DOUBLY STOCHASTIC RADIAL BASIS FUNCTION METHODS

FENGLIAN YANG[*], LIANG YAN[†], AND LEEVAN LING[‡]

Abstract. We propose a doubly stochastic radial basis function (DSRBF) method for function recoveries. Instead of a constant, we treat the RBF shape parameters as stochastic variables whose distribution were determined by a stochastic leave-one-out cross validation (LOOCV) estimation. A careful operation count is provided in order to determine the ranges of all the parameters in our methods. The overhead cost for setting up the proposed DSRBF method is $O(n^2)$ for function recovery problems with n basis. Numerical experiments confirm that the proposed method not only outperforms constant shape parameter formulation (in terms of accuracy with comparable computational cost) but also the optimal LOOCV formulation (in terms of both accuracy and computational cost).

Key words. Kernel methods; Collocation; Function recovery; Stochastic LOOCV; Random shape parameters

1. Introduction.

In recent years, radial basis function methods (or more generally, kernel-based methods) have been used to solve many problems in science and engineering. In this paper, we focus on *function recovery* problems that include interpolation, approximation, and partial differential equations in some bounded domain Ω. The general framework of these problems is as follows. Suppose we pick a *translation-invariant* radial kernel $K = \varphi(\|\cdot - \cdot\|_{\ell^2(\mathbb{R}^d)}) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ with a function $\varphi : \mathbb{R} \to \mathbb{R}$ known as the radial basis function (RBF). Commonly used kernel, include the Gaussian $\varphi(r) = \exp(-r^2)$, the multiquadrics $\varphi(r) = (1 + r^2)^{\beta/2}$ with $\beta = 1$ and inverse multiquadrics with $\beta = -1$, Sobolev $\varphi(r) = r^{\tau-d/2}\mathcal{K}_{\tau-d/2}(r)$ with $\mathcal{K}$ being the Bessel functions of the second kind, and a class of compactly supported piecewise polynomial kernels [22]. Spatial discretization requires a selection of *trial centers* $Z = \{z_1, \dots, z_N\} \subset \Omega$. Since all the existing convergence theories [7, 9, 17] are given in terms of the *fill distance* of Z, i.e.,

$$h_{Z,\Omega} := \sup_{\zeta\in\Omega} \inf_{z\in Z} \|z - \zeta\|_{\ell^2(\mathbb{R}^d)},$$

it is common to require Z being quasi-uniform; that is, the fill distances of all admissible Z remain proportional to its minimum separating distance

$$q_{Z,\Omega} := \frac{1}{2} \inf_{z_i \neq z_j \in Z} \|z_i - z_j\|_{\ell^2(\mathbb{R}^d)}$$

during refinement and the mesh ratio $h_{Z,\Omega}/q_{Z,\Omega}$ is asymptotically bounded above. We define a finite-dimensional trial space in the form of

$$\mathcal{U}_Z = \mathcal{U}_{Z,\Omega,K} := \mathrm{Span}\{\varphi(\|\cdot - z_j\|_{\ell^2(\mathbb{R}^d)} \mid z_j \in Z\}, \tag{1.1}$$

from which we seek or *recover* the numerical solution for a given problem via some minimization process. For interpolation problems, one seeks the trial function from $\mathcal{U}_Z$ so that it passes through a function f at all given data $f_Z \in \mathbb{R}^N$ at Z. In approximation theory, the interpolant is a linear best approximation problem from $\mathcal{U}_Z$ with respect to the native space norm of K, see [23]. For function approximation,

[*] College of Science, Hohai University, Nanjing 210098, China (yangfenglian@hhu.edu.cn).

[†] School of Mathematics, Southeast University, Nanjing 210096, China (yanliang@seu.edu.cn).

[‡] Department of Mathematics, Hong Kong Baptist University, Hong Kong (lling@hkbu.edu.hk).

one can provide another set of data points $X = \{x_1, \dots, x_M\} \subset \Omega$ denser than Z and minimizes, say, the least-squares residual $\|u(X) - f_X\|_{\ell^2(\mathbb{R}^M)}$ to recover an approximant $u \in \mathcal{U}_Z$ of an unknown function f. Pioneered by Kansa [14], solving partial differential equations by meshfree collocation methods follows a similar setup. Consider boundary value problems in their general form $\mathcal{L}u = f$ in Ω and $\mathcal{B}u = g$ on $\partial\Omega$. Using a sufficiently dense set of collocation points $X \subset \Omega \cup \partial\Omega$ in the interior and on the boundary for collocating $\mathcal{L}$ and $\mathcal{B}$ respectively, we can obtain an overdetermined collocation system that can be solved by least-squares minimization.

In all these applications, it is trivial that having a suitable trial space is essential for the performance of kernel methods. A good trial space should contain some good approximation to the solution, which makes it problem dependent. However, this is not straightforward in practice. The most important thing in defining a trial space as in (1.1) is to select a kernel K or equivalently a RBF φ. Existing theories tell us that C^∞ kernels (i.e., Gaussian and multiquadrics) are excellent for recovering smooth functions and high orders of convergence can be expected. For functions with low regularity, one should go for kernels with algebraic smoothness orders (i.e., Sobolev and compactly supported Wendland's functions). Yet, scaled RBF φ_ε can be easily obtained from any RBF φ of our choice via a *shape parameter* $\varepsilon > 0$ by $\varphi_\varepsilon(r) = \varphi(\varepsilon r)$. This opens up a large amount of research aiming to seek for the *optimal* shape parameter [6, 8, 12, 18]. The research interest was not limited to constant shape parameters. Soon after the invention of the RBF collocation method, it was observed that using variable shape parameters can improve accuracy of meshfree methods [15]. More recent works in this direction can be found in both global [24] and local [1] RBF formulations. These variable shape parameter strategies take spatial information into account and assign different shape parameters to each kernel based on the local density of its trial centers. As of today, we still lack of theoretical results regarding the variable shape parameter formulations. In [4], it was shown that one can mimic variable shape parameter approach by treating treating it as an additional coordinate, which also results in an isometric reproduction kernel Hilbert space as in the case of constant shape parameter.

Recently, it was reported that some cost-efficient *random* approaches [3, 16, 19] can also help improve the accuracy of RBF methods. After clarifying our notations in Section 2, we will present the doubly stochastic radial basis function (DSRBF) method that uses quasi-optimal *random shape parameters* in Section 3. Implementation and other numerical aspects of the proposed method will also be addressed. We conclude the paper with some numerical demonstrations in Section 4.

2. Abstract formulation. Without loss of generality, we consider problems in the form of

$$\Lambda u = f_\Lambda \quad \text{in } \Omega \subset \mathbb{R}^d, \tag{2.1}$$

that is defined via an infinite set of functionals $\Lambda : C(\Omega) \to C(\Omega)$ and has exact solution u^*. For interpolation and function approximation problems, we have

$$\Lambda := \{\lambda_x = \delta_x \,|\, x \in \Omega\} \text{ and } f_\Lambda := \{f_{\lambda_x} = \lambda_x u^* \,|\, \lambda_x \in \Lambda\}.$$

For the boundary value problem in Section 1, we can compact the two differential operators into

$$\Lambda := \{\lambda_x \,|\, \lambda_x = \delta_x \mathcal{L} \text{ if } x \in \Omega \text{ or } \lambda_x = \delta_x \mathcal{B} \text{ if } x \in \partial\Omega\}$$

and f_Λ can be defined as in interpolation problems. We assume that (2.1) is well-posed and consider spatial discretization at $X \in \Omega \cup \partial\Omega$ by a finite set of functionals $\Lambda_X := \{\lambda_x \in \Lambda \,|\, x \in X\} \subset \Lambda$.

