to \mathbb{R}$, where $\Omega \subset \mathbb{R}^2$, from a set of sample values $\{f(x_\mathsf{m}, y_\mathsf{m})\}_{\mathsf{m}=1}^{\mathsf{M}}$ on a discrete set $X = \{(x_\mathsf{m}, y_\mathsf{m})\}_{\mathsf{m}=1}^{\mathsf{M}} \subset \overline{\Omega}$.

2.2 RBF approximation

We approximate f by a linear combination of radial basis functions (RBFs)

$$f_\mathsf{N}(x,y) = \sum_{\mathsf{n}=1}^{N} a_\mathsf{n} \phi_\mathsf{n}(x,y), \qquad (x,y) \in \overline{\Omega}. \tag{1}$$

The RBFs $\phi(x,y)$, $\mathsf{n} = 1, \dots, N$ can be expressed in the form

$$\phi_\mathsf{n}(x,y) = \Phi(r_\mathsf{n}), \qquad r_\mathsf{n}^2 = (x - x_\mathsf{n})^2 + (y - y_\mathsf{n})^2. \tag{2}$$

Thus each RBF ϕ_n is associated with a point $(x_\mathsf{n}, y_\mathsf{n})$. These points $\{(x_\mathsf{n}, y_\mathsf{n})\}_{\mathsf{n}=1}^{\mathsf{N}}$ are usually referred to as *centers*. We shall assume that we have N_int interior centers $\{(x_\mathsf{n}, y_\mathsf{n})\}_{\mathsf{n}=1}^{\mathsf{N}_\mathsf{int}}$ and N_bry boundary centers $\{(x_\mathsf{n}, y_\mathsf{n})\}_{\mathsf{n}=\mathsf{N}_\mathsf{int}+1}^{\mathsf{N}_\mathsf{int}+\mathsf{N}_\mathsf{bry}}$ and take $\mathsf{N} = \mathsf{N}_\mathsf{int} + \mathsf{N}_\mathsf{bry}$. Often, RBFs include *shape parameters*. In this paper, the shape parameter will be denoted by c. The determination of the optimal value of shape parameter remains a major challenge.

For example, we mention the following popular RBFs which include a shape parameter:

- the normalized multiquadric (MQ) basis function

$$\phi_\mathsf{n}(x,y) = \Phi(r_\mathsf{n}) = \sqrt{(cr_\mathsf{n})^2 + 1}, \qquad r_\mathsf{n}^2 = (x - x_\mathsf{n})^2 + (y - y_\mathsf{n})^2, \tag{3}$$

  and
- the inverse multiquadric (IMQ) basis function

$$\phi_\mathsf{n}(x,y) = \Phi(r_\mathsf{n}) = \frac{1}{\sqrt{(cr_\mathsf{n})^2 + 1}}, \qquad r_\mathsf{n}^2 = (x - x_\mathsf{n})^2 + (y - y_\mathsf{n})^2. \tag{4}$$

In the case of shape parameter-dependent RBFs we shall denote $\Phi(r_\mathsf{n})$ in (2) by $\Phi(c, r_\mathsf{n})$. We therefore replace approximation (1) by

$$f_\mathsf{N}(x,y) = \sum_{\mathsf{n}=1}^{\mathsf{N}} a_\mathsf{n} \, \Phi(c, r_\mathsf{n}), \qquad (x,y) \in \overline{\Omega}. \tag{5}$$

