

# RBF Based Grid-Free Local Scheme With Spatially Variable Optimal Shape Parameter for Steady Convection-Diffusion Equations

Sanyasiraju V.S.S. Yedida\* and Chirala Satyanarayana  
Department of Mathematics, Indian Institute of Technology Madras,  
Chennai, 600036, INDIA

Received: 25/12/2011 – Revised 24/11/2012 – Accepted 06/12/2012

## Abstract

In this work, a *local* algorithm has been proposed to obtain an optimal shape parameter for the infinitely smooth Radial Basis Functions (RBF) when they have been used to solve Convection-Diffusion Equations (CDE) under grid-free environment. The algorithm is based on re-construction of the forcing term in the CDE using the collocation over the centers of the local support domain. The residual errors are calculated using the Rippa's "leave one out cross validation" algorithm originally been developed for interpolation using RBF functions. It has been shown that the cost function and RMS error functions obtained with the developed local scheme are oscillation free unlike the existing global collocation schemes. It has been also observed that, for most of the diffusion dominated problems, the pattern of the (*global*) Cost function of the proposed algorithm is appeared to be similar to the (R.M.S) error function, however, the same is not been found true for convection dominated problems. Therefore, for the latter case, an (near) optimal variable shape parameter has been obtained by minimizing the local Cost function at each center (node) of the computational domain. The Local RBF (LRBF) scheme with the proposed local optimization algorithm has been tested over several one and two-dimensional linear convection-diffusion problems with strong boundary layer and found to be accurate.

*Keywords: grid-free scheme; radial basis function; convection-diffusion; multiquadric; optimal shape parameter.*

## 1. Introduction

The Radial Basis Function(RBF) based grid-free local schemes are becoming very attractive choice for solving fluid flow and heat transfer problems due to their better conditioning and flexibility in handling the non-linearities. When infinitely smooth RBFs are used then the shape parameter of these RBFs plays a very significant role in obtaining accurate and stable solutions. In most of the existing literature, researchers have chosen the shape parameter either by trial and error or by some ad hoc means. This strategy may not give accurate and stable solutions for the problems with non-smooth solutions like convection dominated problems of the convection-diffusion equations. The convection-diffusion equation may be regarded as a simplified model problem to

\* Corresponding Author: Sanyasiraju VSS Yedida  
Email: [sryedida@iitm.ac.in](mailto:sryedida@iitm.ac.in) Phone: + 91 44 22574621  
© 2013 All rights reserved. ISSR Journals

Fax: +91 44 22574602  
PII: S2180-1363(12)44151-X

Navier-Stokes equations and plays an important role in Computational Fluid Dynamics(CFD). The steady form of the CDE is given by

$$\mathcal{L}u(\underline{x}) = f(\underline{x}), \forall \underline{x} \in \Omega \subset \mathbb{R}^d \quad (1.1)$$

$$\mathcal{B}u(\underline{x}) = g(\underline{x}), \forall \underline{x} \in \partial\Omega \subset \mathbb{R}^d \quad (1.2)$$

where  $\mathcal{L} = \bar{b} \cdot \nabla - a\Delta$ ,  $\bar{b} \cdot \nabla = b_j \frac{\partial}{\partial x_j}$ ,  $\Delta = \frac{\partial^2}{\partial x_j \partial x_j}$ ,  $\bar{b}(\underline{x})$ , is the convection coefficient,  $a (> 0)$  is the diffusion coefficient,  $\mathcal{B} = \alpha + \beta \frac{\partial}{\partial x_j}$  is the boundary operator (based on the values of  $\alpha$  and  $\beta$ , it can be a Dirichlet, Neumann or a mixed operator),  $\Omega$  is a bounded domain,  $\partial\Omega$  is the boundary of  $\Omega$  and  $d$  is the dimension of the problem. The numerical solution of any CDE is very challenging, when the convective part is dominant over the diffusion because in such cases the numerical approximations get contaminated due to the spurious oscillations and numerical diffusion. Traditional central difference based methods can be used for solving diffusion dominated problems, however, when the convective term is dominant, some accuracy has to be sacrificed to stabilize the numerical scheme.

RBFs are one of the important tools for the interpolation of sparse and scattered data points in multi-dimensions. The most popular and conditionally positive definite [1] RBFs have been listed in the Table 1, along with their order  $m$ . The RBF collocation method was initiated by Kansa [2] using globally supported interplant, which was popular as non-symmetric collocation scheme. Micchelli [1] proved that the Kansa's collocation matrix is invertible (i.e., the method is well posed) for any distinct set of scattered centers, in particular, with the Multi-Quadric(MQ), an infinitely smooth RBF. Another collocation method based on the Hermite - Brikhoff interpolation which is also known as the symmetric collocation scheme was proposed by Fasshauer [3] and Wu [4]. The required theoretical justification for the well posed criteria of these schemes is given by Wu [4] and Shaback [5]. Due to globally supported RBFs, the collocation matrices are highly dense and ill-conditioned which is also may increase with the number of centers. Recently Shu[6], Wright and Fornberg [7] and Chandini and Sanyasiraju [8,9,10] proposed some RBF-based *local* schemes, by sacrificing the spectral accuracy which is inherent in the global collocation schemes. However, these schemes produce better conditioned linear systems and also more flexible in handling the non-linearities.

TABLE 1: THE STRICTLY CONDITIONALLY POSITIVE DEFINITE RBFs

| Name of the RBF            | $\phi(r)$   | Order of the RBF   |
|----------------------------|---|--|
| Multiquadric (MQ)          | $(1 + (\epsilon r)^2)^\nu, \nu > 0, \nu \in \mathbb{N}$   | $m = \lfloor \nu \rfloor$  |
| Inverse Multiquadric (IMQ) | $(1 + (\epsilon r)^2)^\nu, \nu < 0$   | $m = 0$  |
| Gaussian                   | $e^{-(\epsilon r)^2}$   | $m = 0$  |
| Polyharmonic splines       | $\begin{cases} r^\nu, & \nu > 0, \text{ if } \nu \in 2\mathbb{N} - 1 \\ r^\nu \log(r), & \text{ if } \nu \in 2\mathbb{N} \end{cases}$ | $m = \begin{cases} \lfloor \nu/2 \rfloor \\ \frac{\nu}{2} + 1 \end{cases}$ |

Franke [11] proved numerically that the Hardy's MultiQuadric (MQ) [12] provides the best approximation when compared with the other methods tested, including other RBFs like Polyharmonic splines and Gaussian. However for MQ, the accuracy of the scheme depends highly on the shape parameter ( $\epsilon$ ). A large shape parameter helps in generating a well conditioned system; however, the corresponding approximation using the RBF becomes poor. If one chooses to use a small shape parameter, the RBF approximation becomes accurate, but the system matrix becomes severely ill-conditioned. That is, if the shape parameter becomes too small the resultant ill-conditioned system matrix causes errors in floating point arithmetic, which in turn affects the accuracy of the scheme. This *trade-off* between the accuracy and conditioning of the scheme is known as *Uncertainty Principle* [13], which is the main drawback of the global RBF collocation [2]. Therefore, to produce accurate and reliable solutions, the shape parameter of the infinitely smooth radial functions must be made optimum.

For the interpolation with scattered data, there is some work on the optimal choice of the shape parameter based on the number of centers and the distance between them. Hardy [12] proposed to use  $\varepsilon = 1/(0.815 d)$ , where  $d = \left(\frac{1}{N}\sum_{i=1}^N d_i\right)$ ,  $N$  is the number of centers and  $d_i$  is the distance from the center  $\underline{x}_i$  to its nearest neighbor. Franke [11] suggested  $\varepsilon = 0.8 (\sqrt{N}/D)$ , where  $D$  is the diameter of the minimal circle enclosing all data points. In [2] Kansa used exponentially

varying shape parameters,  $\varepsilon_i^2 = (\varepsilon_{min})^2 \left(\frac{(\varepsilon_{max})^2}{(\varepsilon_{min})^2}\right)^{\frac{(i-1)}{(N-1)}}$ ,  $i = 1, 2, \dots, N$ , where  $\varepsilon_{min}$

and  $\varepsilon_{max}$  are the input parameters. Rippa [14], for interpolation using RBF functions based on global collocation, proposed a “leave-one-out” cross-validation (LOOCV) algorithm, according to which the shape parameter should depend on the number of centers, distribution of centers, RBF interpolant ( $\phi$ ), conditional number of the matrix and finally the machine precision. Kansa [15], Fornberg [16] proposed to use spatially variable shape parameters to improve the accuracy in MQ-RBF interpolation instead of the constant shape parameter. Fasshauer [17] extended the LOOCV algorithm for finding the optimal shape parameter using pseudo-spectral (PS) methods to solve elliptic type of equations. Roque [18] experimented the LOOCV algorithm for finding the optimal shape parameter in global RBF collocation to solve partial differential equations. The global optimization techniques may be good for the problems having smooth solutions but inaccurate in capturing boundary layer solutions. They also suffer from the ill-conditioning as the number of centers increase. For the problems like CDEs, special attention has to be paid on the optimization of the shape parameter and also on the numerical scheme in order to get accurate and stable solutions.

In the present work global and local optimization schemes based on local RBF (LRBF) collocation have been proposed to find an (near) optimal shape parameter for solving Convection-Diffusion Equations (CDE). The algorithm is based on the re-construction of the forcing term in CDE using collocation over the centers in the local support domain. To realize this, Rippa's [14] leave one out cross validation algorithm has been used to calculate the residual errors. The (near) optimal shape parameter is obtained by minimizing the *local* Cost function at each center (node) of the computational domain. For most of the diffusion dominated problems, the (*global*) Cost function of the proposed algorithm is ideally imitates the (R.M.S) error function. However, for the convection dominated problems the algorithm produces a larger shape parameter in the regions where the solution varies rapidly and a small value elsewhere. This variable shape parameter makes the LRBF scheme to produce stable and accurate solutions for solving CDEs. Another most important feature of the developed local optimization schemes is the oscillation free nature of its cost and RMS error functions unlike their global counterpart. The proposed optimization algorithm with LRBF scheme is tested over linear CDEs in one and two-dimensions with strong boundary layer solutions.

Rest of the paper is organized as follows. In Section 2, the derivation of the RBF grid-free *local* (LRBF) scheme has been presented. In Section 3, the proposed *local* optimization algorithm to optimize the shape parameter has been described. In Section 4, first the implementation of the proposed optimization in over local collocation has been presented. After demonstrating the oscillations in the optimization based on the global collocation, the optimization schemes based on the local collocation have been validated by applying them to solve several CDE. Finally, some concluding remarks are made in the last section.

## 2. Development of grid-free local scheme based on RBF's

The local RBF (LRBF) grid-free scheme has been developed by Chandini and Sanyasiraju [8] and it is tested for the steady convection diffusion type problems and unsteady incompressible

viscous flow problems [9]. In this section, the LRBF scheme is discussed briefly, for the sake of continuity.

Let  $\mathcal{L}$  be a linear convection-diffusion operator,  $N$  is the total number of centers. If  $\mathcal{L}u_i$ , at any center  $\underline{x}_i$ , is expressed as a linear combination of the function values  $u$  at  $n_i$  ( $\ll N$ ) centers  $\mathcal{C}_i = \{\underline{x}_1, \dots, \underline{x}_{n_i}\}$  in the neighbourhood of  $\underline{x}_i$ , then it is given by

$$\mathcal{L}u(\underline{x}_i) \approx \sum_{j=1}^{n_i} c_j u(\underline{x}_j), \text{ for each } \underline{x}_i \in \Omega \quad (2.1)$$

where  $j$  is the local index and  $i$  is the global index. Then the computation of the weights  $c_j$  gives the required approximation. If  $s(\underline{x})$  is the interpolant

$$u(\underline{x}) \approx s(\underline{x}) = \sum_{j=1}^{n_i} \lambda_j \phi(\|\underline{x} - \underline{x}_j\|, \varepsilon) + \sum_{j=1}^l \gamma_j p_j(\underline{x}) \quad (2.2)$$

that interpolates the data,  $\{\underline{x}_j, u(\underline{x}_j)\}$ ,  $j = 1, 2, \dots, n_i$ . In (2.2),  $\|\cdot\|$  is the Euclidean norm,  $\varepsilon$  is a shape parameter,  $\{p_j(\underline{x})\}$ ,  $j = 1, \dots, l$ , is the basis for  $\Pi_{m-1}^d$  (space of all  $d$ -variate polynomials with degree  $\leq m-1$ , where  $m$  is the order of  $\phi$ ) and its inclusion may be necessary to guarantee the wellposedness of the interpolation problem [1]. (2.2) at  $\underline{x} = \underline{x}_j$ ,  $j = 1, 2, \dots, n_i$  gives  $n_i$  conditions for the interpolation problem, to take care of the extra degrees of freedom,  $l$  extra conditions are chosen by taking the coefficient vector  $\lambda \in \mathbb{R}^{n_i}$  orthogonal to  $\Pi_{m-1}^d|_{\mathcal{C}_i}$  (the polynomial space restricted to  $\mathcal{C}_i = \{\underline{x}_1, \dots, \underline{x}_{n_i}\}$ , i.e.,

$$\sum_{j=1}^{n_i} \lambda_j p_i(\underline{x}_j) = 0, \quad i = 1, 2, \dots, l \quad (2.3)$$

Imposing the interpolation conditions  $u(\underline{x}_j) = s(\underline{x}_j)$  and the orthogonality conditions (2.3) on  $s(\underline{x})$  gives a linear system,

$$\begin{bmatrix} \Phi & P \\ P^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \bar{\lambda} \\ \bar{\gamma} \end{bmatrix} = \begin{bmatrix} \mathbf{u} \\ \mathbf{0} \end{bmatrix} \quad (2.4)$$

where  $\Phi := \phi(\|\underline{x}_i - \underline{x}_j\|, \varepsilon)$ ,  $i, j = 1, 2, \dots, n_i$  and  $P := p_j(\underline{x}_i)$ ,  $j = 1, 2, \dots, l$ ,  $i = 1, 2, \dots, n_i$ .