It is known that the meshfree interpolation process in trial space (1.1) is equivalent [21] to finding the unbiased estimator for a stochastic Gaussian process with covariance K from realization f_Z at Z. We define the *doubly stochastic trial space* as

$$\mathcal{U}_{Z,\mathcal{E}} = \mathcal{U}_{Z,\mathcal{E},\Omega,K} := \operatorname{Span}\{\varphi(\varepsilon_j \| \cdot - z_j\|_{\ell^2(\mathbb{R}^d)} \mid z_j \in Z,\, \varepsilon_j \sim \mathcal{E}\} \tag{2.2}$$

for some quasi-uniform trial centers Z and some stochastic shape parameters following the probability distribution $\mathcal{E}$. The trial space in (2.2) can therefore be related to some stochastic Gaussian process with a stochastic covariance, and thus, *doubly stochastic*. It is strongly encouraged to define a semi-discretized problem as follows

$$u_{Z,\mathcal{E}} = \underset{u \in \mathcal{U}_{Z,\mathcal{E}}}{\arg\inf} \|\lambda_x u - f_{\lambda_x}\|_{L^2(\Omega)}, \tag{2.3}$$

for the sake of computational efficiency, in order to identify some numerical approximation.

Let us consider interpolation problems; recall that any trial function is a linear combination of the basis used in defining (2.2) and is in the form of

$$u(x) = \sum_{z \in Z} \alpha_j \varphi_j(x) := \sum_{z \in Z} \alpha_j \varphi(\varepsilon_j \|x - z_j\|_{\ell^2(\mathbb{R}^d)}) \tag{2.4}$$

with $\varepsilon_j \sim \mathcal{E}$ and for some coefficients $\alpha = [\alpha_1, \ldots, \alpha_{n_Z}]^T \in \mathbb{R}^{n_Z}$. To have a well-posed fully-discretized problem, we make an observation $\varepsilon = \{\varepsilon_j\}_{j=1}^{n_Z} \sim \mathcal{E}^{n_Z}$ and define the least-squares numerical solution

$$u_{X,Z,\varepsilon} = \underset{u \in \mathcal{U}_{Z,\varepsilon}}{\arg\inf} \|\lambda_x u - f_{\lambda_x}\|_X^2 := \underset{u \in \mathcal{U}_{Z,\mathcal{E}}}{\arg\inf} \sum_{x \in X} |\lambda_x u - f_{\lambda_x}|^2. \tag{2.5}$$

For every realization $\varepsilon_j \sim \mathcal{E}$, imposing interpolation condition on (2.5) at $X = Z$ yields the familiar square unsymmetric system

$$A_\varepsilon(Z, Z)\alpha = f(Z),$$

with matrix entries $[A_\varepsilon(Z, Z)]_{ij} = \varphi(\varepsilon_j \|z_i - z_j\|_{\ell^2(\mathbb{R}^d)})$ for $z_i, z_j \in Z$. Unless $\mathcal{E}$ in (2.2) is a degenerate distribution, i.e., constant shape parameter, there is no theory to ensure the invertibility of $A_\varepsilon(Z, Z)$. But, as long as the matrix $A_\varepsilon(Z, Z)$ is non-singular for that particular realization ε of $\mathcal{E}$, we have an interpolant from $\mathcal{U}_{Z,\mathcal{E}}$, including all those interpolants with constant shape parameters from the set of possible outcomes of $\mathcal{E}$. All interpolants can attain the zero minimum of (2.5). Without any extra information, it is unclear how to pick a winning interpolant out of the infinitely many. On the other hand, numerical evidence in the literature suggests that they can outperform those with a constant coefficient.

Similar arguments also apply to the proposed PDE solvers and we can use (2.5) to define DSRBF solutions. Well-posedness of the PDE guarantees that there is a uniformly stable and convergent discretization [20]. The explicit form of stability requires case by case analysis, but existing theories suggest that the collocation points X should be denser than the trial centers Z. For example, convergence estimates of least-squares RBF collocation methods for second order strongly elliptic PDEs [7] were built upon an $H^2(\Omega)$ stability estimate on a class of kernels that can reproduce Sobolev spaces. That is, for sufficiently dense X, the X-norm residual in (2.5) is an upper bound to the error function in $H^2(\Omega)$-norm. Without going into the details of convergence analysis, we shall focus on cost-efficient formulation for determining some quasi-optimal trial space in the form of (2.2) and numerically verify that (2.5) outperforms other techniques in accuracy.

It is out of the scope of this paper to thoroughly analyze the convergence behaviour of the proposed DSRBF approach. There are some convincing arguments to explain why the proposed least-squares solutions on some doubly stochastic trial space can improve accuracy. The near-singular resultant matrix can be a result of closely-placed data points either in the set of Z or X. When two trial centers $z_i, z_j \in Z$ are closed with respect to given computational arithmetics, identically shaped trial basis functions centered at these centers have nearly equal values at X. This leads to two columns of nearly identical values and the problem of ill-conditioning. By using two different shape parameters $\varepsilon_i \neq \varepsilon_j$ at z_i and z_j respectively, the problem is circumvented. If two collocation points are too close to each other, the same situation occurs to the rows of the resultant matrix. However, these nearly identical rows will not affect the condition numbers of overdetermined matrices provided that there are enough rows to form a well-conditioned square submatrix within, see [16]. Yet, we still need to decide what the RBF φ and distribution $\mathcal{E}$ are in order to start using the proposed DSRBF approach.

3. Quasi-optimal doubly stochastic trial space. To begin, we consider the problem of finding optimal shape parameter for trial spaces in the form of (1.1) for some scaled RBF φ_ε. For any function recovery problem from a given trial space, we define the optimal shape parameter ε^* in such a way that u_{X,Z,ε^*} is *better* than other $u_{X,Z,\varepsilon}$ with $\varepsilon \neq \varepsilon^*$ under some error measure.

The idea of using leave-one-out cross validation to select optimal shape parameters for RBF interpolation was proposed in [18]. Readers can find an extension to pseudo-spectral RBF methods for solving PDEs in [10]. We refer readers to the original articles for details and we will only include the important formula here. For any $N \times N$ system of linear equation $A_\varepsilon \alpha = b$, whose matrix depends on a single shape parameter $\varepsilon > 0$, the LOOCV cost vector $e(\varepsilon) = [e_1, \ldots, e_N]^T$ can be computed by using the formula

$$e_j(\varepsilon) = \frac{\alpha_j}{[A_\varepsilon^{-1}]_{jj}} \quad \text{for } j = 1, \ldots, N \tag{3.1}$$

for its jth error estimator. The ℓ^2-LOOCV optimal shape parameter can then be defined as

$$\varepsilon^* = \underset{\varepsilon>0}{\arg\min}\, \|e(\varepsilon)\|_{\ell^2(\mathbb{R}^N)}. \tag{3.2}$$

The minimization problem (3.2) can be solved by golden section search, i.e., by the function `fminbnd` of Matlab, which takes $\mathcal{O}(1)$ steps to find the optimal. From the computational point of view, this formula reduces the expensive $\mathcal{O}(N^4)$ overall cost of LOOCV down to the $\mathcal{O}(N^3)$ cost of solving an $N \times N$ linear system by LU-factorization once per iteration.