The coefficients $\{a_\mathsf{n}\}_{\mathsf{n}=1}^{\mathsf{N}}$ in eqn (5) are determined from the collocation equations

$$f_\mathsf{N}(x_\mathsf{m}, y_\mathsf{m}) \;=\; f(x_\mathsf{m}, y_\mathsf{m}), \quad \mathsf{m} = 1, \dots, \mathsf{M}, \tag{6}$$

where the points $\{(x_\mathsf{m}, y_\mathsf{m})\}_{\mathsf{m}=1}^{\mathsf{M}} \in \overline{\Omega}$ are the *collocation points*. Of these, we have M_int interior collocation points $\{(x_\mathsf{m}, y_\mathsf{m})\}_{\mathsf{m}=1}^{\mathsf{M}_\mathsf{int}}$ and M_bry boundary collocation points $\{(x_\mathsf{m}, y_\mathsf{m})\}_{\mathsf{m}=\mathsf{M}_\mathsf{int}+1}^{\mathsf{M}_\mathsf{int}+\mathsf{M}_\mathsf{bry}}$ and take $\mathsf{M} = \mathsf{M}_\mathsf{int} + \mathsf{M}_\mathsf{bry}$. Note that, in general, the collocation points are not the same as the centers. Clearly, we need to have $\mathsf{M} \geq \mathsf{N} + 1$.

2.3 Improved RBF approximation

Let $\mathbb{P}_\mathsf{p}$ be the set of bivariate polynomials of total degree up to p and $\{p_\mathsf{k}\}_{\mathsf{k}=1}^{\mathsf{K}}$ be a basis of $\mathbb{P}_\mathsf{p}$. It is known that the number of polynomial terms for degree p is $\mathsf{K} = (\mathsf{p}+1)(\mathsf{p}+2)/2$. The polynomial basis is thus

$$p_\mathsf{k}(x,y) = x^{i-j}y^j, \quad 0 \le j \le i,\ 0 \le i \le \mathsf{p}, \quad \text{for} \quad \mathsf{k} = 1,\dots,\mathsf{K}. \tag{7}$$

In the RBF approximation with an augmented polynomial basis, the approximation (5) is replaced by

$$f_\mathsf{N}(x,y) = \sum_{\mathsf{n}=1}^{\mathsf{N}} a_\mathsf{n}\,\Phi(c, r_\mathsf{n}) + \sum_{\mathsf{k}=1}^{\mathsf{K}} a_{\mathsf{N}+\mathsf{k}}\,p_\mathsf{k}(x,y), \qquad (x,y) \in \overline{\Omega}. \tag{8}$$

Also, in addition to the collocation eqns (6), we have the insolvency conditions

$$\sum_{\mathsf{n}=1}^{\mathsf{N}} a_\mathsf{n}\,p_\mathsf{k}(\mathsf{x}_\mathsf{n}, \mathsf{y}_\mathsf{n}) \ = 0, \quad \mathsf{k} = 1,\dots,\mathsf{K}. \tag{9}$$

Clearly, we still need to have $\mathsf{M} \ge \mathsf{N}+1$.

3 IMPLEMENTATIONAL DETAILS

Since the shape parameter appears nonlinearly in the collocation eqns (6), the system of $M+K$ eqns (6) and (9) is nonlinear and can be written in the form

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

where $\boldsymbol{a} = [a_1, \dots, a_{\mathsf{N}+\mathsf{K}}]^T$ are the coefficients in approximation (8).

The nonlinear system (10) may be solved with the MATLAB$^{©}$ [15] optimization toolbox routine `fsolve`. The algorithm used is either a trust-region-dogleg/reflective or the Levenberg–Marquardt algorithm [17]. The routine `fsolve` does not require the user to provide the gradient. When the system is not square, i.e. the number of equations is larger than the number of unknowns, which is the case in the current problem, the routine uses the Levenberg–Marquardt algorithm. Alternatively, the solution of the nonlinear system (10) may be recast as a nonlinear least squares minimization problem by setting

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

where the $\{F_\mathsf{m}\}_{\mathsf{m}=1}^{\mathsf{M}+\mathsf{K}}$ are defined in (10). The functional $S(\boldsymbol{a}, c)$ is then minimized using the MATLAB$^{©}$ optimization toolbox routine `lsqnonlin` which solves nonlinear least squares problems using, like `fsolve`, either a subspace trust region method or the Levenberg–Marquardt algorithm. The routine `lsqnonlin` does not require the user to provide the gradient and, in addition, offers the option of imposing lower and upper bounds on the elements of the vector of unknowns $\boldsymbol{x} = [\boldsymbol{a}, c]^T$ through the vectors `lb` and `up`. We can thus easily impose the constraints on the values of the coefficients and in particular on the values of the shape parameter. It should be noted that both `lsqnonlin` and `fsolve` might give only local solutions.