Denote the coefficient matrix in (2.4) as  $A$  for any future reference. To compute the weights  $c_j$ , also consider the Lagrange representation of RBF interpolant (2.2), for  $n_i$  nodes, given by

$$u(\underline{x}) \approx s(\underline{x}) = \sum_{j=1}^{n_i} \psi_j(\underline{x}) u(\underline{x}_j) \quad (2.5)$$

where  $\psi_j(\underline{x})$ 's satisfies the cardinal conditions

$$\psi_j(\underline{x}_k) = \delta_{j,k}, \quad j, k = 1, 2, \dots, n_i \quad (2.6)$$

The closed form representation for  $\psi_j(\underline{x})$ , in terms of  $\phi(\|\underline{x} - \underline{x}_j\|, \varepsilon)$ 's by constructing a set of RBF interpolation problems for which the data is obtained from the cardinal conditions on  $\psi_j$ , is given by

$$\psi_j(\underline{x}) = \frac{\det(A_j(\underline{x}))}{\det(A)} \quad (2.7)$$

where  $A_j(\underline{x})$  is same as the matrix  $A$  with size  $(n_i + l)$ , except that  $j$ th row is replaced by the vector

$$B(\underline{x}) = [\phi(\|\underline{x} - \underline{x}_1\|, \varepsilon), \dots, \phi(\|\underline{x} - \underline{x}_{n_i}\|, \varepsilon) | p_1(\underline{x}), \dots, p_l(\underline{x})] \tag{2.8}$$

The representation derived through (2.5)-(2.7) can be used to approximate the derivatives of a function or in general, values of  $\mathcal{L}u$  at a given set of centers, say  $\underline{x}_i$ . Applying the operator  $\mathcal{L}$  to the Lagrange representation of RBF interpolant (2.5) gives

$$\mathcal{L}u(\underline{x}_i) \approx \mathcal{L}s(\underline{x}_i) = \sum_{j=1}^{n_i} \mathcal{L}\psi_j(\underline{x}_i)u(\underline{x}_j) \tag{2.9}$$

Comparing equations (2.1) and (2.9), gives

$$c_j = \mathcal{L}\psi_j(\underline{x}_i) = \mathcal{L} \frac{\det(A_j(\underline{x}_i))}{\det(A)}, j = 1, 2, \dots, n_i \tag{2.10}$$

If  $c_j$  are given by (2.10), then using the Cramer's rule backwards they can also be obtained by solving the linear system,

$$A[\underline{c} / \bar{y}] = (\mathcal{L}B(\underline{x}_i))^T, \text{ for each } \underline{x}_i \in \Omega \tag{2.11}$$

where  $A$  is a matrix given in (2.4),  $B(\underline{x})$  is a vector given in (2.8) and  $\underline{c}$  is a vector of weights. We apply a similar procedure to discretize the boundary operator  $\mathcal{B}$ . For any center  $\underline{x}_i \in \Omega \cup \partial\Omega$ , the system (2.11) is of size only  $n_i + l$  and can be solved using any direct method say, Gauss elimination. It is clear from the development of the final linear system (2.11) that, though it is dense, the size  $(n_i + l)$  is very small, that makes the system more stable for a wide range of  $\varepsilon$ . Further, only the right-hand side of (2.11) depends on the operator  $\mathcal{L}$ , for which the weights are to be computed. This optimizes the computation, if the weights have to be computed for many operators with the same distribution of the centers, as in the case of non-linear equations.

### 3. Local algorithm to optimize the shape parameter

It has been observed that, the optimized shape parameter depends on the differential operator, forcing term, boundary conditions, in addition to RBF interpolant ( $\phi$ ), the number of centers and their distribution. In this section, an algorithm, which can be used as a *global* or a *local* scheme and also free from ill-conditioning, has been proposed to optimize the shape parameter of the infinitely smooth radial basis functions. The proposed algorithm also satisfies the above stated requirements. It is clear from the development of the LRBF scheme, presented in the earlier section, that when collocation is used at any center  $\underline{x}_i \in \Omega$  over a local support domain  $\mathcal{C}_i$  for  $i = 1, 2, \dots, N$ , the collocation conditions are,  $\{\underline{x}_j, f(\underline{x}_j) = \mathcal{L}u(\underline{x}_j)\}, j = 1, 2, \dots, n_i$ .

Let  $\mathcal{C}_i^{(k)} = (\underline{x}_1, \dots, \underline{x}_{(k-1)}, \underline{x}_{(k+1)}, \dots, \underline{x}_{n_i})$   
 and  $f_i^{(k)} = \{f_i(\underline{x}_1), \dots, f_i(\underline{x}_{(k-1)}), f_i(\underline{x}_{(k+1)}), \dots, f_i(\underline{x}_{n_i})\}$  (3.1)

be the set of data points and their function values  $f$ , respectively, after removing the  $k^{th}$  center and its function value from the support  $\mathcal{C}_i$ . Then the forcing term  $f$  can be re-constructed, locally over  $\mathcal{C}_i^{(k)}$  using the collocation, as

$$f_i^{(k)} \approx s^{(k)}(\underline{x}) = \sum_{j=1(j \neq k)}^N (\mu_{i,j})^{(k)} \mathcal{L}^2 \phi(\|\underline{x} - \underline{x}_j\|, \varepsilon) + \sum_{j=1(j \neq k)}^l (Y_{i,j})^{(k)} p_j(\underline{x}) \tag{3.2}$$

where  $\mathcal{L}^2$  is a linear differential operator applied on RBF as a function of the second argument(center). The second term in the R.H.S of the equation (3.2) is necessary to guarantee the

non-singularity of the collocation matrix and the extra  $l$  conditions are chosen by taking the coefficient vector  $\mu_i \in \mathbb{R}^{n_i}$  orthogonal to  $\Pi_{m-1}^d|_{(C_i)^{(k)}}$  i.e.,

$$\sum_{j=1(j \neq k)}^{n_i} (\mu_{i,j})^{(k)} p_i(\underline{x}_j) = 0, \quad i = 1, 2, \dots, l \quad (3.3)$$

The coefficients,  $\mu$  and  $\gamma$ , in the equation (3.2) can be obtained by imposing the collocation conditions (3.1) and the orthogonality conditions (3.3). If  $f_i(\underline{x}_k)$  is the function value at the point that has been removed from  $C_i$  and  $f_i^{(k)}(\underline{x}_k)$  is the corresponding re-constructed value using (3.2) then the residual error,  $r_{i,k}$ , can be computed using

$$r_{i,k}(\varepsilon) = f_i(\underline{x}_k) - f_i^{(k)}(\underline{x}_k), \quad k = 1, 2, \dots, n_i \quad (3.4)$$

For each center  $\underline{x}_i \in \Omega$ , (3.4) gives residual errors for  $k = 1, 2, \dots, n_i$ , therefore (3.4) requires  $O(n_i)^4$  operations for each center. To reduce the computational complexity, the Rippa's [14] formula (been developed for interpolation problems)

$$r_{i,k}(\varepsilon) = \frac{\mu_{i,k}}{(A_{\mathcal{L}}^{-1})_{kk}}, \quad k = 1, 2, \dots, n_i, \quad \text{each } \underline{x}_i \in \Omega \quad (3.5)$$

where  $\mu_{i,k}$  is the  $k^{\text{th}}$  component in the vector  $\mu_i$  and  $(A_{\mathcal{L}}^{-1})_{kk}$  is the  $k^{\text{th}}$  diagonal element of the inverse of the collocation matrix

$$A_{\mathcal{L}} = \begin{bmatrix} \mathcal{L}^2 \Phi & P \\ P^T & \mathbf{0} \end{bmatrix} \quad (3.6)$$

based on the data, over  $C_i$  has been used. Since (3.5) requires the inversion only once, the computational complexity is now  $O(n_i)^3$  only. By adopting a similar procedure for the boundary operator,  $\mathcal{B}$ , the residual error is computed using

$$r_{i,k}(\varepsilon) = \frac{\mu_{i,k}}{(A_{\mathcal{B}}^{-1})_{kk}}, \quad k = 1, 2, \dots, n_i, \quad \text{each } \underline{x}_i \in \partial\Omega \quad (3.7)$$

Equations (3.5) and (3.7) can now be used to calculate the (global) or (local) optimal shape parameter in the following way.

### 3.1. Global optimal shape parameter

Using equations (3.5) and (3.7), define the (*global*) Cost function, that measures the quality of fit for the global collocation, as

$$\text{global cost}(\varepsilon) = \left( \frac{1}{N} \sum_{i=1}^N \|R_i\|_2^2 \right)^{1/2} \quad (3.8)$$

where the vector  $R_i = \frac{r_{i,k}(\varepsilon)}{n_i}$ ,  $k = 1, 2, \dots, n_i$  for each  $i$  and  $N$  is the total number of centers in the domain. Then the (near) global optimal shape parameter is obtained by minimizing the (global) Cost function (3.8).

The (global) Cost function obtained using (3.8) is different from such global Cost functions exist in the literature (because they are obtained using global collocation). It is also free from the ill-conditioning of the collocation systems. To compare the (global) Cost function with the (RMS) error function, for different values of the shape parameter  $\varepsilon$  in the neighborhood of optimal shape parameter, test problems are solved using LRBF method and the corresponding RMS error is calculated using

$$\text{RMS error}(\varepsilon) = \left( \frac{1}{N} \sum_{i=1}^N |\text{Analytical solution}(i) - \text{LRBF solution}(i)| \right)^{1/2} \quad (3.9)$$

For diffusion dominated problems, the (global) Cost function ideally imitated the (RMS) error function, therefore, the (near) global optimal shape parameter obtained by minimizing the (global) Cost function is very ideal. However, for the convection dominated problems, the residual errors, in general, are very large in the boundary layer regions when compared to the other smoother regions, therefore, for these problems, the (global) Cost function may not match with the (RMS) error function. That means, there may not be a global optimum shape parameter which can minimize the RMS error for these problems.

That necessitates, for the convection dominated problems, an alternate algorithm which may generate a variable (local) optimum shape parameter but satisfy the requirements stated at the beginning of this section.

### 3.2. Local optimized variable shape parameter

Define a (local) Cost function, which measures the quality of fit for the local collocation, as

$$\text{local cost}(\varepsilon_i) = \|R_i\|_2 \quad (3.10)$$

where the vector  $R_i = \frac{r_{i,k}(\varepsilon)}{n_i}$ ,  $k = 1, 2, \dots, n_i$  and  $r_{i,k}(\varepsilon)$  is as given in equations (3.5), (3.7) for every  $\underline{x}_i \in \Omega$  or  $\underline{x}_i \in \partial\Omega$ , respectively. In this process, the (near) optimal shape parameter is calculated by minimizing the (local) Cost function at each center of the computational domain locally and the same will be used in the LRBF scheme.

The size of the coefficient matrices,  $A_L$ ,  $A_B$ , in the equations (3.5), (3.7) is  $(n_i + 1)$  very small, when compared to the size  $(N + 1)$  of the global collocation matrix, which makes the system more stable for wide range of shape parameters. The routine function *Brent* has been used from the Numerical Recipes [19] to carry out the optimization. The implementation of the developed *local* optimization algorithm along with LRBF scheme, for solving CDEs is presented in the next Section.