Seeking the optimal shape parameter is simply a problem of point estimation in statistics. In practice, a complication is that the theoretical optimal and the numerical optimal shape parameters do not necessarily coincide due to the problem of ill-conditioning as $\varepsilon \searrow 0$ and trial basis functions become flat. In this section, we attempt to estimate the numerical optimal by making a small amount of *observations* in double precision. Without further information about the statistics of the observation error, we propose using the χ^2 distribution for $\mathcal{E}$, whose degree of freedom is to be determined by the mean of our observations.

3.1. Stochastic LOOCV estimations. Suppose the recovery problem in Section 2 is defined by some given set of functionals Λ. Further suppose that the required user inputs, including RBF φ, trial centers Z, and collocation points X with $n_X \geq n_Z$, are fixed and a constant shape parameter $\varepsilon > 0$ remains undetermined. Then, identifying the trial function from the minimization problem (2.5) is equivalent to finding $\alpha = \alpha(\varepsilon)$ in (2.4) from some matrix system

$$A_\varepsilon(X, Z)\alpha = f_{\Lambda_X}, \tag{3.3}$$

which implicitly depends on ε. Unless we have $n_X = n_Z$, $A_\varepsilon(X,Z)$ does not fit the interpolation framework in [18] to allow an $\mathcal{O}(N^3)$ LOOCV computations. Instead of reformulating and redefining the LOOCV optimal value for $X \neq Z$, we want to have a fast numerical recipe to estimate the optimal shape parameters ε^* as follows.

Firstly, we random sample the collocation points to yield square systems. Then, we use formula (3.1) to compute the cost vector for a leave one-pair (of collocation point and trial center) out cross validation.

A1 (Sampling X). If $n_X > n_Z$, we randomly select a subset $\widetilde{X} \subset X$ of $n_{\widetilde{X}} = n_Z$ collocation points to yield a reduced square submatrix system $A_\varepsilon(\widetilde{X}, Z)\alpha = f_{\Lambda_{\widetilde{X}}}$ and denote the corresponding optimal shape parameter by $\widetilde{\varepsilon}^*$. Otherwise, when $n_X = n_Z$, we have $\widetilde{\varepsilon}^* = \varepsilon^*$, the LOOCV optimal of the full problem. □

When $A_\varepsilon(X,Z)$ is overdetermined, we shall run Algorithm **A1** multiple times to yield different optimal ε for $A_\varepsilon(\widetilde{X}, Z)$ associated with different sampling $\widetilde{X}$ of X. To evaluate formula (3.1), we need to solve (3.3) by an LU-factorization. It is not cost effective to spend $\mathcal{O}(n_Z^3)$ to solve for these optimal shape parameters in every iteration. Moreover, there is a high chance that the computed value of $\widetilde{\varepsilon}^*$ will be polluted by rounding error when the matrix system is ill-conditioning and deviate from the optimal one. Thus, we propose using a fast procedure to estimate the value of $\widetilde{\varepsilon}^*$.

We adopt the following stochastic framework [2] to estimate the diagonal of any $N \times N$ matrix M. Let $\{v_j \in \mathbb{R}^{nz}\}_{j=1}^{N_{\text{rv}}}$ be a sequence of $N_{\text{rv}} < N$ random vectors. Then $\mathtt{diag}(M)$ can be estimated by the following vector sequence

$$\mathtt{diag}(M) \approx \Big[\sum_{k=1}^{N_{\text{rv}}} v_k \odot M v_k\Big] \oslash \Big[\sum_{k=1}^{N_{\text{rv}}} v_k \odot v_k\Big], \tag{3.4}$$

where $\odot$ and $\oslash$ represent element-wise multiplication and division operators of vectors, respectively. To estimate the diagonal entries of $M = A_\varepsilon(\widetilde{X}, Z)^{-1}$, we generate a sequence $\{w_j \in \mathbb{R}^{nz}\}_{j=1}^{N_{\text{rv}}}$ of N_{rv} normal random vectors and compute $Mv_j = w_j$, i.e., $v_j = A_\varepsilon(\widetilde{X}, Z)w_j$, for $j = 1, \ldots, N_{\text{rv}}$ to form the $\{v_j\}$ random sequence of vectors in the column space of $A_\varepsilon(\widetilde{X}, Z)$. Putting these random vectors into (3.4) yields

$$\mathtt{diag}(A_\varepsilon(\widetilde{X}, Z)^{-1}) \approx \Big[\sum_{k=1}^{N_{\text{rv}}} v_k \odot w_k\Big] \oslash \Big[\sum_{k=1}^{N_{\text{rv}}} v_k \odot v_k\Big], \tag{3.5}$$

which can be used in (3.1) to approximate the error estimator. Since the column space $Col(V)$ of $V = [v_1, \dots, v_{N_{\rm rv}}]$ is a subspace of the column space of $A_\varepsilon(\widetilde{X}, Z)$, we can seek for an approximation to the solution α of (3.3) from $Col(V)$. We do so by projecting the right-hand vector $\mathrm{P}_V f_{\Lambda_{\widetilde{X}}} \in Col(V)$. Then, an approximate solution is given by

$$\alpha \approx A_\varepsilon(\widetilde{X}, Z)^{-1}\mathrm{P}_V f_{\Lambda_{\widetilde{X}}} = A_\varepsilon(\widetilde{X}, Z)^{-1} V V^+ f_{\Lambda_{\widetilde{X}}} = W V^+ f_{\Lambda_{\widetilde{X}}}, \tag{3.6}$$

where V^+ is the pseudo-inverse of V and $W = [w_1, \dots, w_s]$. Putting (3.5) and (3.6) into (3.1) yields an approximation to ε^*. Below is a summary of this subroutine.

A2 (Estimating $\widetilde{\varepsilon}^*$). Given $A_\varepsilon := A_\varepsilon(\widetilde{X}, Z)$ from **A1** and a small integer $N_{\rm rv} < n_Z$, we generate an $n_Z \times N_{\rm rv}$ random matrix W and compute $V = A_\varepsilon W$. Then, we estimate $\mathrm{diag}(A^{-1})$ by putting the columns of W and V into (3.5). Next, we solve (3.6) to obtain an approximation to the solution α of (3.3). An estimated cost vector is defined by using these approximated α_j and A_ε^{-1} in (3.1). □

We now have all the necessary numerical procedures to make *observations* of the optimal shape parameter. Although we do not expect any of these estimate $\widetilde{\varepsilon}^*$ from **A2** to be exact, we expect their statistics to reflect some truth after a good amount of repeated observations. Below is a summary of the whole algorithm, whose computational cost will be analyzed in the next section.