4 NUMERICAL EXAMPLES

In the numerical examples considered in this section, we calculated the approximate solution f_{N} at L test points in Ω. We calculated the maximum relative error E defined by

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

and the root mean square error (RMSE) $\mathcal{E}$ from

$$\mathcal{E} = \sqrt{\frac{1}{\mathsf{L}}\sum_{\ell=1}^{\mathsf{L}} \left[f(x_\ell, y_\ell) - f_{\mathsf{N}}(x_\ell, y_\ell)\right]^2}\,. \tag{13}$$

In Example 1 we used MQ (3) and IMQ (4) while in the other numerical examples we used only the MQ. For all examples considered the analytical solution is known. In each case, once the value of the shape parameter c was obtained, the solution was re-calculated for this value of c. The interior collocation points are taken to be the Halton points [7, Appendix A.1] while the boundary points are uniformly distributed. The corresponding boundary and interior centers are obtained in a similar way. We chose a set of $\mathsf{L} = 101$ interior Halton points to be the set of test points. These may be generated using the MATLAB$^{©}$ code `haltonseq.m`. It was observed that the addition of the bivariate polynomials did not improve the accuracy of the solution, and hence in all the examples presented we took $\mathsf{p} = 0$. In both numerical examples, the domain Ω was the unit square $(0, 1) \times (0, 1)$.

4.1 Example 1

We approximate Franke's function [7, Section 2.2]

$$f(x, y) = \frac{3}{4}\mathrm{e}^{-\left(\frac{(9x-2)^2+(9y-2)^2}{4}\right)} + \frac{3}{4}\mathrm{e}^{-\left(\frac{(9x+1)^2}{49}+\frac{(9y+1)^2}{10}\right)}$$
$$+\frac{1}{2}\mathrm{e}^{-\left(\frac{(9x-7)^2+(9y-3)^2}{4}\right)} - \frac{1}{5}\mathrm{e}^{-\left((9x-4)^2+(9y-7)^2\right)}$$

(see Fig. 1).

4.1.1 Experiments with MQ

In Fig. 2(a) we present the results obtained while fixing the value of the shape parameter for various such values in the case $\mathsf{M} = 336$, $\mathsf{N} = 278$. In Table 1 we present the corresponding results for the same numbers of collocation points and centers but with the value of the shape parameter c free with different initial values c_0. The obtained results for different c_0 are remarkably consistent and agree with the value of c giving the apparent minima in Fig. 2(a).