## 4. Numerical implementation

In this section, the RBF grids free *local* scheme with the proposed optimization algorithms has been tested over several Convection-Diffusion Equations (CDE), using the following algorithm

### 4.1. Algorithm

- Choose a reference centre  $\underline{x}_i \in \Omega \cup \partial\Omega$  and for every such reference center, select  $C_i = \{\underline{x}_1, \dots, \underline{x}_{n_i}\}$  in the neighbourhood of  $\underline{x}_i$ , such that the reference center  $\underline{x}_i$  is always part of the local support domain, i.e.,  $\underline{x}_i \in C_i$ .
- Find the (near) optimal shape parameter, using the optimization algorithm, described in the Section 3 (the global optimization gives a single  $\varepsilon$  & the local optimization gives a variable  $\varepsilon$ ).
- Calculate the weights  $c_j$ ,  $j = 1, 2, \dots, n_i$  (for every reference centre  $\underline{x}_i$ ) by solving the local linear system (2.11), using the optimum shape parameter obtained in the earlier step.
- Assemble the weights of (2.11) at proper locations of the  $i^{\text{th}}$  row, known from the centers belong to  $C_i$ , to form the global system

$$\mathbf{L}u^* = \mathbf{f} + \mathbf{b} \quad (4.1)$$

where  $\mathbf{L}$  is the matrix containing the weights for  $\mathcal{L}$ ,  $\mathcal{B}$  at  $C_i$  and the vector  $\mathbf{b}$  contains the contributions from the boundary  $\partial\Omega$ . The matrix  $\mathbf{L}$  is sparse and its sparseness depends on the centers chosen in  $C_i$ . It is also non-symmetric.

- Finally, solve the global system (4.1) using any iterative method (for most of the iterative solvers explicit generation of the global system may not be needed).

## 4.2. Validation

One and two-dimensional CDEs, with known analytical solutions are chosen, for testing the developed algorithm. The forcing term and Dirichlet boundary conditions have been calculated from the analytical solution wherever they are needed. All the computations have been carried out using MQ (with  $\nu = 1/2$ ) as the basis function, over uniformly distributed centers. Though the proposed scheme works with any scattered data set, the uniformity has been imposed on the set of centers for the sake of simplicity. The list of problems are:

### Example 1

$$u_x - a u_{xx} = a \pi^2 \sin(\pi x) + \pi \cos(\pi x), 0 < x < 1, u(0) = 0, u(1) = 1 \text{ with} \quad (4.2)$$

$$u(x) = \sin(\pi x) + \frac{e^{x/a} - 1}{e^{1/a} - 1} \quad (4.3)$$

### Example 2

$$u_x - a u_{xx} = -2a \pi^2 \cos(2\pi x) + \pi \sin(2\pi x) - e^{(x-1)/a}, 0 < x < 1 \quad (4.4)$$

$$u(0) = 0, u(1) = 1 \text{ with } u(x) = \sin^2(\pi x) + x e^{(x-1)/a} \quad (4.5)$$

### Example 3

$$\frac{1}{1+x} u_x - a u_{xx} = \left( \frac{1}{1+x} - a \right) e^x, 0 < x < 1 \text{ with} \quad (4.6)$$

$$u(0) = 1 + 2^{-1/a}, u(1) = e + 2, u(x) = e^x + 2^{-1/a} (1+x)^{(1+\frac{1}{a})} \quad (4.7)$$

### Example 4

$$u_x - a (u_{xx} + u_{yy}) = f(x, y), 0 < x, y < 1 \quad (4.8)$$

$$u(x, y) = e^{\frac{\pi}{a}} \sin(\pi y) \left( \frac{2 e^{-\frac{\pi}{a}} \sinh(\sigma x) + \sinh(\sigma(1-x))}{\sinh \sigma} \right), \text{ where } \sigma^2 = \pi^2 + a^{-2} \quad (4.9)$$

### Example 5

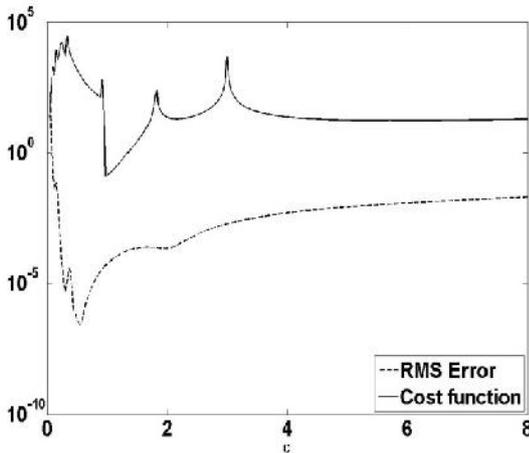
$$u_x + u_y - a (u_{xx} + u_{yy}) = f(x, y), 0 < x, y < 1 \quad (4.10)$$

$$u(x, y) = e^{-(1-x)(1-y)/2a} \quad (4.11)$$

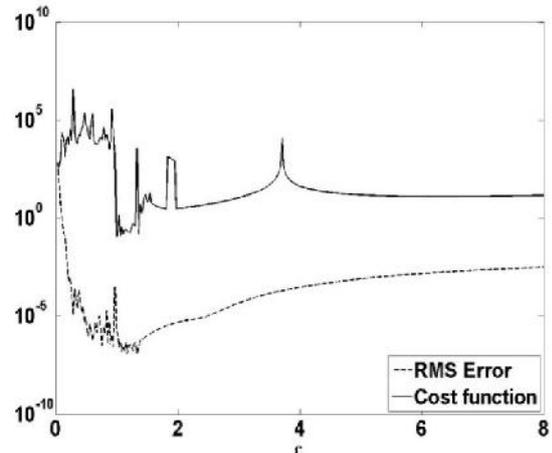
### Example 6

$$\frac{1}{1+y} u_y - a (u_{xx} + u_{yy}) = f(x, y), 0 < x, y < 1 \quad (4.12)$$

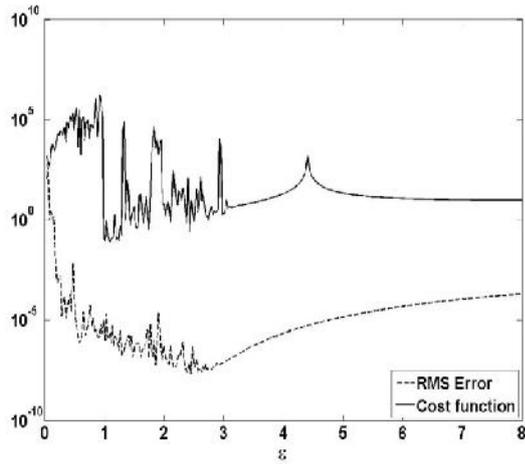
$$u(x, y) = e^{y-x} + 2^{-1/a} (1+y)^{(1+\frac{1}{a})} \quad (4.13)$$



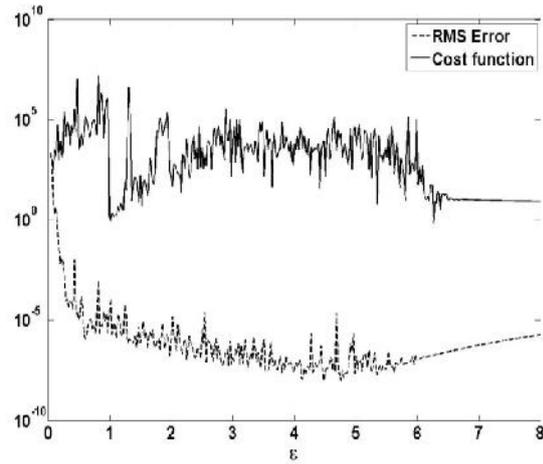
(a)  $N = 11$



(b)  $N = 21$

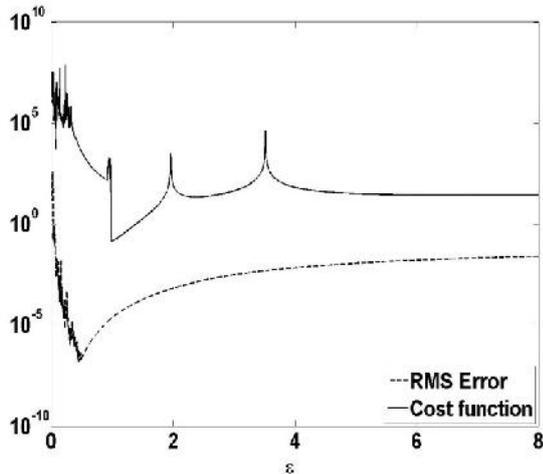


(c)  $N = 41$

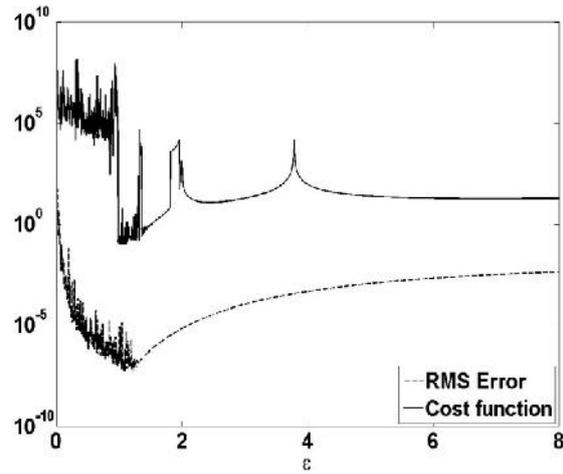


(d)  $N = 81$

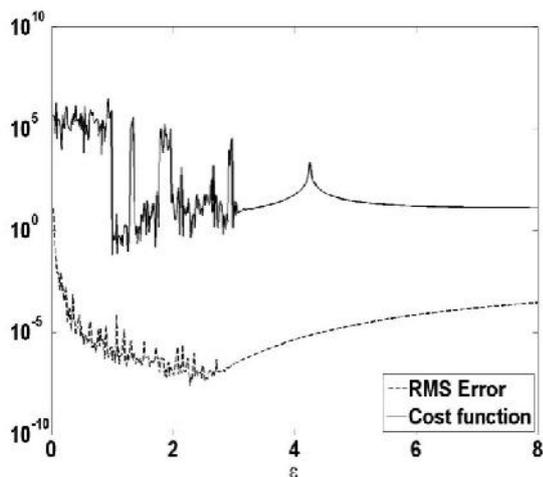
Figure 1. Comparison of (RMS) error function and Cost function against the Shape parameter, using Global Collocation, at  $\alpha = 1, N = 11 - 81$  for Example 1.



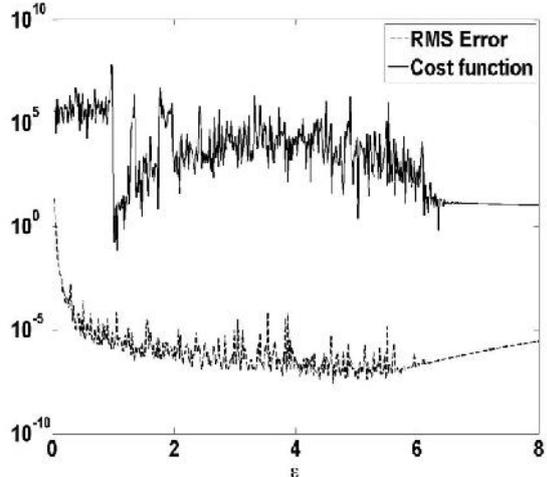
(a)  $N = 11$



(b)  $N = 21$



(c)  $N = 41$



(d)  $N = 81$

Figure 2. Comparison of (RMS) error function and Cost function against the Shape parameter, using Global Collocation, at  $\alpha = 1, N = 11 - 81$  for Example 3.