A3 (Making observations). For $j = 1$ to some $N_{\rm obs} > 0$, we apply **A1** to randomly select a square submatrix $A_\varepsilon(\widetilde{X}_j, Z)$ of $A_\varepsilon(X, Z)$, and estimate its optimal shape parameter by some minimization algorithm, in which we use the estimated cost computed by **A2** with $N_{\rm rv}$ random vectors. Then, store result as $\widetilde{\varepsilon}_j$. □

3.2. Doubly stochastic RBF methods.

Now we have a way to collect a set of observations $\widetilde{\varepsilon} = [\widetilde{\varepsilon}_1, \dots, \widetilde{\varepsilon}_{N_{\rm obs}}]^T$ of the optimal shape parameter for problem Λ discretized by RBF ϕ_ε on trial centers Z and collocation point X. Although all attempts in the literature so far use uniform distributions $\mathcal{E}$ in (2.2), we focus on $\mathcal{E}$ being the χ^2 distribution, in which we only need to estimate one parameter. To finally solve the function recovery problem, we execute the procedure below:

- Generate a random vector $[\varepsilon_j]_{j=1}^{nz} \sim \mathcal{E}^{nz} = [\chi^2(\mathrm{mean}(\widetilde{\varepsilon}))]^{nz}$ based on the observations made in **A3**.
- Construct the matrix system in (3.3), where

  $$[A_\varepsilon(X, Z)]_{ij} = \lambda_{x_i}\phi(\varepsilon_j(\cdot - z_j)) \text{ and } [f_{\Lambda_X}]_i = \lambda_{x_i} u^*$$

  for $1 \le i \le n_X$ and $1 \le j \le n_Z$.
- Seek for numerical solution in the form of (2.4) by solving (3.3) for the expansion coefficient $\alpha \in \mathbb{R}^{nz}$ in the least-squares sense (2.5).

The total complexity for pre-optimizing the trial space as in **A3** depends on the three parameters:

- $N_{\rm rv}$, the number of random vectors used in estimating the LOOCV error estimators in **A1**,
- $N_{\rm obs}$, the number of observations of optimal shape parameter we made, and
- $\overline{N}_{\rm iter}$, the average number of iterations required for the minimization algorithm in **A3** to terminate.

The overall cost for determining $\mathcal{E} = \chi^2(\cdot)$ is $N_{\text{obs}}\overline{N}_{\text{iter}}$ times of the cost of the innermost computation of **A2**, i.e., (3.5) and (3.6). The leading cost in each iteration of optimal shape parameter search is as follows. It requires $\mathcal{O}(n_Z^2)$ to form the matrix system. Computing the N_{rv} random vectors in the columns of V and the projection $\mathrm{P}_V f_{\Lambda_{\widetilde{X}}}$ share the same $\mathcal{O}(n_Z^2 N_{\text{rv}})$ complexity. Thus, the overhead of using the proposed stochastic LOOCV quasi-optimization is

$$\text{Cost} = \mathcal{O}(N_{\text{obs}}\overline{N}_{\text{iter}}N_{\text{rv}}n_Z^2).$$

The values of N_{obs} and N_{rv} are user determined, but $\overline{N}_{\text{iter}}$ depends on the employed minimization algorithm. By picking some $N_{\text{rv}} = \mathcal{O}(1)$ and keeping track of the *total number of iterations* $N_{\text{obs}}\overline{N}_{\text{iter}}$, a trivial strategy is to stop making new observations when $N_{\text{obs}}\overline{N}_{\text{iter}} = \mathcal{O}(1)$. By doing so, we keep the overhead at $\mathcal{O}(n_Z^2)$ that is lower than the $\mathcal{O}(n_X n_Z^2)$ cost required for solving an $n_X \times n_Z$ square or overdetermined system in the least-squares sense by any direct solver. In the next section, we will give some numerical demonstrations to show that this efficient strategy is also effective.

4. Numerical demonstrations.

It was shown in [2] that the diagonal estimator quickly yields somewhat accurate approximations for a very small number of vectors (N_{rv}). In all presented examples, we run the stochastic LOOCV optimization algorithm in Section 3.1 with $N_{rv} = 15$ to determine $\mathcal{E} = \chi^2(\cdot)$ and hence, setup the proposed DSRBF method. We stop making new observations of the optimal shape parameter when the total number of innermost loops executed in **A2**, $N_{\text{obs}}\overline{N}_{\text{iter}}$, exceeds n_X. To maintain $\mathcal{O}(n_Z^2)$ complexity, we impose another condition that at most 10 observations will be made, i.e., $N_{obs} \leq 10$. Under the typical assumption that $\overline{N}_{\text{iter}} = \mathcal{O}(1)$, which will be numerically verified soon, the setup cost for the proposed DSRBF methods is $\mathcal{O}(n_X n_Z^2)$ for small problems and is $\mathcal{O}(n_Z^2)$ asymptotically.

In **A3** when we estimate $\widetilde{\varepsilon}^*$, we deploy the Matlab function `fminbnd` to solve (3.2), i.e., calling

$$\varepsilon^* = \texttt{fminbnd}(@(\varepsilon)\,\texttt{e}(\varepsilon), \texttt{a}, \texttt{b}), \tag{4.1}$$

for some subroutine $\texttt{e}(\varepsilon)$ that executes **A2** to compute cost vector in (3.1) for a fixed value of ε. Except in the first example, in which we study the effect of the search interval $[a, b]$ on $\overline{N}_{\text{iter}}$, we use $a = 0$ and $b = 5$ in all computations.

To validate the accuracy of the proposed DSRBF method, all reported errors are in $L^2(\Omega)$, which is approximated by sufficiently dense sets of regular or quasi-uniform evaluation points. As for efficiency, we reported the CPU clock times required to run the MATLAB 2015a scripts on a 3.20 GHz CPU and 3.17 GB RAM computer.

Example 1 (Interpolation). This example aims to compare the numerical performance of the proposed stochastic LOOCV optimization and DSBRF methods on infinitely smooth Gaussian (GA) and multiquadric (MQ) basis. For the sake of easy reproducibility, we consider a few simple and artificial interpolation problems in $\Omega = [-1, 1]^2$ for an exponential function $f_1 = \exp(x + y)$, the `peaks` function of Matlab and the Franke's function [13], denoted by f_2 and f_3 respectively. All data sites $X = Z$ in this example are regularly placed in this example.

In Tables 4.1 and 4.2, we present the detailed results obtained for $n_Z = 50 \times 50$ of both test functions. With three different search intervals $[a, b]$, we first note that the average numbers of iterations $\overline{N}_{\text{iter}}$ required for `fminbnd` in the stochastic LOOCV algorithm to converge are around 20, i.e., $\mathcal{O}(1)$ as claimed, and are comparable to N_{iter} required in the full LOOCV algorithm. In terms of accuracy, the proposed

	Constant LOOCV-optimal RBF methods					Proposed stochastic LOOCV/DSRBF			
	$[a,b]$	ε^*	N_{iter}	L^2 errors	CPU(s)	mean($\widetilde{\varepsilon}$)	$\overline{N}_{\text{iter}}$	L^2 errors	CPU(s)
f_1	$[0,3]$	2.366	18	8.708(-07)	106.377	2.144	19.3	7.307(-12)	8.798
	$[0,5]$	4.289	21	4.790(-07)	122.831	3.286	21.2	2.524(-11)	9.779
	$[0,10]$	4.259	22	1.039(-05)	127.622	6.103	18.2	1.457(-09)	8.433
f_2	$[0,3]$	1.053	18	1.191(-11)	100.846	2.195	18.5	8.616(-13)	8.531
	$[0,5]$	3.079	20	2.158(-07)	112.268	3.291	18.6	4.950(-12)	8.611
	$[0,10]$	5.948	17	1.169(-05)	105.925	6.493	19.8	2.543(-10)	9.168
f_3	$[0,3]$	2.898	21	1.029(-02)	129.716	2.306	19.1	1.201(-05)	8.712
	$[0,5]$	4.716	30	1.763(-05)	173.013	3.562	19.3	1.679(-08)	8.919
	$[0,10]$	7.502	24	2.653(-08)	156.313	5.192	20.1	1.148(-09)	9.296

Table 4.1

Example 1: Interpolation problems with the Gaussian basis. Comparison of the optimal shape parameters ε^* by by LOOCV and estimated mean($\widetilde{\varepsilon}$) by the proposed stochastic LOOCV algorithm with $N_{rv} = 15$ for setting up an $n_Z = 50 \times 50$ interpolation problems. Corresponding CPU times and resulting accuracy of the optimal-constant shape parameter RBF methods and the proposed DSRBF methods were provided.