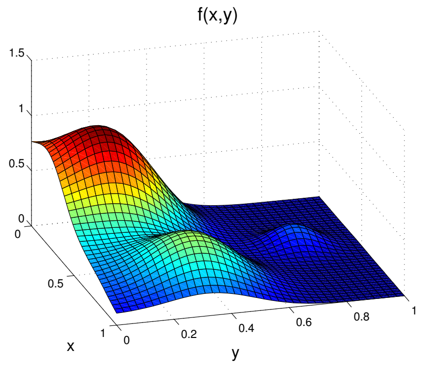


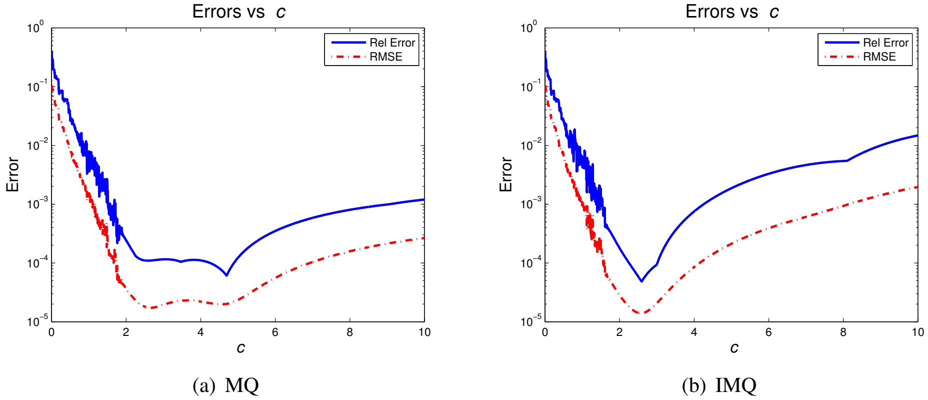
Figure 1: Example 1. The function $f(x, y)$.

Table 1: Example 1: Results for various c_0 using MQ with $\mathsf{M} = 336$, $\mathsf{N} = 278$ after 2000 iterations.

M	N	c_0	c	E	$\mathcal{E}$
336	278	1.0000	4.8689	9.4509(-5)	1.9220(-5)
336	278	2.0000	4.8629	9.4089(-5)	1.9352(-5)
336	278	3.0000	4.8663	9.4283(-5)	1.9025(-5)
336	278	4.0000	4.8902	9.9538(-5)	1.9664(-5)
336	278	5.0000	4.8768	9.7329(-5)	1.9331(-5)

4.1.2 Experiments with IMQ

In Fig. 2(b) we present the results obtained while fixing the value of the shape parameter for various such values in the case $\mathsf{M} = 336$, $\mathsf{N} = 278$. In Table 2 we present the corresponding results for the same numbers of collocation points and centers but with the value of the



(a) MQ

(b) IMQ

Figure 2: Example 1. Results with MQ and IMQ for varying shape parameter for $\mathsf{M} = 336, \mathsf{N} = 278, \mathsf{p} = 0$.

Table 2: Example 1: Results for various c_0 using IMQ with $\mathsf{M} = 336$, $\mathsf{N} = 278$ after 2000 iterations.

M	N	c_0	c	E	$\mathcal{E}$
336	278	1.0000	3.5573	4.0088(-4)	4.8095(-5)
336	278	2.0000	3.5709	4.1049(-4)	4.8774(-5)
336	278	3.0000	3.5687	4.0895(-4)	4.8864(-5)
336	278	4.0000	3.5812	4.1749(-4)	4.9525(-5)
336	278	5.0000	3.5785	4.1542(-4)	4.9293(-5)