First, the one dimensional test problems 1 and 3 have been solved using the conventional global collocation scheme [2]. The RBF parameter  $\epsilon$  has been optimized using Rippa's (global) leave one out cross validation algorithm [18]. By varying the number of centers to discretize the domain with 11, 21, 41 and 81, the global cost function and the corresponding (RMS) error function ( here after referred as error function ) with respect to the shape parameter have been obtained and compared in the Figures 1 and 2 at  $\alpha = 1.0$ , for the problems 1 and 3, respectively. It is clear from these comparisons that, except with  $N = 11$ , the cost function and also the error function for larger  $N$  are very oscillatory. It has been noticed that the oscillations have increased very rapidly with the number of centers  $N$ . Therefore, minimizing these functions is highly complicated and often leads to spurious minimums. Further, it has been observed that, the amplitude and frequency of these oscillations are also increased with decrease in  $\alpha$ . To demonstrate this, the cost and error functions, by varying  $\alpha$ , are given in Figures 3 and 4 for  $N = 11$  and  $N = 81$ , respectively. It is clear from these figures that the cost and error functions are highly oscillatory at  $\alpha = 0.01$ . Since the Rippa's algorithm requires an initial interval of shape parameter to minimize the cost function, the optimum value is very much dependent on the initial interval and due to high oscillations the initial interval

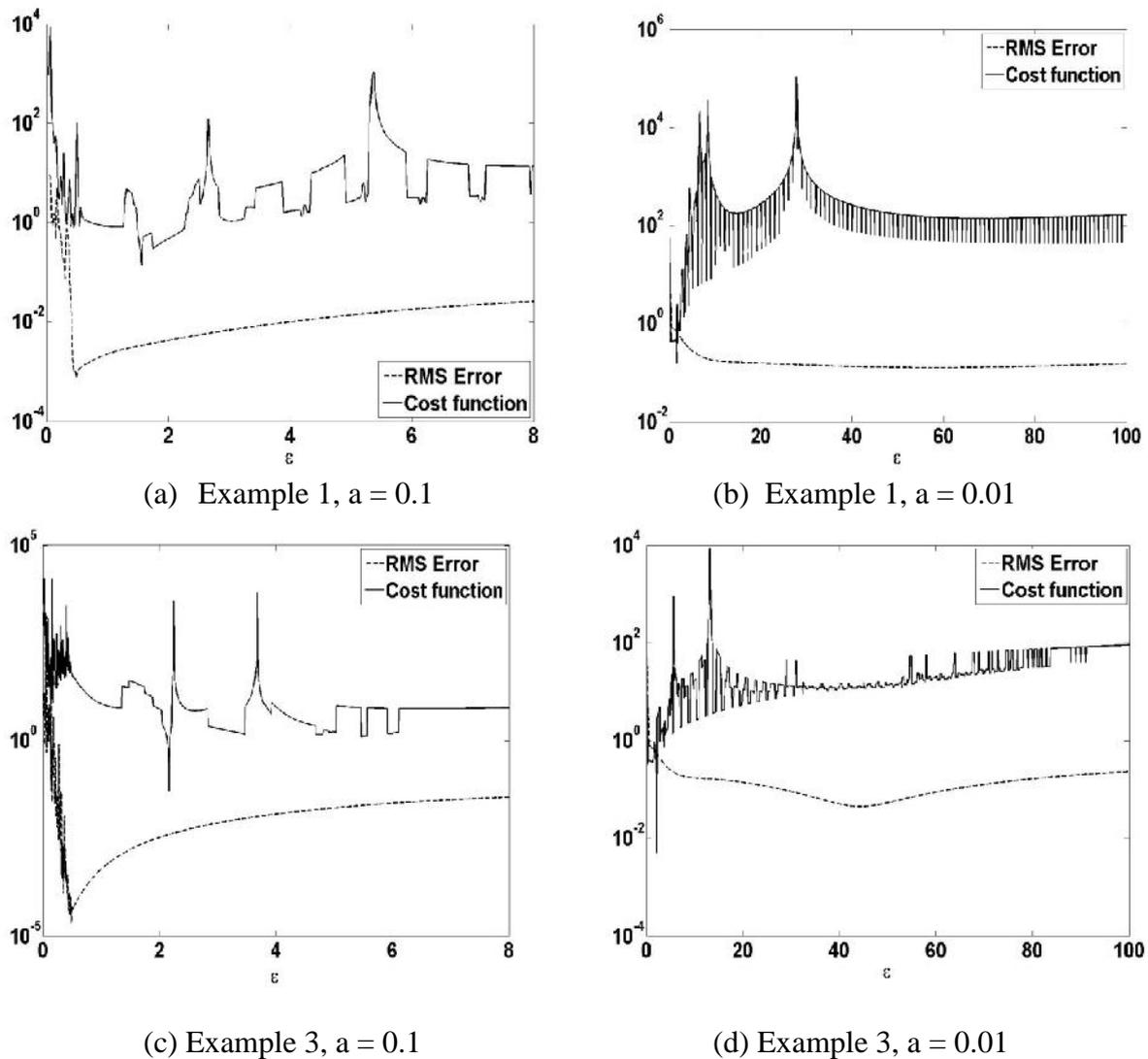


Figure 3. Comparison of (RMS) error function and Cost function against the Shape parameter, using Global Collocation, with  $N = 11$ .

must be very fine to avoid too many local minimums to group inside the chosen interval and spoil the optimization process. In other words, the oscillatory nature of these functions makes the optimization algorithm to be very ineffective and the corresponding RBF solution can be inaccurate. Looking at the plots of the cost function and error function, a very narrow interval of shape parameter has been chosen for test problem 1 and the corresponding error norms and optimum shape parameters have been reported in Table 2 for different values of  $\alpha$  with different values of  $N$ . It must be noted that the results presented, in this table, under ‘Global’ are obtained with the global optimization procedure for the shape parameter (after choosing a very narrow band of initial interval taken by looking at the graph of the cost function) and the results presented under ‘Best’ are the ones obtained without using the optimization algorithm but choosing a shape parameter at which the error function is minimum and finally the results presented under ‘Fixed’ are the ones obtained with  $\varepsilon = 1$ . The results presented in this table are obtained by manually looking at the graphs of cost and error functions therefore, cannot be reproduced for any general problem. However, these can be used as guideline values for the validation of the developed, in Section 3, local optimization procedure.

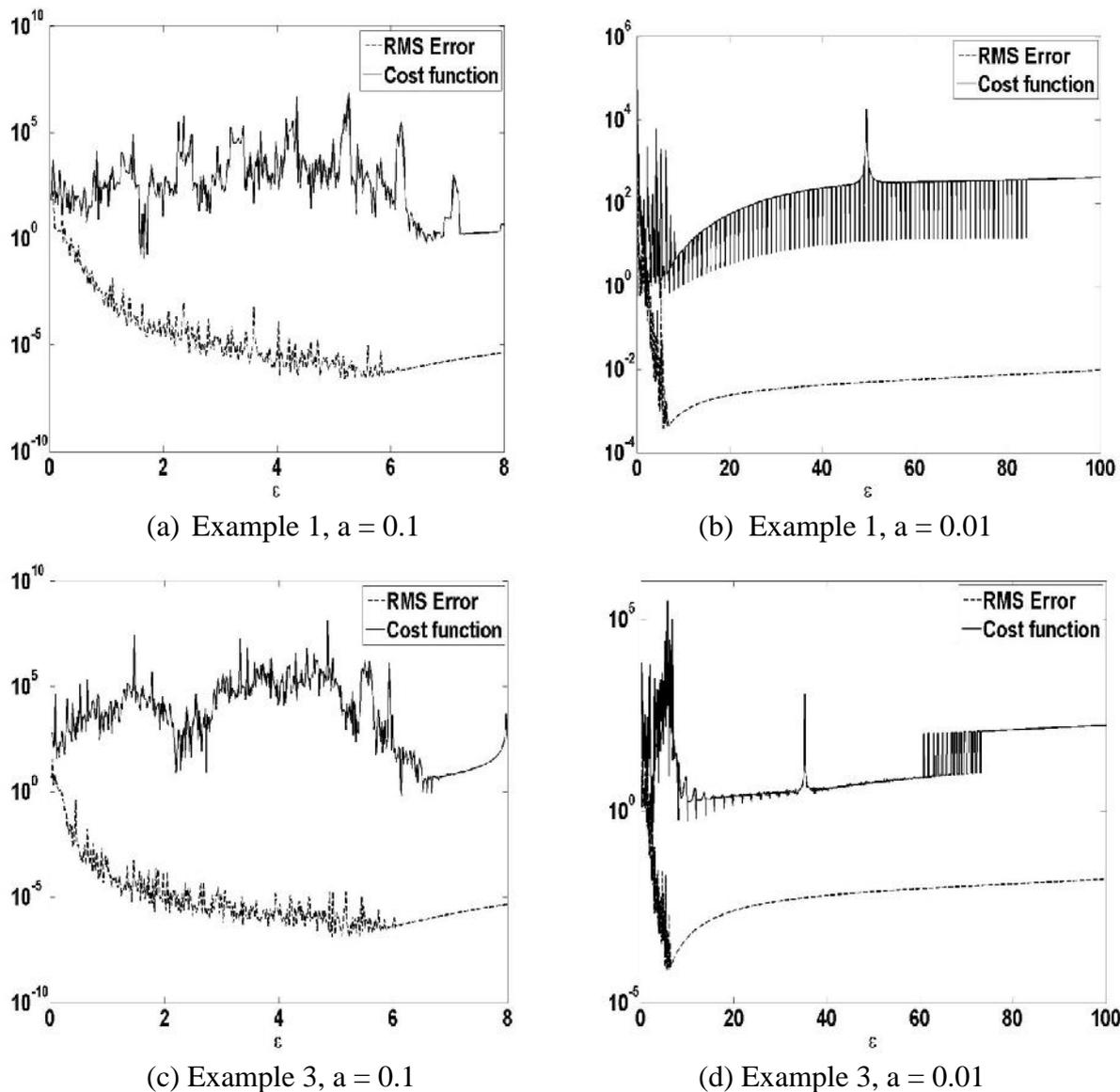


Figure 4. Comparison of (RMS) error function and Cost function against the Shape parameter, using Global Collocation, with  $N = 81$ .

TABLE 2: COMPARISON OF THE FIXED (Fixed), GLOBAL (Global), BEST (Best) SHAPE PARAMETERS AND THE CORRESPONDING RMS ERRORS FOR EXAMPLE 1, OBTAINED USING GLOBAL COLLOCATION WITH OPTIMIZATION.

| $N \rightarrow$ |        | Number of nodes ( $N$ ) |           |               |           |               |           |               |           |
|-----------------|--------|-------------------------|-----------|---------------|-----------|---------------|-----------|---------------|-----------|
|                 |        | 11                      |           | 21            |           | 41            |           | 81            |           |
| $a$             |        | $\varepsilon$           | RMS error | $\varepsilon$ | RMS error | $\varepsilon$ | RMS error | $\varepsilon$ | RMS error |
| 1.0             | Fixed  | 1.0                     | 3.25(-03) | 1.0           | 8.60(-04) | 1.0           | 2.18(-04) | 1.0           | 3.50(-05) |
|                 | Global | 0.9713                  | 4.86(-05) | 1.359         | 4.94(-07) | 1.085         | 3.71(-06) | 6.266         | 1.77(-07) |
|                 | Best   | 0.5453                  | 3.07(-07) | 1.336         | 4.73(-07) | 2.455         | 4.60(-07) | 4.486         | 4.65(-08) |
| 0.1             | Fixed  | 1.0                     | 3.40(-02) | 1.0           | 7.76(-03) | 1.0           | 1.90(-03) | 1.0           | 4.73(-04) |
|                 | Global | 1.565                   | 3.34(-03) | 1.587         | 4.80(-05) | 2.569         | 6.72(-05) | 1.656         | 3.31(-04) |
|                 | Best   | 0.4921                  | 1.08(-03) | 1.336         | 2.25(-05) | 2.569         | 6.72(-05) | 5.193         | 3.97(-07) |
| 0.01            | Fixed  | 1.0                     | 6.90(-01) | 1.0           | 4.34(-01) | 1.0           | 1.93(-01) | 1.0           | 5.57(-02) |
|                 | Global | 1.546                   | 6.70(-01) | 1.413         | 7.26(-02) | 2.946         | 3.12(-02) | 5.012         | 3.65(-03) |
|                 | Best   | 55.94                   | 1.28(-01) | 7.545         | 2.73(-02) | 3.279         | 7.57(-03) | 5.697         | 1.36(-03) |

Next, the test problems 1 to 6 have been used to test the developed local optimization schemes. Once again, the one dimensional (1D) domains have been discretised with 11, 21, 41 and 81 centers and two dimensional (2D) domains have been discretised with  $11 \times 11$ ,  $21 \times 21$ ,  $41 \times 41$  and  $81 \times 81$  centers. The diffusion parameter  $\alpha$  has been varied from 1 to 0.01 (that is, the Peclet number, which is the ratio of convection over diffusion, is equivalent to 1 to 100). To find the optimal shape parameter, the number of centers in the local support domain has been fixed as  $n_i = 4$  for one dimension and  $n_i = 7$  for two dimensional problems. However, for solving the CDE, the number of centers in the local support domain has been fixed as  $n_i = 3$  and  $n_i = 5$  for one and two dimensional problems, respectively.