	Constant LOOCV-optimal RBF methods					Proposed stochastic LOOCV/DSRBF			
	$[a,b]$	ε^*	N_{iter}	L^2 errors	CPU(s)	mean($\widetilde{\varepsilon}$)	$\overline{N}_{\text{iter}}$	L^2 errors	CPU(s)
f_1	$[0,3]$	2.306	17	1.256(-06)	98.507	1.896	18.4	2.217(-09)	6.022
	$[0,5]$	3.001	19	6.893(-07)	114.990	2.782	20.1	6.793(-09)	6.586
	$[0,10]$	2.728	21	3.982(-07)	125.355	5.074	20.9	6.438(-09)	6.816
f_2	$[0,3]$	2.659	20	4.289(-07)	119.166	1.906	17.9	1.943(-10)	5.797
	$[0,5]$	3.820	18	5.885(-06)	115.231	2.958	19.7	7.861(-10)	6.346
	$[0,10]$	3.738	18	5.167(-06)	116.148	4.984	21.4	1.258(-08)	6.877
f_3	$[0,3]$	2.999	25	2.473(-06)	147.279	2.156	18.4	1.754(-08)	5.897
	$[0,5]$	4.293	21	7.688(-09)	132.794	3.549	20.1	2.322(-09)	6.388
	$[0,10]$	6.311	24	6.019(-08)	152.457	6.911	21.8	5.046(-09)	6.891

Table 4.2

Example 1: Results of multiquadirc basis under the same setting as of those in Table 4.1.

DSRBF methods outperform the classical constant LOOCV-optimal shape parameter (that changes with n_Z) formulations in all test cases up to 5 orders of magnitude. It is also important to compare the robustness of these methods with respect to the search interval $[a,b]$. In Table 4.1, interpolation errors of constant LOOCV-optimal RBF methods with search interval $[0,5]$ are two order of magnitudes smaller than those with $[0,10]$. If we focus on the values of optimal shape parameter ε^* for f_1 in Table 4.2, LOOCV selected $2.728 \in [0,10]$ but $3.001 \in [0,5]$ suggesting either one must be suboptimal. Problem of ill-conditioning takes the blame for such behaviours. Instead of tracing for the real optimal, our stochastic approach obtains quasi-optimal shape parameters much quicker. By design, our proposed method took less than one-tenth of the time of the full LOOCV algorithm to find mean($\widetilde{\varepsilon}$) and to fix the DSRBF trial space for a given problem. Once setup, both LOOCV and DSRBF takes the same time to solve the resulting $n_Z \times n_Z$ linear systems by some direct method. Results in Table 4.2 reconfirm the above observations.

The fact that the proposed DSRBF is more efficient and accurate than the constant LOOCV approach is robust with respect to the selection N_{rv} and N_{obs}. This is a result of the use of stochastic shape parameters whose construction is insensitive to the accuracy of the approximated value of mean($\tilde{\varepsilon}$). In Table 4.3, we show more numerical results for the case of interpolating f_2 on $n_Z = 50 \times 50$. The search range is set to be $[a,b] = [0,5]$. Using larger values N_{rv} and N_{obs} does not have significant

	GA				MQ			
(N_{rv}, N_{obs})	mean($\widetilde{\varepsilon}$)	$\overline{N}_{\text{iter}}$	L^2 errors	CPU(s)	mean($\widetilde{\varepsilon}$)	$\overline{N}_{\text{iter}}$	L^2 errors	CPU(s)
(15,30)	3.159	19.1	6.681(-12)	24.803	2.944	18.7	1.143(-09)	16.605
(15,50)	3.103	19.5	7.133(-12)	42.186	2.818	19.3	7.287(-10)	29.228
(15,100)	3.234	19.5	3.761(-11)	84.011	3.001	19.8	5.872(-10)	58.416
(30,10)	3.277	21.3	2.116(-11)	10.062	3.135	18.6	6.250(-10)	6.024
(30,30)	3.384	19.7	2.997(-11)	28.079	3.418	19.2	1.621(-08)	19.532
(30,50)	3.247	19.5	5.896(-12)	46.009	3.180	19.3	5.936(-10)	31.292

Table 4.3

Example 1: Comparison of various user defined parameters N_{rv} and N_{obs} and their effects on the approximated mean of optimal shape parameters, resulting L^2-errors, and CPU times of DSRBF approach for interpolating f_2 with $n_Z = 50 \times 50$.

effects in the mean($\widetilde{\varepsilon}$) and interpolation errors, but increases the CPU times in the setup of DSRBF.

Next, we further confirm that the above observations hold asymptotically with increasing n_Z. Numerical experiments were run under identical setup except with varying n_Z. Figure 4.1 shows the LOOCV-optimal shape parameters ε^* and our proposed quasi-optimal $\widetilde{\varepsilon}_j$ obtained using **A3** for interpolating f_2. For multiquadrics basis, the LOOCV-optimal shape parameters grow exponentially with n_Z. Such an expected trend cannot be seen in the case of Gaussian basis due to ill-conditioned nature in their linear systems. Since our algorithm only make $N_{\text{obs}} = 10$ observations, we are not ready to conclude if the statistics will turn out to be correct eventually. The answer to this question has no real impact to our proposed method, but we do see the optimal sitting inside the range of estimated values for large n_Z. Instead of identifying the real optimal shape parameter, it is more important to examine the speed and accuracy of the DSRBF methods.

Figure 4.2(a)-(b) show the CPU times for finding LOOCV-optimal shape parameters and the proposed stochastic-LOOCV algorithm respectively. Two reference lines of slope 2 and 3 were included to confirm that their complexities of $\mathcal{O}(n_Z^3)$ and $\mathcal{O}(n_Z^2)$ are as claimed. Lastly, the convergence profiles in Figures 4.3(a)-(b) show that the proposed DSRBF is more robust, efficient and accurate than LOOCV. By comparing to the accuracy of the constant shape parameters $\varepsilon = 1, 3, 5$, the worthiness of the $\mathcal{O}(n_Z^2)$ overhead cost can be justified by the exact accuracy. The proposed stochastic LOOCV algorithm and DSRBF methods yield stable solution fast convergence and stable solution for large n_Z, both of which are desired for practitioners.

Example 2 (Boundary value problems). This example aims to verify the performance of DSRBF for solving PDEs. We compare the accuracy of our DSRBF methods and RBF methods with various constant shape parameters (among all n_Z).