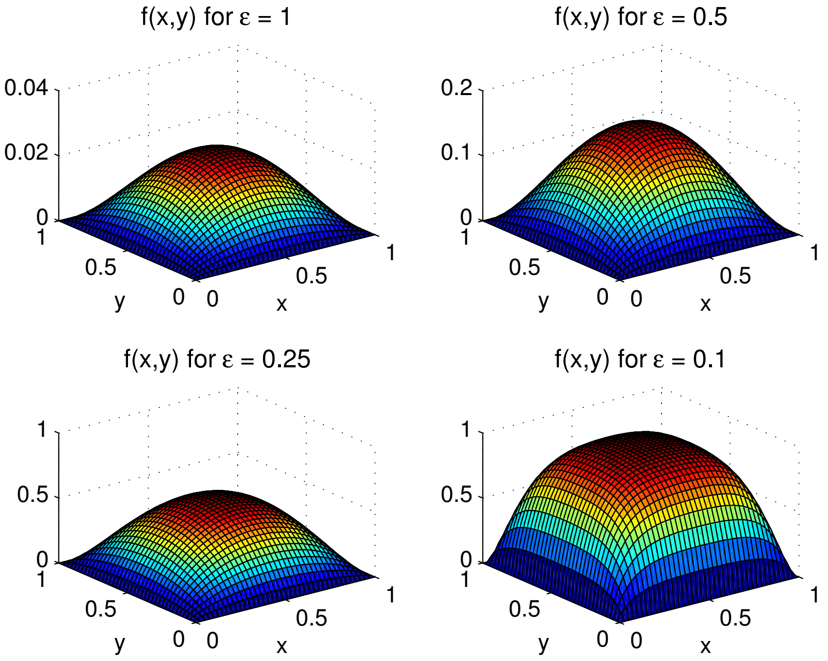


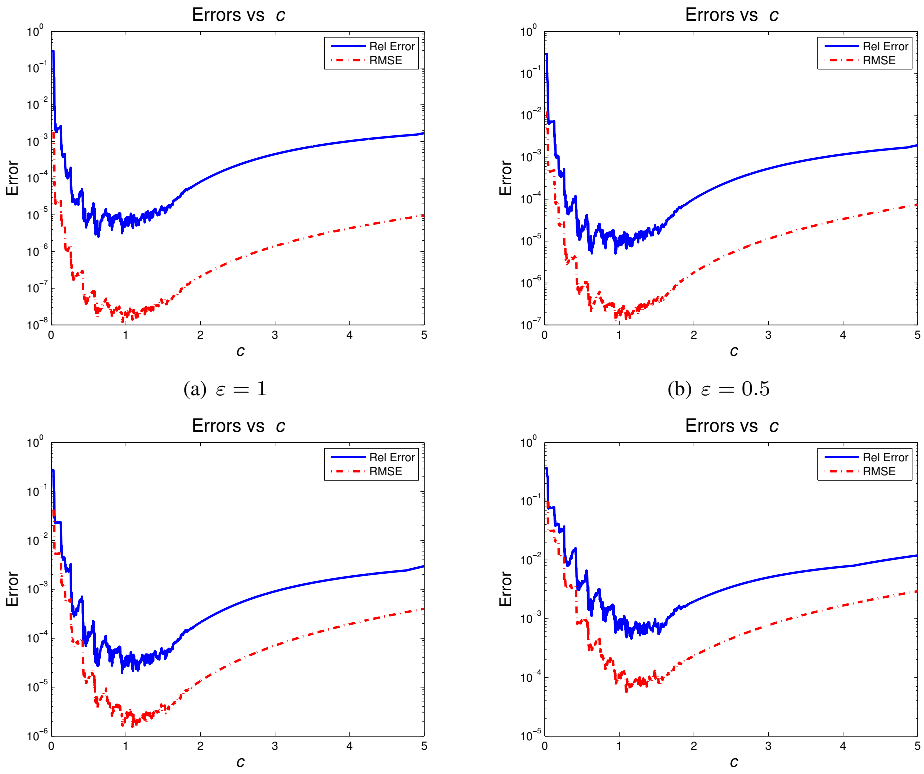
Figure 3: Example 2. The function $f(x, y)$.

shape parameter c free with different initial values c_0. The obtained results for different c_0 are remarkably consistent and agree well with the value of c giving the apparent minima in Fig. 2(b).

4.2 Example 2

We next approximate the function [6], [8], [13]

$$f(x,y) = \left(1 + \mathrm{e}^{-1/\varepsilon} - \mathrm{e}^{-x/\varepsilon} - \mathrm{e}^{(x-1)/\varepsilon}\right)\left(1 + \mathrm{e}^{-1/\varepsilon} - \mathrm{e}^{-y/\varepsilon} - \mathrm{e}^{(y-1)/\varepsilon}\right).$$



(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 $\varepsilon = 1, 0.5, 0.25$ and 0.1, $\mathsf{M} = 336, \mathsf{N} = 278, \mathsf{p} = 0$.