Before looking at the optimized shape parameter, the (global) Cost function (3.8) and the error function (3.9), have been compared by varying the shape parameter  $\varepsilon$ . For the diffusion parameter  $\alpha = 1, 0.1$  and  $0.01$ , the comparisons for all the six problems have been presented in Figures 5, 6 and 7, respectively. To avoid any dependence of the solution on the discretization, the comparisons with coarser 11 (or  $11 \times 11$  for 2D) centers to finer 81 (or  $81 \times 81$  for 2D) centers have been included in all these comparisons. Unlike, the case of global collocation, the cost and error functions in this local case are non-oscillatory even for  $N = 81$  and  $\alpha = 0.01$ . Further, it is also clear from the Figure 5 that, for all the problems and for all the chosen cases, the (global) Cost function, may be due to the high smoothness of the solutions, more or less followed the same pattern of the error function. That means, the value of the RMS error at the optimum  $\varepsilon$  obtained by optimizing the (global) Cost function is expected to match with the minimum of the error function. This is similar to what has been reported in the literature, mostly for the elliptic Laplacian or Poisson equations. However, the same is not true if the diffusion parameter  $\alpha$  is reduced or the Peclet number  $Pe$  is increased from one. Looking at the Figures 6 and 7 in which the (global) Cost function and error functions for the test problems 1 to 6, have been compared for the diffusion parameters  $\alpha = 0.1$  and  $\alpha = 0.01$ , respectively, unlike the results known from the literature for pure diffusion problems, it is clear from these comparisons that the minimum value of the error function is not occurring at the optimum shape parameter (minimum of the (global) Cost function). In particular, at  $\alpha = 0.1$ , say for the test problem 2, the minimization of the (global) Cost function is giving an  $\varepsilon$  value less than 2 for all the chosen cases, however the minimum of the error function can be seen at close to  $\varepsilon = 3$ . Similarly, for the test problem 6, when the global optimum is near  $\varepsilon = 1$ , one can see a sharp fall in the error after  $\varepsilon = 3$  giving a large gap between the  $\varepsilon$  obtained from the global optimization and the best shape parameter (the value of the shape parameter where the error function is minimum, which is of course can be obtained if the analytical solution of CDE is known). A similar deviation of the optimized shape parameter and the best shape parameter can also be seen for the other test problems and also for the test problems at  $\alpha = 0.01$  which have been shown in the Figure 7. The deviation of optimized and best shape parameters is even more clear in

Figure 7 wherein the comparisons have been made for 2D examples with small  $N$  and  $\alpha = 0.01$ . All these comparisons demonstrate that the optimization of the (global) Cost function may not lead to a shape parameter at which the error function is minimum for CDE problems, particularly, at low diffusion parameters.

Alternatively, the optimization can be made local, as it is given in the local optimization algorithm (3.10) to produce a variable optimum shape parameter. In the local optimization, the optimal shape parameter varies spatially according to the nature of the solution and the corresponding forcing function. The value of the shape parameter in this case may be large where the solution varies rapidly and could be small elsewhere. This nature has been reported in the Figures 8 & 9 for the test problems 2 and 4.

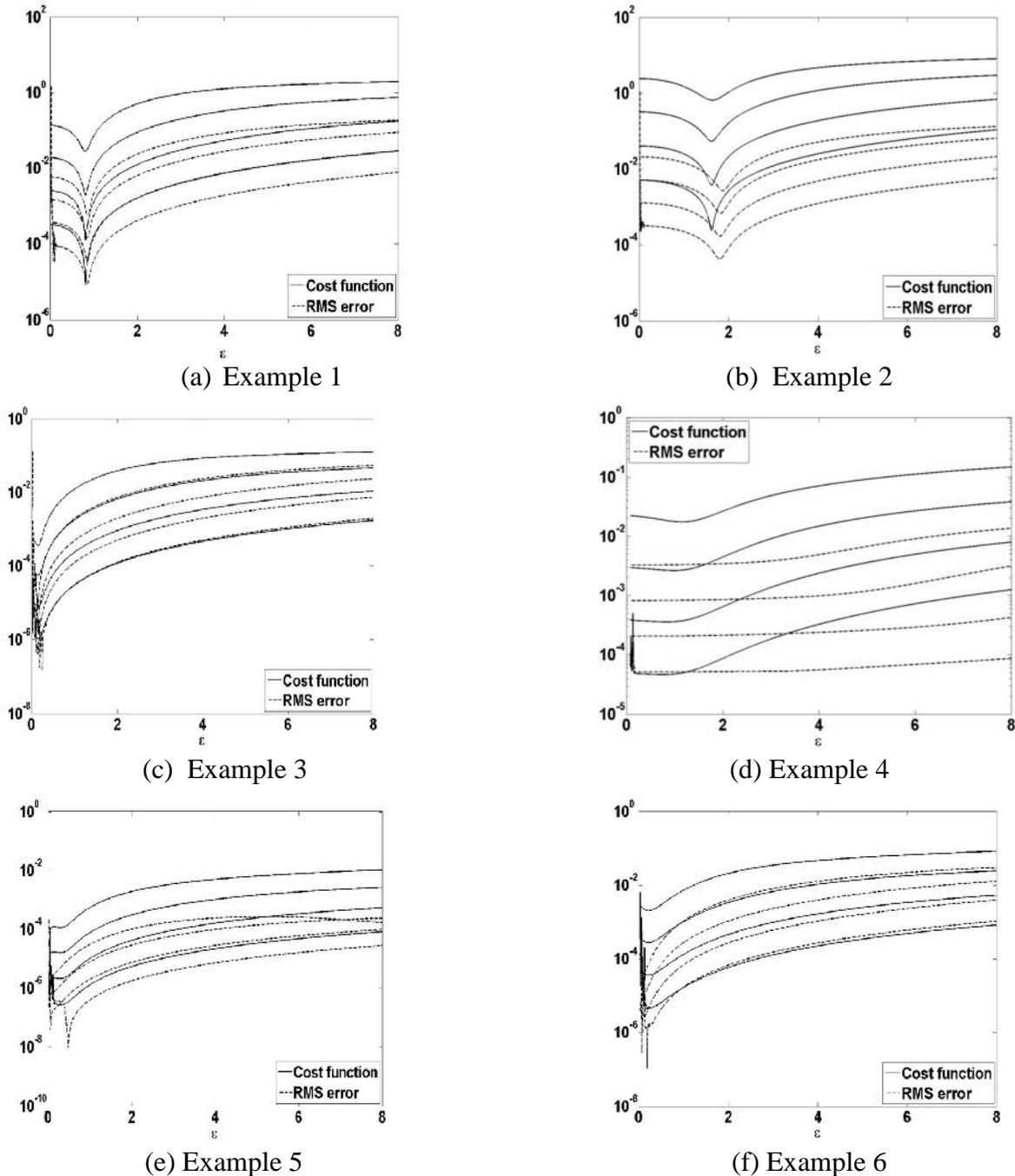
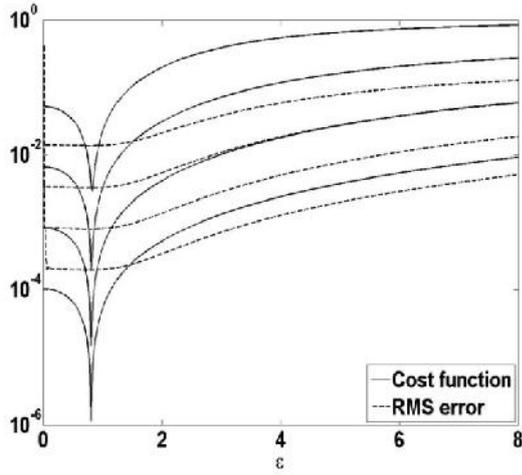
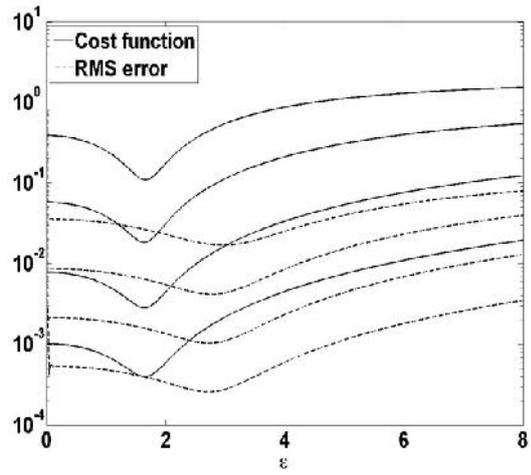


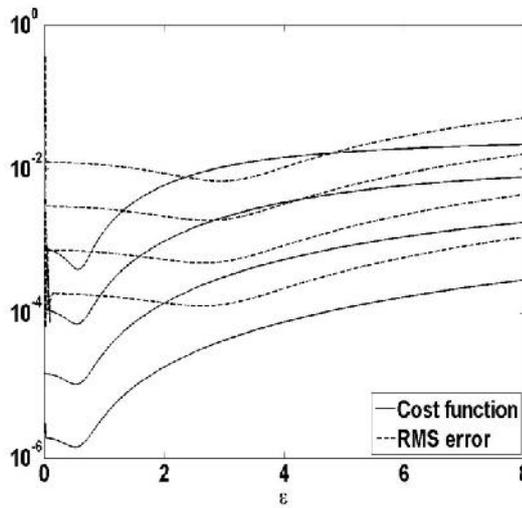
Figure 5. Comparison of (RMS) error function & (global) Cost function against the Shape parameter at  $\alpha = 1, N = 11, 21, 41, 81$  (from the top to bottom) for Examples 1–3 and  $N = 11 \times 11, 21 \times 21, 41 \times 41, 81 \times 81$  for Examples 4–6 (from the top to bottom).