To further verify the numerical performance of our proposed DSRBF methods, we first consider two-dimensional Possion problems subject to the Dirichlet boundary conditions in the unit square. We choose two analytical solutions; a trigonometric function $u_1^* = \sin(\pi x)\sin(\pi y)$, and a Runge function $u_2^* = 1/(1 + 25(x^2 + y^2))$. The collocation points X were evenly distributed in the domain $\Omega = [-1, 1]^2$ and its boundary $\partial\Omega$. We denote n with the number of partition in each coordinate, and

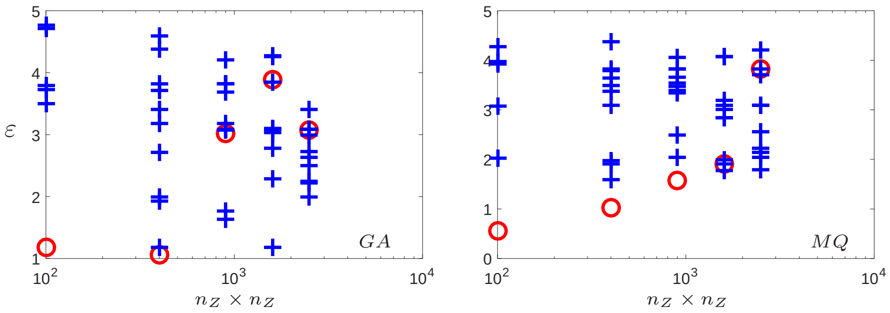

FIG. 4.1. *Example 1: LOOCV-optimal shape parameters (labelled by ○) and 10 stochastic estimates (labelled by +) for interpolation problems of different sizes $n_Z \times n_Z$ with the Gaussian (GA) and multiquadric (MQ) basis.*

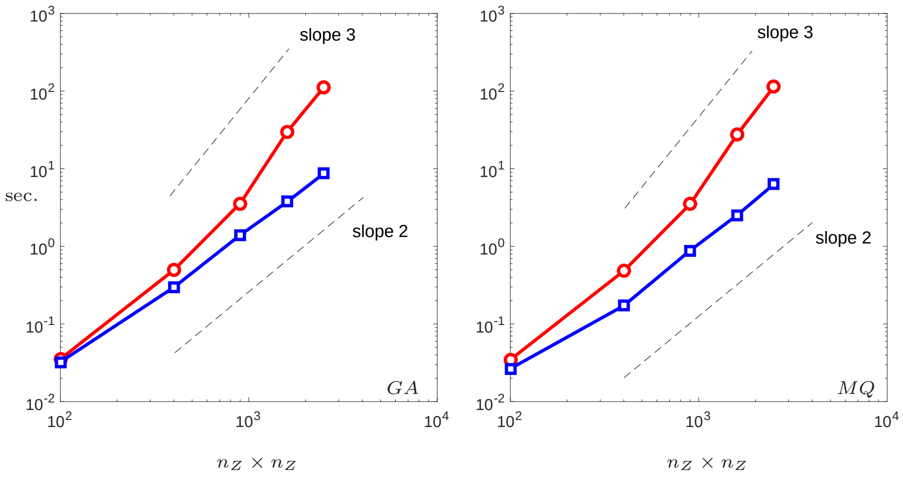

FIG. 4.2. *Example 1: CPU times required for the LOOCV-optimal shape parameters (labelled by ○) and all 10 stochastic estimates (labelled by □) in Figure 4.1.*

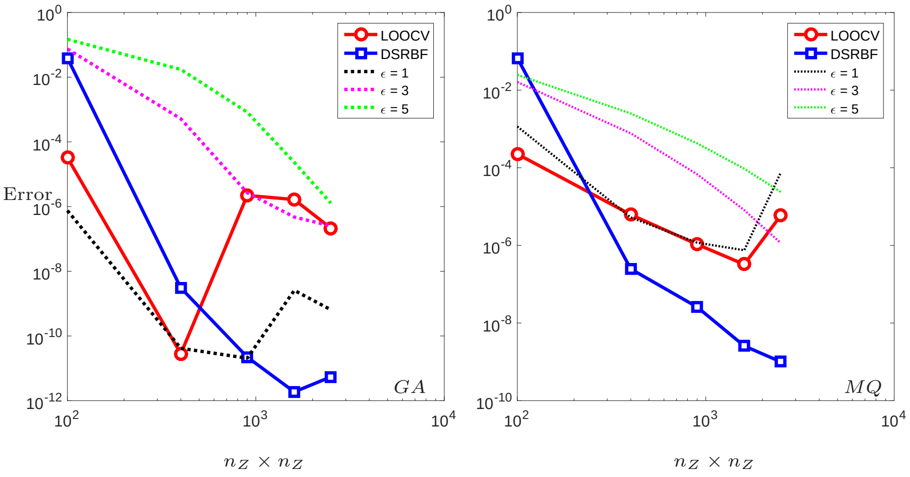

FIG. 4.3. *Example 1: Interpolation error of the constant LOOCV-optimal shape parameter RBF methods (labelled by ○), the proposed DSRBF methods with χ^2-distributed random shape parameters (labelled by □) using the shape parameters in Figure 4.1, and RBF methods with constant parameters.*

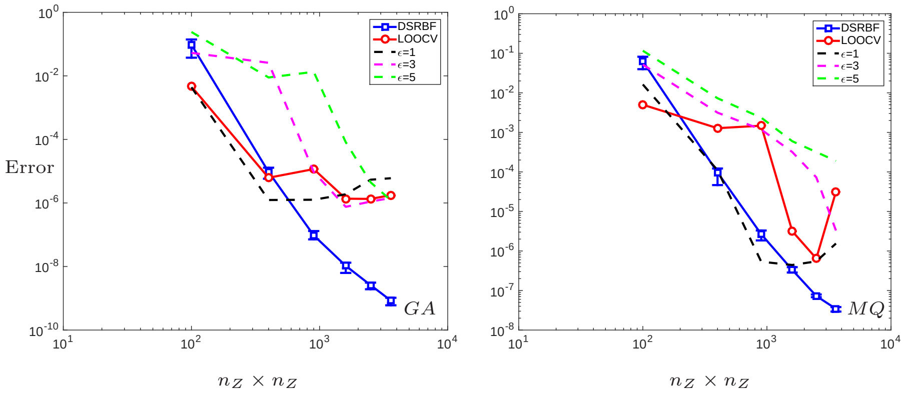

FIG. 4.4. *Example 2: Error profiles when solving a 2D Poisson equation with exact solution $u_1^* = \sin(\pi x)\sin(\pi y)$ by the proposed DSRBF methods using χ^2-distributed shape parameters determined by the stochastic LOOCV algorithm and by RBF methods using various constant shape parameters.*

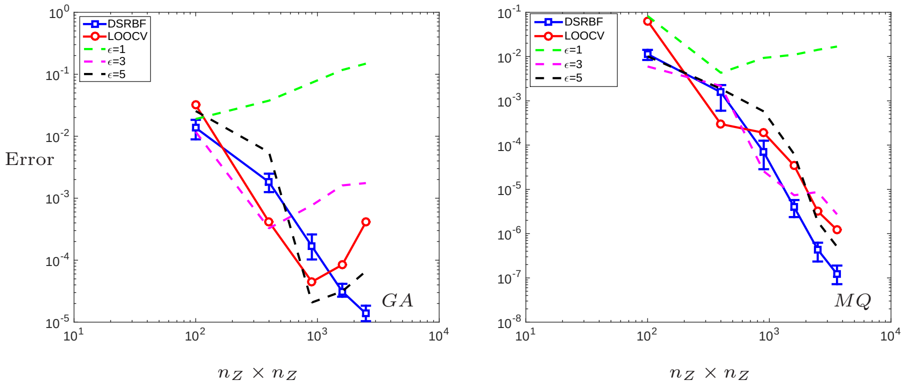

FIG. 4.5. *Example 2: Results of $u_2^* = 1/(1+25(x^2+y^2))$ under the same setting as of those in Figure 4.4.*

thus the total number of collocation points is $N = n^2$. We use the $(n-1)\times(n-1)$ regular grid to generate the trial centers in Z. Other parameters were set up exactly as in Example 1.