As $\varepsilon > 0$ decreases, the problem becomes more difficult (see Fig. 3). In Figs 4(a)–(d) we present the results obtained while fixing the value of the shape parameter for various such values in the case $\mathsf{M} = 336, \mathsf{N} = 278$ and $\varepsilon = 1$, 0.5, 0.25 and 0.1. In Table 3 we present the corresponding results for the same numbers of collocation points and centers but with the value of the shape parameter c free with different initial values c_0.

4.2.1 Case $\varepsilon = 1$

In this case the solution is quite flat and convergence is rapid. In Table 3 we present typical results obtained using MQ RBFs with different initial values for c_0. The results show that the final value of c is close to the one giving a minimum in Fig. 4(a). The nonlinear system was solved with the MATLAB© optimization toolbox routine `fsolve`.

4.2.2 Case $\varepsilon = 0.5$

In this case the solution is not as flat and convergence is not as rapid as for $\varepsilon = 1$. In Table 3 we present typical results obtained using MQ RBFs. We used the routine `lsqnonlin` for different initial values c_0. The find value of the shape parameter c is close to the optimal value of Fig. 4(b).

Table 3: Example 2: Results for various c_0 using MQ with $\mathsf{M} = 336$, $\mathsf{N} = 278$ after 2000 iterations for $\varepsilon = 1$, 0.5, 0.25 and 0.1.

	M	N	c_0	c	E	$\mathcal{E}$
$\varepsilon = 1$						
	336	278	0.4000	1.4312	1.0344(-5)	3.5067(-8)
	336	278	0.5000	1.4384	1.0296(-5)	3.5667(-8)
	336	278	0.6000	1.4277	9.7104(-6)	3.3584(-8)
	336	278	0.7000	1.4171	1.1609(-5)	3.3984(-8)
	336	278	0.8000	1.4102	1.1596(-5)	3.4626(-8)
$\varepsilon = 0.5$						
	336	278	0.3000	0.6222	4.1773(-6)	2.2671(-7)
	336	278	0.4000	0.6301	4.9118(-6)	2.4898(-7)
	336	278	0.5000	0.6477	8.2606(-6)	2.8472(-7)
	336	278	0.6000	0.6310	4.7927(-6)	2.3990(-7)
	336	278	0.7000	0.6575	1.1300(-5)	3.1951(-7)
$\varepsilon = 0.25$						
	336	278	0.3000	0.8754	5.2289(-5)	3.7164(-6)
	336	278	0.4000	0.9115	5.5683(-5)	3.8877(-6)
	336	278	0.5000	0.9189	6.4466(-5)	4.2627(-6)
	336	278	0.6000	0.9159	6.9501(-5)	4.6381(-6)
	336	278	0.7000	0.9347	4.7612(-5)	3.0524(-6)
$\varepsilon = 0.1$						
	336	278	0.4000	0.9668	8.4964(-4)	9.5993(-5)
	336	278	0.5000	0.9857	7.2374(-4)	9.2681(-5)
	336	278	0.6000	0.9955	9.2337(-4)	1.1115(-4)
	336	278	0.7000	1.0219	7.3247(-4)	1.0153(-4)
	336	278	0.8000	1.0111	9.5063(-4)	1.2014(-4)

4.2.3 Case $\varepsilon = 0.25$

Again, the solution is not as flat and convergence is not as rapid as for $\varepsilon = 1$ and $\varepsilon = 0.5$. In Table 3 we present typical results obtained using MQ RBFs. We used the routine `lsqnonlin` for different initial values c_0. The find value of the shape parameter c is close to the optimal value of Fig. 4(c).

4.2.4 Case $\varepsilon = 0.1$

This is the most difficult case. Covergence is much slower as the solution is not flat. In Table 3 we present typical results obtained using MQ RBFs. We used the routine `lsqnonlin` for different initial values c_0. The find value of the shape parameter c is close to the optimal value of Fig. 4(d).