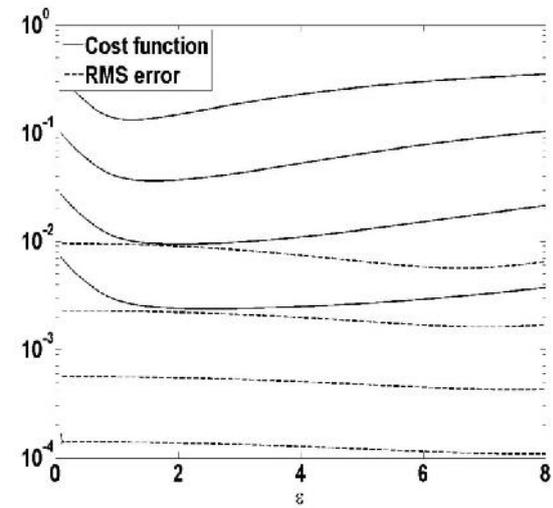
(a) Example 1



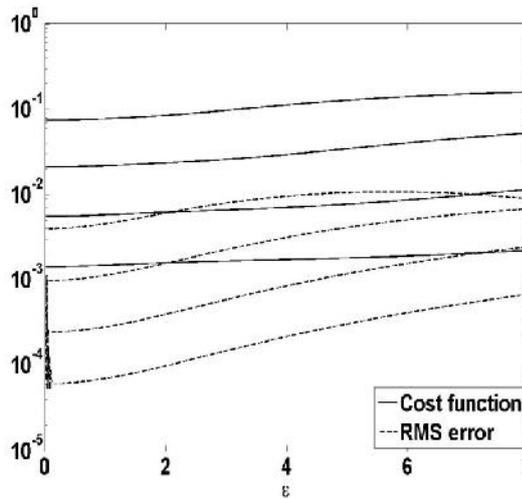
(b) Example 2



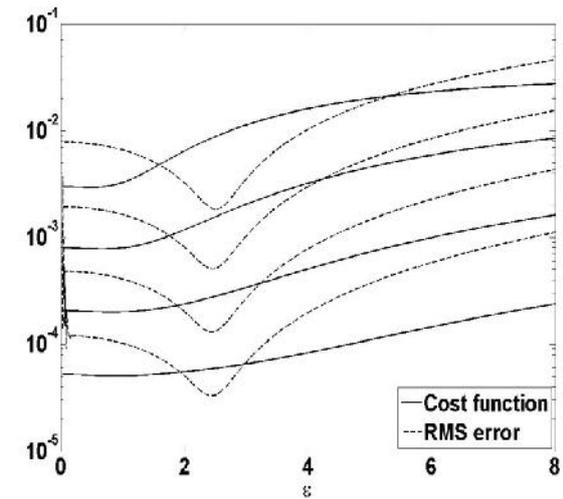
(c) Example 3



(d) Example 4

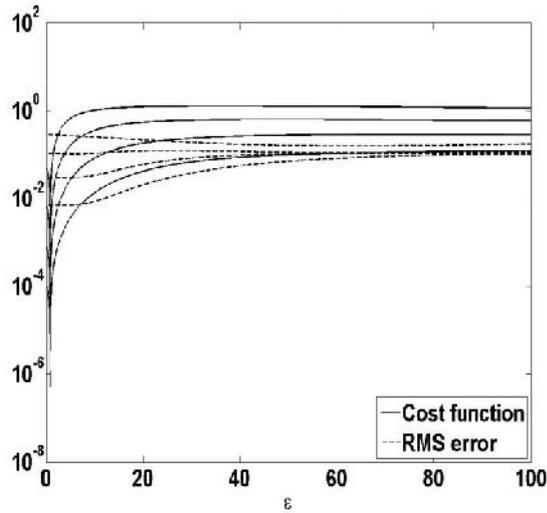


(e) Example 5

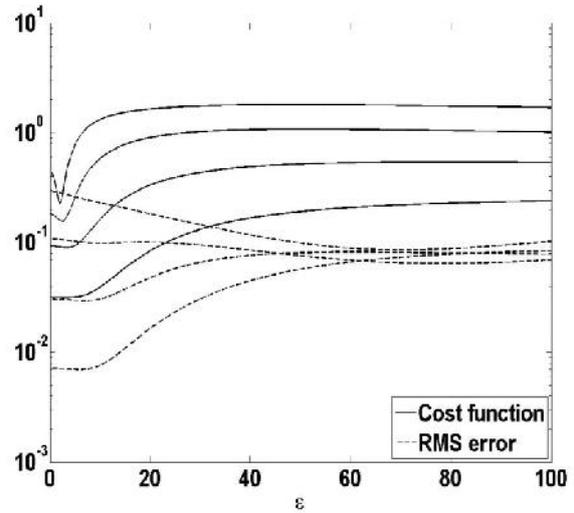


(f) Example 6

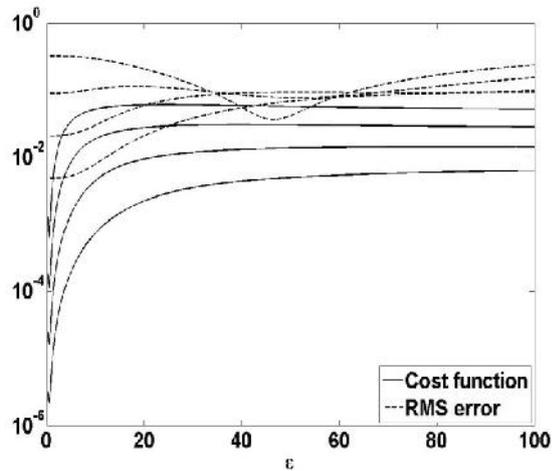
Figure 6. Comparison of (RMS) error function & (*global*) Cost function against the Shape parameter at  $a = 0.1, N = 11, 21, 41, 81$  (from the top to bottom) for Examples 1–3 and  $N = 11 \times 11, 21 \times 21, 41 \times 41, 81 \times 81$  for Examples 4–6 (from the top to bottom).



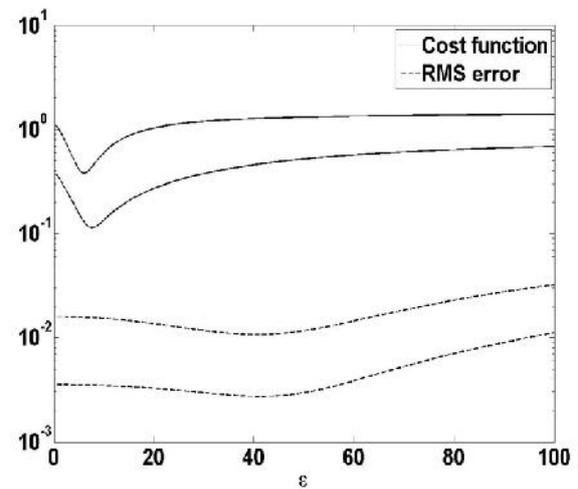
(a) Example 1



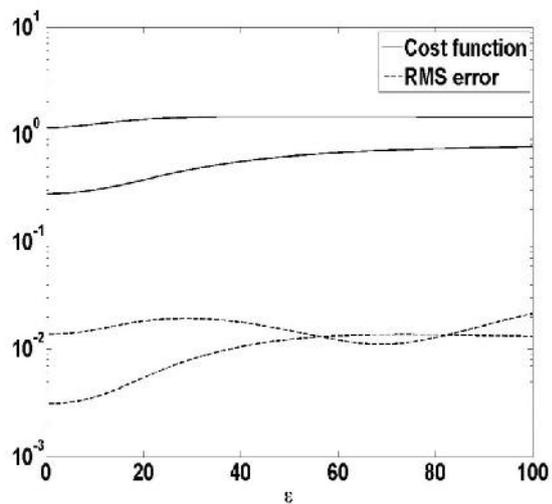
(b) Example 2



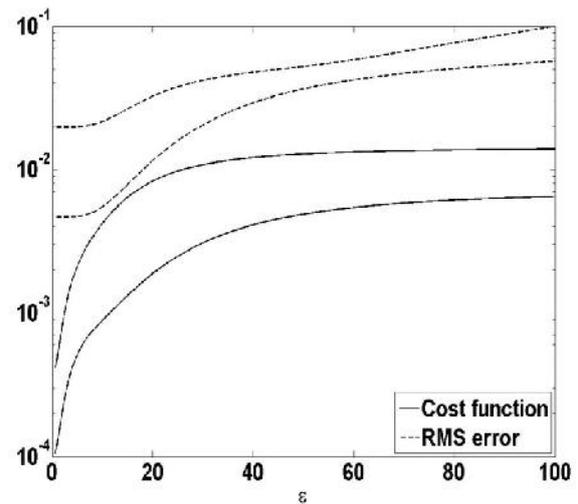
(c) Example 3



(d) Example 4

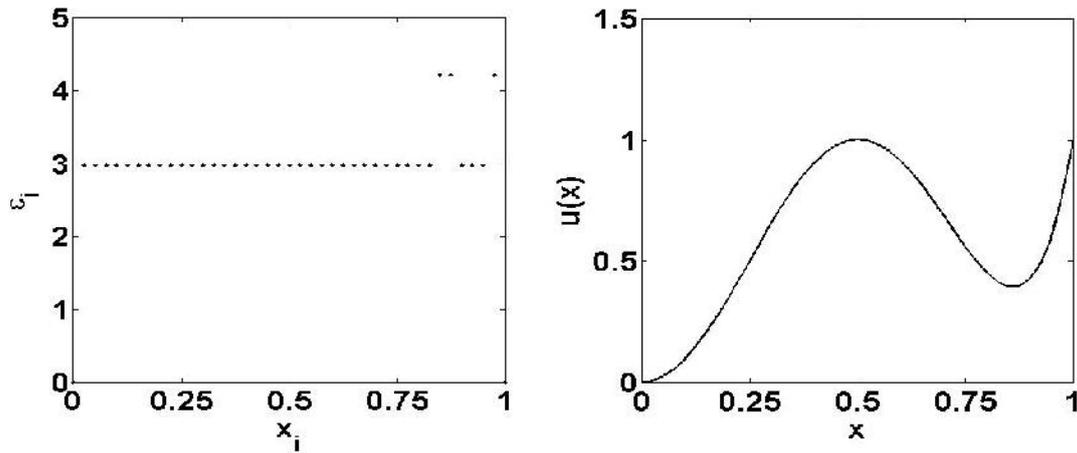


(e) Example 5

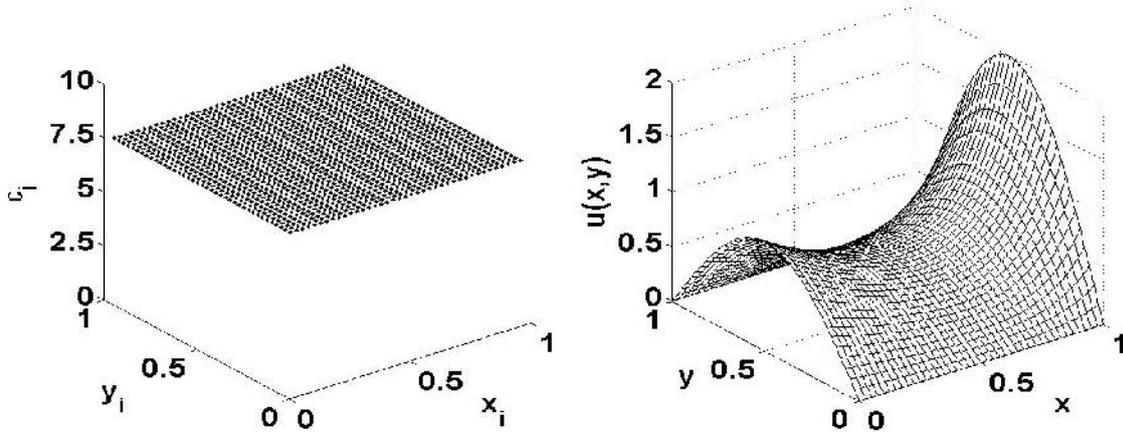


(f) Example 6

Figure 7. Comparison of (RMS) error function & (*global*) Cost function against the Shape parameter at  $\alpha = 0.01, N = 11, 21, 41, 81$  (from the top to bottom) for Examples 1–3 and  $N = 41 \times 41, 81 \times 81$  for Examples 4–6 (from the top to bottom).



(a) Example 2



(b) Example 4

Figure 8. Local optimal variable shape parameter ( $\epsilon_i$ ) (left) and the corresponding numerical solution ( $u$ )(right), for the Examples 2 & 4 (from top to bottom) at  $\alpha = 0.1$ ,  $N = 41$  (for 1D) and  $N = 41 \times 41$  (for 2D).

Due to the variable nature of the optimum shape parameter in this case, the comparison of the (local) Cost function and error function doesn't make any sense, therefore, one can compare the error obtained using the local variable shape parameter with the corresponding error at the best shape parameter. If this is more close to the error at the best shape parameter than the error at the global optimum shape parameter then the purpose of developing local optimization is justified at least for the CDE problems with low diffusion parameters. To demonstrate the same, the errors obtained using global and local optimization procedures are compared with the errors at the best shape parameter for all the six test problems in Tables 3, 4 & 5 for diffusion parameters  $\alpha = 1, 0.1 \text{ \& } 0.01$ , respectively. Once again the number centers in these computations are varied from 11 to 81 for 1D and  $11 \times 11$  to  $81 \times 81$  for 2D problems. For the sake of completeness, the computations at  $\alpha = 1$  are also performed and the corresponding error norms have been reported in the Table 3. It is clear from this table that, for many problems, the local optimization also produced a unique optimal shape parameter which is very close to the corresponding global optimum shape parameter. Therefore the minimum error obtained using these algorithms are one and same. We have reported the range (minimum and maximum) of the shape parameter for the problems wherein the local optimization has produced a variable shape parameter. It can be seen that the errors obtained with the local scheme are more close to the error at the best

shape parameter than the global scheme. One issue which must be noted here is the amount of improvement in the solution. Since the error at the best shape parameter is one of the best one can obtain, the improvement in the solution due to local optimization depends on the gradient of the error at the best shape parameter. That is, if the error curve for a particular problem has a marginal variation at the best shape parameter then the improvement in the solution also will be marginal and the improvement in the solution will be substantial only if the error curve has a good variation at the best shape parameter like, for example, for the test problem six in the Figure 5. Therefore, instead of looking at the improvement of the solution due to local scheme one should look at the obtained error whether it is close to the error at the best shape parameter than the corresponding error with the global shape parameter.