Due to the stochastic nature of the proposed methods, they yield different numerical solutions every time. To supply convincing evidence, we perform 10 trial runs to each tested problem and demonstrate the resulting error profiles in Figures 4.4 to 4.5. The error bars on the DSRBF curves represent the range of error among all 10 runs and the solid lines are the mean error of all runs. At first glance, the proposed method not only is more accurate, it also shows no instability due to ill-conditioning in the tested range of n_Z. It is obvious that the proposed DSRBF methods with adaptively chosen random shape parameters using the stochastic LOOCV algorithm are a highly attractive alternative to the standard RBF methods.

Next, we employ the Gaussian kernel to solve a Possion equation with Dirichlet boundary conditions in an amoeba shape domain, whose boundary is given in polar form as $r(\theta) = \exp(\sin\theta)\sin^2 2\theta + \exp(\cos\theta)\cos^2 2\theta$ [5]. We take the rational function $u_3^* = 65/(65+(x-0.2)^2+(y+0.1)^2)$ to generate the boundary data. The numerical results are shown in Figures 4.6 and 4.7.

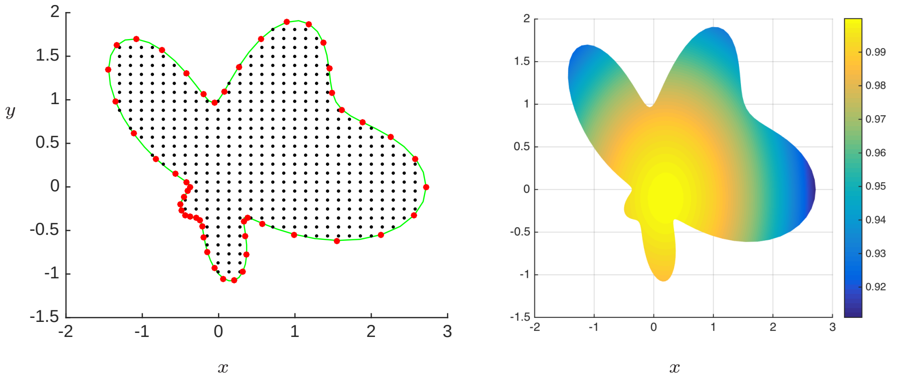

FIG. 4.6. *Example 2: (Left) A schematic diagram of the domain Ω and locations of data points; (Right) The exact solution $u_3^* = 65/(65 + (x - 0.2)^2 + (y + 0.1)^2)$.*

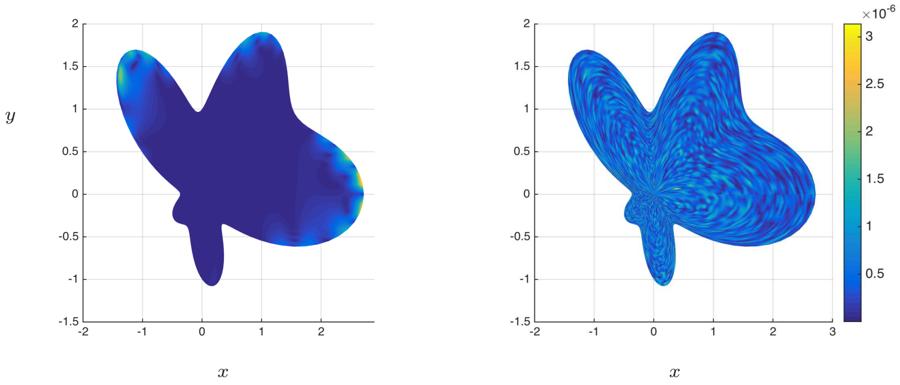

FIG. 4.7. *Example 2: The resulting error functions in the approximation the PDE solution u_3^* by (left) DSRBF and (right) LOOCV.*

Our last demonstration focuses on a three-dimensional Laplace equation $-\Delta u = 0$ on $\Omega = [0, 1]^3$ subject to $u = g$ on the boundary, where g is computed based on the analytical solution $u^* = \exp((x + y)/\sqrt{2}) \cos(z)$. In this example, the data sites $X = Z$ are regularly placed as in the original Kansa method, which does not require one to decide on the ratio of overtesting $n_X : n_Z$. The number of data points tested are $n_Z = 5^3, 10^3, 15^3$, and 20^3. In Figure 4.8, we show the error functions of the proposed methods using the Gaussian basis. Note that the maximum error occurs near boundary, which suggests that our proposed methods, although sophisticated, behave like the original Kansa methods [11]. For that, we also expect all the empirical strategies that improve Kansa methods to work with our proposed DSRBF methods.

Despite the assumptions of the full LOOCV algorithm do not fit into the PDE framework and the algorithm was excluded in this example so far, we include it in our last demonstration and see how it behaves. In Figure 4.9(Left), we show the CPU times for determining shape parameters in this 3D test problem and see the same predicted trends as in the 2D cases in Figure 4.2. For accuracy, see Figure 4.9(Right); using the full LOOCV shape parameters yield no trivial benefit, whereas our proposed methods outperform all other tested methods.

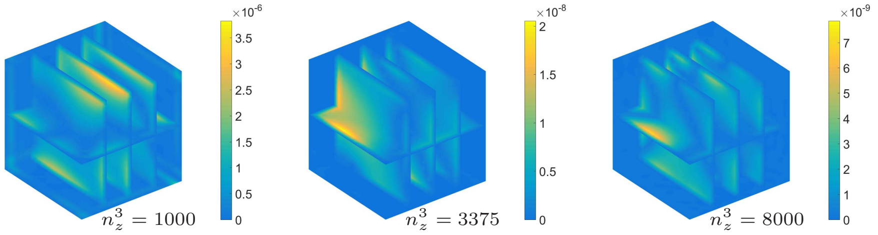

FIG. 4.8. *Example 2: Error distribution when solving a 3D Laplace equation by the proposed DSRBF methods using χ^2-distributed shape parameters determined by the stochastic LOOCV algorithm with $n_z^3 = 1000$, 3375, and 8000.*