5 CONCLUSIONS

In this work we have considered the approximation of functions by RBF expansions. The coefficients in these expansions are obtained by collocation. The issue of the optimal value of the shape parameter in the RBFs used is addressed by including its value in the unknowns of the problem to be determined by collocation. The resulting system of nonlinear equations is solved using standard software. The results of several tests reveal that, provided the initial value is taken within a certain range, the value of the shape parameter obtained is close to the optimal value of the shape parameter. A criticism of the proposed approach is clearly the cost, as we are essentially turning a linear problem into a nonlinear one. This however has been the case with various other approaches proposed for the determination of a suitable shape parameter. For example, we mention the use of the so called leave-one-out cross validation (LOOCV) proposed by Rippa [18]. Future work could involve the use of different shape parameters for a each RBF in the approximation expansion. The idea of using variable shape parameters in RBF approximations has been previously used in [1], [19], [21]. Moreover, since both `fsolve` and `lsqnonlin` offer the user the option of providing the Jacobian of the nonlinear system under consideration, it would be interesting to study how this modification affects the convergence of the method.

REFERENCES

[1] Afiatdoust, F. & Esmaeilbeigi, M., Optimal variable shape parameters using genetic algorithm for radial basis function approximation. *Ain Shams Eng. J.*, **6** (2015), 639–647.

[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. *Eng. Anal. Bound. Elem.*, **36** (2012), 220–239.

[5] Dyn, N., *Interpolation of scattered data by radial basis function. Topics in Multivariate Approximation*, eds. C.K. Chui, L.L. Schumarket & F.I. Utreras, Academic Press: New York, pp. 47–61, 1987.

[6] Fasshauer, G.E., Newton iteration with multiquadrics for the solution of nonlinear PDEs. *Comput. Math. Appl.*, **43** (2002), 423–438.

[7] Fasshauer, G.E., Meshfree approximation methods with MATLAB. *Interdisciplinary Mathematical Sciences*, vol. 6, World Scientific Publishing Co. Pte. Ltd: Hackensack, NJ, 2007.

[8] Fasshauer, G.E., Gartland, E.C. & Jerome, J.W., Newton iteration for partial differential equations and the approximation of the identity. *Numer. Algorithms*, **25** (2000), 181–195.

[9] Fasshauer, G.E. & Zhang, J.G., On choosing optimal shape parameters for RBF approximation. *Numer. Algorithms*, **45** (2007), 345–368.

[10] Fornberg, B. & Wright, G., Stable computation of multiquadric interpolants for all values of the shape parameter. *Comput. Math. Appl.*, **48**(5-6) (2004), 853–867.

[11] Franke, R., Scattered data interpolation: tests of some methods. *Math. Comp.*, **48** (1982), 181–200.

[12] Jankowska, M.A., Karageorghis, A. & Chen, C.S., Kansa RBF method for nonlinear problems. *Int. J. Comp. Meth. Exp. Meas.*, **6** (2018), 1000–1007.

[13] Jankowska, M.A., Karageorghis, A. & Chen, C.S., Improved Kansa RBF method for the solution of nonlinear boundary value problems. *Eng. Anal. Bound. Elem.*, **87** (2018), 173–183.

[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. *Comput. Math. Appl.*, **19** (1990), 147–161.

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

[16] Micchelli, C.A., Interpolation of scattered data: distance matrices and conditionally positive definite functions. *Constr. Approx.*, **2** (1986), 11–12.

[17] Moré, J.J., The Levenberg–Marquardt algorithm: implementation and theory. *Numer. Analysis*, (1977), 105–116.

[18] Rippa, S., An algorithm for selecting a good value for the parameter c in radial basis function interpolation. *Adv. Comput. Math.*, **11** (1999), 193–210.

[19] Sarra, S.A. & Sturgill, D., A random variable shape parameter strategy for radial basis function approximation methods. *Eng. Anal. Bound. Elem.*, **33** (2009), 1239–1245.

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

[21] 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. *Appl. Math. Model.*, **36** (2012), 1931–1938.