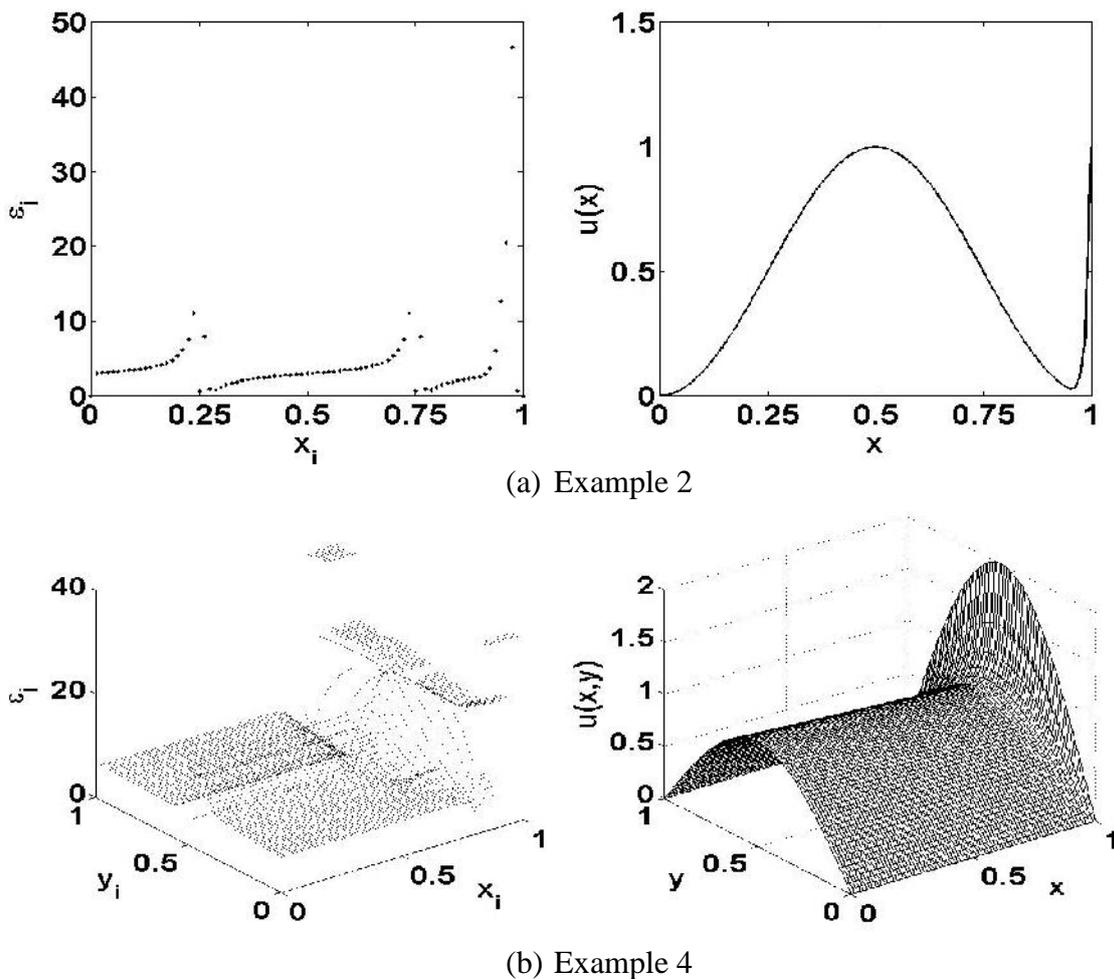


Figure 9. Local optimal variable shape parameter ( $\epsilon_i$ ) (left) and the corresponding numerical solution ( $u$ ) (right), for the Examples 2 & 4 (from top to bottom) at  $\alpha = 0.01$ ,  $N = 81$  (for 1D) and  $N = 81 \times 81$  (for 2D).

Similarly, for the diffusion parameter  $\alpha = 0.1$ , presented in Table 4, the RMS errors obtained for all the test problems using the local optimal shape parameters are better than the RMS error calculated with the global optimal shape parameter. In particular, once again for the test problem 6, the RMS errors calculated with the local optimal shape parameters are almost one order better than the RMS error calculated with global optimal shape parameter and these RMS errors are almost close to the RMS errors calculated with the best shape parameter. Similar kind of improvement in the accuracy can also be seen for the diffusion parameter  $0.01$ , in the Table 5. Therefore, for the convection dominated problems, the local RBF grid free scheme with a local variable optimal shape parameter produces more accurate results than the global optimal shape parameter.

TABLE 3: COMPARISON OF THE GLOBAL (Global), BEST (Best) and LOCAL (Local) OPTIMUM SHAPE PARAMETERS AND THE CORRESPONDING RMS ERRORS FOR EXAMPLES:  $1 - 6$ , at  $\alpha = 1$ .

|       |        | Number of nodes (N) |           |              |           |              |           |              |           |
|-------|--------|---------------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|
| N     |        | 11                  |           | 21           |           | 41           |           | 81           |           |
| →     |        |                     |           |              |           |              |           |              |           |
| a     |        | $\epsilon$          | RMS error | $\epsilon$   | RMS error | $\epsilon$   | RMS error | $\epsilon$   | RMS error |
| Ex. 1 | Global | 0.855               | 5.87(-04) | 0.855        | 1.48(-04) | 0.855        | 3.71(-05) | 0.855        | 9.29(-06) |
|       | Best   | 0.856               | 5.87(-04) | 0.848        | 1.47(-04) | 0.848        | 3.68(-05) | 0.848        | 9.20(-06) |
|       | Local  | 0.852               | 5.89(-04) | 0.852        | 1.47(-04) | 0.852        | 3.69(-05) | 0.852        | 9.23(-06) |
| Ex. 2 | Global | 1.624               | 5.49(-03) | 1.616        | 1.23(-03) | 1.616        | 2.98(-04) | 1.623        | 7.40(-05) |
|       | Best   | 1.848               | 2.74(-03) | 1.816        | 6.84(-04) | 1.808        | 1.72(-04) | 1.808        | 4.29(-05) |
|       | Local  | 1.868               | 2.70(-03) | (1.6, 2.421) | 5.31(-04) | (1.52, 2.47) | 1.39(-04) | (1.1, 1.9)   | 4.52(-05) |
| Ex. 3 | Global | 0.138               | 6.82(-06) | 0.138        | 1.71(-06) | 0.138        | 4.41(-07) | 0.189        | 1.63(-07) |
|       | Best   | 0.140               | 6.76(-06) | 0.140        | 1.67(-06) | 0.130        | 4.14(-07) | 0.240        | 1.46(-07) |
|       | Local  | (0.01, 0.28)        | 9.11(-06) | (0.01, 0.3)  | 3.25(-06) | (0.02, 0.31) | 4.09(-07) | (0.08, 0.4)  | 5.02(-07) |
| N     |        | 11 × 11             |           | 21 × 21      |           | 41 × 41      |           | 81 × 81      |           |
| →     |        |                     |           |              |           |              |           |              |           |
| a     |        | $\epsilon$          | RMS error | $\epsilon$   | RMS error | $\epsilon$   | RMS error | $\epsilon$   | RMS error |
| Ex. 4 | Global | 1.531               | 3.47(-03) | 1.133        | 8.66(-04) | 1.327        | 2.17(-04) | 1.602        | 5.45(-05) |
|       | Best   | 0.144               | 3.27(-03) | 0.253        | 8.24(-04) | 0.382        | 2.06(-04) | 0.184        | 4.90(-05) |
|       | Local  | 0.432               | 3.42(-03) | (0.46, 1.2)  | 8.62(-04) | (2.88, 12.7) | 2.07(-04) | (1.15, 3.6)  | 3.76(-05) |
| Ex. 5 | Global | 0.012               | 2.69(-06) | 1.013        | 7.83(-07) | 0.061        | 1.42(-08) | 0.473        | 1.50(-08) |
|       | Best   | 0.033               | 2.48(-06) | 0.053        | 6.16(-07) | 0.063        | 4.05(-08) | 0.483        | 9.82(-09) |
|       | Local  | (0.04, 0.28)        | 2.72(-06) | (0.09, 0.17) | 6.53(-07) | (0.02, 0.18) | 1.63(-07) | (0.16, 0.33) | 8.69(-08) |
| Ex. 6 | Global | 0.029               | 2.55(-05) | 0.026        | 8.79(-03) | 0.059        | 1.55(-06) | 0.245        | 5.53(-07) |
|       | Best   | 0.027               | 2.53(-05) | 0.067        | 7.80(-04) | 0.067        | 2.82(-06) | 0.187        | 1.09(-07) |
|       | Local  | 0.04, 0.12)         | 2.67(-05) | (0.1, 0.11)  | 8.97(-06) | (0.1, 0.106) | 1.87(-06) | (0.1, 0.41)  | 6.44(-07) |

TABLE 4: COMPARISON OF THE GLOBAL (Global), BEST (Best) and LOCAL (Local) OPTIMUM SHAPE PARAMETERS AND THE CORRESPONDING RMS ERRORS FOR EXAMPLES:  $1 - 6$ , at  $\alpha = 0.1$ .

|       |        | Number of nodes (N) |           |              |           |              |           |              |           |
|-------|--------|---------------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|
| N     |        | 11                  |           | 21           |           | 41           |           | 81           |           |
| →     |        |                     |           |              |           |              |           |              |           |
| a     |        | $\epsilon$          | RMS error | $\epsilon$   | RMS error | $\epsilon$   | RMS error | $\epsilon$   | RMS error |
| Ex. 1 | Global | 0.814               | 1.37(-02) | 0.814        | 3.25(-03) | 0.814        | 8.00(-04) | 0.814        | 1.99(-04) |
|       | Best   | 0.776               | 1.36(-02) | 0.776        | 3.25(-03) | 0.776        | 8.00(-04) | 0.776        | 1.99(-04) |
|       | Local  | 0.688               | 1.37(-02) | 0.688        | 3.26(-03) | 0.688        | 8.01(-04) | 0.688        | 1.99(-04) |
| Ex. 2 | Global | 1.659               | 2.65(-02) | 1.612        | 6.49(-03) | 1.623        | 1.60(-03) | 1.639        | 3.96(-04) |
|       | Best   | 2.936               | 1.69(-02) | 2.760        | 4.17(-04) | 2.720        | 1.04(-03) | 2.712        | 2.60(-04) |
|       | Local  | (2.85, 2.97)        | 1.66(-02) | (0.36, 4.25) | 3.54(-03) | (2.99, 4.22) | 9.11(-04) | (2.63, 5.33) | 2.11(-04) |
| Ex. 3 | Global | 0.569               | 1.21(-02) | 0.538        | 2.97(-03) | 0.521        | 7.40(-04) | 0.554        | 1.84(-04) |
|       | Best   | 2.940               | 6.84(-03) | 2.730        | 1.95(-03) | 2.640        | 5.02(-04) | 2.660        | 1.26(-04) |
|       | Local  | (1.59, 69.)         | 5.17(-03) | (1.59, 2.60) | 1.67(-03) | (1.59, 2.60) | 4.63(-04) | (1.59, 2.6)  | 1.18(-04) |
| N     |        | 11 × 11             |           | 21 × 21      |           | 41 × 41      |           | 81 × 81      |           |
| →     |        |                     |           |              |           |              |           |              |           |
| a     |        | $\epsilon$          | RMS error | $\epsilon$   | RMS error | $\epsilon$   | RMS error | $\epsilon$   | RMS error |
| Ex. 4 | Global | 1.438               | 9.24(-03) | 1.825        | 2.22(-03) | 2.279        | 5.45(-04) | 2.829        | 1.34(-04) |
|       | Best   | 6.611               | 5.64(-03) | 7.107        | 1.61(-03) | 7.504        | 4.24(-04) | 7.702        | 1.08(-04) |
|       | Local  | (2.98, 5.27)        | 5.20(-03) | (7.14, 33.0) | 1.47(-03) | 7.515        | 4.25(-04) | 7.740        | 1.08(-04) |
| Ex. 5 | Global | 0.296               | 4.05(-03) | 0.296        | 1.00(-03) | 0.295        | 2.50(-04) | 0.295        | 6.26(-05) |
|       | Best   | 0.084               | 4.00(-03) | 0.094        | 9.89(-04) | 0.124        | 2.47(-04) | 0.012        | 6.13(-05) |
|       | Local  | (0.01, 1.59)        | 3.78(-03) | (0.33, 2.22) | 1.06(-03) | (0.1, 2.12)  | 2.47(-04) | (0.26, 3.28) | 5.43(-05) |
| Ex. 6 | Global | 0.543               | 7.45(-03) | 0.435        | 1.88(-03) | 0.825        | 4.31(-04) | 0.905        | 1.05(-04) |
|       | Best   | 2.509               | 1.84(-03) | 2.460        | 5.08(-04) | 2.440        | 1.30(-04) | 2.440        | 3.28(-05) |
|       | Local  | 2.514               | 1.84(-03) | 2.456        | 5.08(-04) | (2.38, 2.54) | 1.30(-04) | 2.464        | 3.29(-05) |

TABLE 5: COMPARISON OF THE GLOBAL (Global), BEST (Best) and LOCAL (Local) OPTIMUM SHAPE PARAMETERS AND THE CORRESPONDING RMS ERRORS FOR EXAMPLES: **1 – 6**, at  $\alpha = 0.01$ .