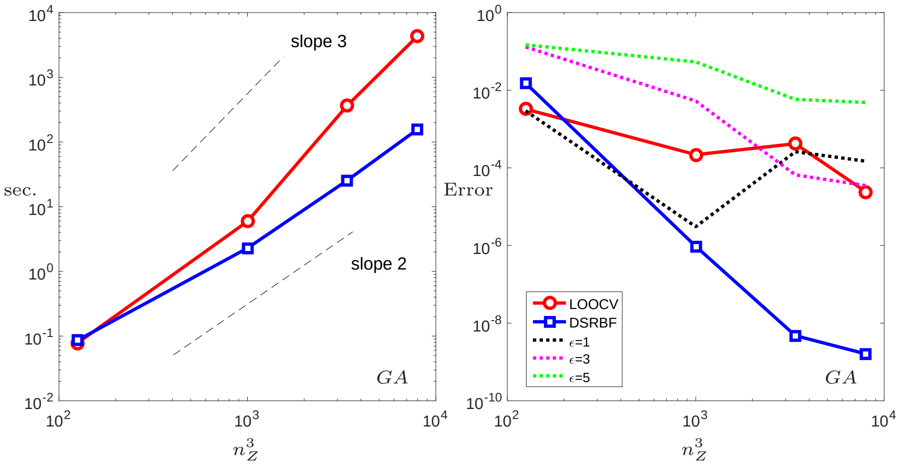

FIG. 4.9. *Example 2: (Left) CPU time required to determine the mean of $\mathcal{E} = \chi^2$ distributions used in the DSRBF methods used in Figure 4.8 and that of the full LOOCV algorithm for comparison; (Right) Error profile of solving a Laplace equation by different methods.*

5. Conclusions .

We propose a stochastic LOOCV algorithm for determining χ^2 random distributed shape parameters for a doubly stochastic radial basis function (DSRBF) method for solving function recovery problems. Similar to the standard radial basis function (RBF) methods, our method is a global method but uses *stochastic* shape parameters. Both the DSRBF and RBF methods require some users to specify in an empirical way shape parameters to complete the definition of trial spaces. To avoid this ad hoc step, one can use a leave-one-out cross validation (LOOCV) approach to select optimal shape parameters for RBF interpolation problems. In this paper, we extend this idea and propose a *stochastic* analogy to estimate the problem dependent optimal shape parameters. Our stochastic approach reduces the $\mathcal{O}(n^3)$ cost of LOOCV down to $\mathcal{O}(n^2)$ for problems of size $m \times n$ with $m \geq n$. Thus, our stochastic algorithm works on overdetermined PDE collocation instead of just the interpolation problem. Numerical experiments confirm that the proposed DSRBF methods are efficient and accurate in comparison with RBF methods using LOOCV-optimal shape parameters. When compared with the empirical constant shape parameter, DSRBF is still favorable because of its robustness and parameter-free nature, all of which justify the $\mathcal{O}(n^2)$ overhead cost for running our stochastic LOOCV algorithm. Since there is no conflict between the DSRBF methods and our previously proposed adaptive trial subspace selection algorithm, the two ideas can work together and further improve the robustness of RBF methods. We leave this to our next exploration.

Acknowledgments. The authors are grateful for the support provided by NSF of China (Nos.11601118, 11771081), the Central University Fund of Hohai University (No. 2013B00714), a Hong Kong Research Grant Council GRF Grant, and a Hong Kong Baptist University FRG Grant.

REFERENCES

[1] V. Bayona, M. Moscoso, and M. Kindelan. Optimal variable shape parameter for multiquadric based RBF-FD method. *J. Comput. Phys.*, 231(6):2466–2481, 2012.

[2] C. Bekas, E. Kokiopoulou, and Y. Saad. An estimator for the diagonal of a matrix. *Appl. Num. Math.*, 57(11):1214–1229, 2007.

[3] J. Biazar and M. Hosami. An interval for the shape parameter in radial basis function approximation. *Appl. Math. Comput.*, 315:131–149, 2017.

[4] M. Bozzini, L. Lenarduzzi, M. Rossini, and R. Schaback. Interpolation with variably scaled kernels. *IMA J. Numer. Anal.*, 35(1):199–219, 2015.

[5] C. S Chen, X. Jiang, W. Chen, and G. Yao. Fast solution for solving the modified helmholtz equation with the method of fundamental solutions. *Commun. Comput. Phys.*, 17(3):867–886, 2015.

[6] A.H.-D. Cheng. Multiquadric and its shape parametera numerical investigation of error estimate, condition number, and round-off error by arbitrary precision computation. *Eng. Anal. Bound. Elem.*, 36(2):220–239, 2012.

[7] K. C. Cheung, L. Ling, and R. Schaback. H^2–convergence of least-squares kernel collocation methods. *SIAM J. Numer. Anal.*, to appear, 2018.

[8] M. Esmaeilbeigi and M. M. Hosseini. A new approach based on the genetic algorithm for finding a good shape parameter in solving partial differential equations by Kansa's method. *Appl. Math. Comput.*, 249:419–428, 2014.

[9] G. E. Fasshauer. *Meshfree approximation methods with Matlab.* Interdisciplinary Mathematical Sciences 6. Hackensack, NJ: World Scientific., 2007.

[10] G. E. Fasshauer and J. G. Zhang. On choosing "optimal" shape parameters for RBF approximation. *Numer. Algorithms*, 45(1-4):345–368, 2007.

[11] A. I. Fedoseyev, M. J. Friedman, and E. J. Kansa. Improved multiquadric method for elliptic partial differential equations via PDE collocation on the boundary. *Comput. Math. Appl.*, 43(3-5):439–455, 2002.

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

[13] R. Franke. Scattered data interpolation: tests of some methods. *Math. Comput.*, 38(157):181–200, 1982.

[14] E. J. Kansa. Multiquadrics—a scattered data approximation scheme with applications to computational fluid-dynamics. II. Solutions to parabolic, hyperbolic and elliptic partial differential equations. *Comput. Math. Appl.*, 19(8-9):147–161, 1990.

[15] E. J. Kansa and R. E. Carlson. Improved accuracy of multiquadric interpolation using variable shape parameters. *Comput. Math. Appl.*, 24(12):99–120, 1992.

[16] L. Ling. A fast block-greedy algorithm for quasi-optimal meshless trial subspace selection. *SIAM J. Sci. Compt.*, 38(2):A1224–A1250, 2016.

[17] L. Ling and R. Schaback. Stable and convergent unsymmetric meshless collocation methods. *SIAM J. Numer. Anal.*, 46(3):1097–1115, 2008.

[18] S. Rippa. An algorithm for selecting a good value for the parameter c in radial basis function interpolation. *Adv. Comput. Math.*, 11(2-3):193–210, 1999.

[19] S. A. Sarra and D. Sturgill. A random variable shape parameter strategy for radial basis function approximation methods. *Eng. Anal. Bound. Elem.*, 33(11):1239–1245, 2009.

[20] R. Schaback. All well–posed problems have uniformly stable and convergent discretizations. *Numer. Math.*, 132:597–630, 2016.

[21] M. Scheuerer, R. Schaback, and M. Schlather. Interpolation of spatial data–A stochastic or a deterministic problem? *European J. Appl. Math.*, 24:601–629, August 2013.

[22] H. Wendland. Error estimates for interpolation by compactly supported radial basis functions of minimal degree. *J. Approx. Theory*, 93(2):258–272, 1998.

[23] H. Wendland. *Scattered data approximation*, volume 17 of *Cambridge Monographs on Applied and Computational Mathematics.* Cambridge University Press, Cambridge, 2005.

[24] S. Xiang, K. M. Wang, Y. T. Ai, Y. D. Sha, and H. Shi. Trigonometric variable shape parameter

and exponent strategy for generalized multiquadric radial basis function approximation. *Appl. Math. Modelling*, 36(5):1931–1938, 2012.