|          |        | Number of nodes ( N ) |           |               |           |               |           |               |           |
|----------|--------|-----------------------|-----------|---------------|-----------|---------------|-----------|---------------|-----------|
| $N$<br>→ |        | 11                    |           | 21            |           | 41            |           | 81            |           |
| $a$      |        | $\varepsilon$         | RMS error | $\varepsilon$ | RMS error | $\varepsilon$ | RMS error | $\varepsilon$ | RMS error |
| Ex. 1    | Global | 0.766                 | 2.89(-01) | 0.766         | 1.07(-01) | 0.766         | 3.05(-02) | 0.766         | 7.14(-03) |
|          | Best   | 61.44                 | 1.56(-01) | 5.204         | 1.04(-01) | 4.165         | 3.02(-02) | 3.496         | 7.10(-03) |
|          | Local  | 68.66                 | 1.57(-01) | 2.800         | 1.06(-01) | 0.926         | 3.05(-02) | (5.4,11.4)    | 6.65(-03) |
| Ex. 2    | Global | 1.783                 | 2.86(-01) | 1.664         | 1.07(-01) | 1.857         | 3.06(-02) | 2.826         | 7.11(-03) |
|          | Best   | 70.66                 | 8.56(-02) | 77.90         | 6.45(-02) | 6.542         | 2.96(-02) | 5.654         | 7.01(-03) |
|          | Local  | 74.76                 | 8.62(-02) | 73.14         | 6.49(-02) | (0.08,23.9)   | 1.29(-02) | (0.68,46.6)   | 4.73(-03) |
| Ex. 3    | Global | 0.627                 | 3.23(-01) | 0.604         | 8.97(-02) | 0.594         | 2.08(-02) | 0.582         | 4.87(-03) |
|          | Best   | 46.68                 | 3.63(-02) | 0.549         | 8.96(-02) | 0.449         | 2.07(-02) | 0.358         | 4.86(-03) |
|          | Local  | 46.85                 | 3.64(-02) | 48.47         | 7.72(-02) | (0.1,9.93)    | 1.66(-02) | (0.01,4.9)    | 4.63(-03) |
| $N$<br>→ |        | 11 × 11               |           | 21 × 21       |           | 41 × 41       |           | 81 × 81       |           |
| $a$      |        | $\varepsilon$         | RMS error | $\varepsilon$ | RMS error | $\varepsilon$ | RMS error | $\varepsilon$ | RMS error |
| Ex. 4    | Global | 39.14                 | 1.09(-01) | 39.40         | 3.93(-02) | 6.646         | 1.56(-02) | 7.628         | 3.51(-03) |
|          | Best   | 39.92                 | 1.08(-01) | 40.27         | 3.90(-02) | 41.28         | 1.07(-02) | 41.48         | 2.72(-03) |
|          | Local  | (7.9,1083)            | 2.00(-01) | (19,2436)     | 2.05(-02) | (5.5,30.4)    | 1.02(-02) | (2.83,31)     | 2.66(-03) |
| Ex. 5    | Global | 57.14                 | 1.94(-02) | 54.52         | 4.43(-03) | 0.468         | 1.38(-02) | 0.512         | 3.11(-03) |
|          | Best   | 59.42                 | 1.83(-02) | 55.90         | 3.88(-03) | 66.78         | 1.11(-02) | 0.344         | 3.10(-03) |
|          | Local  | (55,1463)             | 1.84(-02) | (50.7,1186)   | 3.90(-03) | 71.39         | 1.12(-02) | (0.18,7.96)   | 3.11(-03) |
| Ex. 6    | Global | 36.36                 | 1.54(-02) | 36.43         | 4.34(-02) | 0.522         | 1.99(-02) | 0.505         | 4.67(-03) |
|          | Best   | 36.11                 | 1.53(-02) | 35.31         | 4.32(-02) | 1.826         | 1.98(-02) | 2.954         | 4.67(-03) |
|          | Local  | (7.1,1251)            | 5.61(-02) | (18.1,1917)   | 3.51(-02) | (1.10,36.1)   | 8.24(-03) | (0.61,1.2)    | 4.67(-03) |

One final comparison is the comparison of Table 2 with Tables 3, 4 and 5. That is, comparing the errors obtained using optimization based on the global collocation with the errors obtained using optimization based on the local collocation. It is clear from these tables that the errors, both global and local, for the test problem 1, in tables 3, 4 and 5 are very close to the errors presented under global of the Table 2, if not the best, demonstrate the robustness of the developed optimization procedure.

## 5. Conclusions

The shape parameter of the infinitely smooth Radial Basis Functions (RBF) plays a significant role in obtaining the accurate and stable solutions in RBF based local grid-free (LRBF) schemes. In the present work, we have shown that the cost and error functions obtained using the existing global optimization schemes are highly oscillatory, particularly for large  $N$  and small diffusion parameters, making the optimization process ineffective. To avoid such oscillations we have proposed a *local* algorithm to find the global and local (near) optimal shape parameter(s) to solve Convection-Diffusion Equations (CDE) using LRBF scheme. The developed algorithm is based on the reconstruction of the forcing term in the CDE using the collocation over the centers in the local support domain and the residual errors are calculated using the Rippa's "leave one out cross validation" algorithm. The proposed optimization algorithm has been tested with LRBF scheme over a large number of one and two-dimensional linear CDEs with strong boundary layer solutions and demonstrated its robustness. The main conclusions of the present work are as follows:

- The (near) optimal shape parameter truly depends on the differential operator, forcing term, boundary conditions, basis function ( $\phi$ ), the number of centers and their distribution.

- The proposed *local* optimization algorithm is very stable for a wide range of shape parameters, even with a large number of centers.
- For diffusion dominated problems, the corresponding (*global*) Cost function ideally imitates the (RMS) error function.
- For the convection dominated problems the (near) local optimal variable shape parameters obtained by minimizing the (*local*) Cost function varies spatially according to the nature of the solution. The value of the shape parameter is generally large where the solution varies rapidly and small elsewhere. The variation in these values makes the LRBF scheme to produce accurate and stable solutions.
- The *local* RBF grid free scheme with local (near) variable optimal shape parameters produces more accurate results, when compared with the global (near) optimal shape parameter.
- The *local* optimization algorithm can also be applied to find the shape parameter if one wishes to improve the local accuracy by clustering the nodes in any selected regions.
- The LRBF scheme with a local optimization algorithm can also be applied for solving the large scale complex problems, since they require the inversion of smaller dense matrices.

Finally, the Brent's bracketing algorithm [19] which has been used to find the local minima requires an initial interval that contains the minimum. The interval dependency of this scheme may be resolved using Particle Swarm Optimization (PSO) algorithm; however, it's not been used in the present work. To conclude, in the present work, we have developed global and local optimization schemes for RBF shape parameter based on a local collocation procedure which is free from ill-condition unlike the global optimization schemes those exist in the literature.

## Nomenclature

---

|                      |   |
|----------------------|---|
| $\mathcal{L}$        | Linear convection diffusion (differential) operator                       |
| $\mathcal{B}$        | Boundary operator   |
| $f$                  | Forcing term  |
| $g$                  | Right hand side function for the boundary conditions                      |
| $u$                  | Unknown function  |
| $d$                  | Dimension of the problem  |
| $\mathbb{R}^d$       | Space of $d$ -dimensional real numbers                                    |
| $\Omega$             | Domain  |
| $\partial\Omega$     | Boundary of the domain  |
| $a$                  | Diffusion parameter   |
| $N$                  | Number of nodes in the computational domain                               |
| $n_i$                | Number of nodes in the local support domain of the node $\underline{x}_i$ |
| $\mathcal{C}_i$      | Set of local supporting nodes for $\underline{x}_i$                       |
| $\phi$               | Radial Basis Function (RBF)   |
| $\varepsilon$        | Shape parameter   |
| $m$                  | Order of the RBF  |
| $\underline{x}$      | Position vector   |
| $(x, y)$             | Position vector in two dimension  |
| $\Pi_{m-1}^d$        | Space of all $d$ -variate polynomials with degree $\leq m - 1$            |
| $p_j(\underline{x})$ | $d$ -variate polynomials with degree $\leq m - 1$                         |
| $s(\underline{x})$   | Interpolant   |
| $\lambda$            | Vector in $\mathbb{R}^{n_i}$  |
| $\Phi$               | Interpolation matrix without polynomial term                              |

|                 |  |
|-----------------|--|
| $A$             | Interpolation matrix with polynomial term                                    |
| $\delta$        | Cronical delta   |
| $\psi_j$        | Lagrange Function  |
| $c_j$           | Weights  |
| $C_i^{(k)}$     | Set of local supporting nodes for $x_i$ excluding $k^{th}$ center            |
| $f_i^{(k)}$     | Set of forcing function values over $C_i^{(k)}$                              |
| $\mathcal{L}^2$ | Linear differential operator applied on RBF as a function of second argument |
| $r_{i,k}$       | Residual error vector  |
| $\mu_{i,k}$     | Vector in $\mathbb{R}^{n_i}$   |

---

## References

- [1] Micchelli, C.A. *Interpolation of scattered data: distance matrices and conditionally positive definite functions*, Constructive Approximation, 1986. **2**: p. 11-22.
- [2] 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 and Mathematics with Applications, 1990. **19**: p. 147-161.
- [3] Fasshauer, G.E. *Solving partial differential equations by collocation with radial basis functions*, In Surface Fitting and Multiresolution Methods, Le Méhauté A, Rabut C, Schumaker LL(eds)., Vanderbilt University Press: Nashville, TN. 1997 pp. 131-138.
- [4] Wu, Z. *Hermite-Brikhoff interpolation of scattered data by radial basis functions*, Approximation Theory and Applications, 1992. **8**(2): p. 1-10.
- [5] Franke, C. and R. Shaback, *Convergence order estimates of meshless collocation methods using radial basis functions*, Advances in Computational Mathematics, 1998. **8**(4): p. 382-399.
- [6] Shu, C., H. Ding and K.S. Yeo, *Local radial basis function-based differential quadrature method and its application to solve two- dimensional incompressible Navier-Stokes equations*, Computer Methods in Applications Mechanics and Engineering, 2003. **192** (3): p. 941-954.
- [7] Wright, G.B. and B. Fornberg, *Scattered node compact finite difference-type formulas generated from radial basis functions*, Journal of Computational Physics, 2006. **212**(1): p. 99-123.
- [8] Chandini, G. and Y.V.S.S. Sanyasiraju, *Local RBF-FD solutions for steady convection-diffusion problems*, International Journal of Numerical Methods in Engineering, 2007. **72** (3): p. 352-378.
- [9] Sanyasiraju, Y.V.S.S. and G. Chandini, *Local radial basis function based gridfree scheme for unsteady incompressible viscous flows*, Journal of Computational Physics, 2008. **227**: p. 8922-8948.
- [10] Sanyasiraju, Y.V.S.S. and G. Chandhini, *A RBF based local gridfree scheme for unsteady convection-diffusion problems*, CFD Letters, 2009. **1** (2): p. 59-67.
- [11] Franke, R *Scattered data Interpolation: test of some methods*, Mathematics of computation, 1982. **38**: p. 181-200.
- [12] Hardy, R.L. *Multiquadric equations of topography and another irregular surfaces*, Journal of Geophysical Research, 1971. **76**: p. 1905-1915.
- [13] Shaback, R. *Error estimates and condition numbers for radial basis interpolation*, Advances in Computational Mathematics, 1995. **3**: p. 251-264.
- [14] Rippa, S. *An algorithm for selecting a good value for the parameter  $c$  in radial basis function interpolation*, Advances in Computational Mathematics, 1999. **11**: p. 193-210.

- [15] Kansa, E. J. and R.E. Carlson, *Improved accuracy of multiquadric interpolation using variable shape parameters*, Computers Mathematics with Applications, 1992. **24**(12): p. 99-120.
- [16] Fornberg, B. and Julia Zuev, *The Runge phenomenon and spacially variable shape parameters in RBF interpolation*, Computers Mathematics with Applications, 2007. **54**: p. 379-398.
- [17] Fasshauer, G. E. and J.G. Zhang, *On choosing "optimal" shape parameters for RBF approximation*, Numerical Algorithms, 2007. **45**: p. 345-368.
- [18] Roque, R. M. C. and A. J. M. Ferreira, *Numerical experiments on Optimal Shape Parameters for Radial Basis functions*, Numerical Methods for Partial Differential Equations, 2010. **26**(3): p. 675-689.
- [19] Press, W.H., B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, Numerical Recipes in C. The art of Scientific Computing. Cambridge university press. Cambridge, 1992.