Journal of Computational Physics 430 (2021) 110094

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Optimal design for kernel interpolation: Applications to uncertainty quantification

Akil Narayan [a], Liang Yan [b,c,*], Tao Zhou [d]

[a] Department of Mathematics and Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, UT 84112, United States of America
[b] School of Mathematics, Southeast University, Nanjing 210096, China
[c] Nanjing Center for Applied Mathematics, Nanjing 211135, China
[d] LSEC, Institute of Computational Mathematics, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China

ARTICLE INFO

Article history:
Available online 30 December 2020

Keywords:
Kernel interpolation
Fekete points
Cholesky decomposition with pivoting
Hermite interpolation
Uncertainty quantification

ABSTRACT

The paper is concerned with classic kernel interpolation methods, in addition to approximation methods that are augmented by gradient measurements. To apply kernel interpolation using radial basis functions (RBFs) in a stable way, we propose a type of quasi-optimal interpolation points, searching from a large set of *candidate* points, using a procedure similar to designing Fekete points or power function maximizing points that use pivot from a Cholesky decomposition. The proposed quasi-optimal points results in smaller condition number, and thus mitigates the instability of the interpolation procedure when the number of points becomes large. Applications to parametric uncertainty quantification are presented, and it is shown that the proposed interpolation method can outperform sparse grid methods in many interesting cases. We also demonstrate the new procedure can be applied to constructing gradient-enhanced Gaussian process emulators.

1. Introduction

The field of uncertainty quantification (UQ) has received intensive attention in recent years as theoreticians and practitioners have tackled problems in the diverse areas of stochastic analysis, high-dimensional approximation, and Bayesian learning. A fundamental problem in UQ is the approximation of a multivariate function $u(Z)$, where the parameter $Z = (Z^{(1)}, \cdots, Z^{(d)})$ is a d-dimensional random vector. The function u might be a solution resulting from a stochastic PDE problem or derived quantities of interest (QoI) from such a system. The most popular approach is to expand the function u in a generalized polynomial chaos (gPC) basis [10,36] and approximate its coefficients with a Galerkin projection. In this case, a serious drawback is that existing deterministic solvers must be rewritten.

In recent years, more attention has been devoted to the construction of a surrogate of u based on data $D_N := \{(z_j, u(z_j))\}_{j=1}^N$. This in general refers to stochastic collocation methods, as the sample points $\{z_j\}$ are chosen from the parametric domain. The procedure is also known as non-intrusive response construction.. "Non-intrusive" effectively means that existing black-box tools can be used in their current form. Most existing work concerns the collocation based polynomial approximations, this include the interpolation techniques based on sparse grid [9,21,23,35], the least-squares projection onto polynomial spaces [3,11,22,13], the compressive sampling method with ℓ_1 minimization [5,16,12,37]. Another approach

* Corresponding author.
E-mail addresses: akil@sci.utah.edu (A. Narayan), yanliang@seu.edu.cn (L. Yan), tzhou@lsec.cc.ac.cn (T. Zhou).

https://doi.org/10.1016/j.jcp.2020.110094
0021-9991/

to construct the surrogate using D_N is the so called Gaussian process (GP) regression [2,1,26,31]. The GP provides an analytically tractable Bayesian framework where prior information about the f can be encoded in the covariance function, and the uncertainty about the prediction is easily quantified given measurements. One drawback of GP is the number of samples required for an accurate surrogate increased exponentially as the number of input parameters grows. One method of mitigating this limitation is the incorporation of gradient information into the training of the surrogate [4,18,20,32]. By incorporating derivative values, the cost associated with training an accurate surrogate can be greatly reduced. Unfortunately, gradient-enhanced GP emulators raise more computational challenges compared to ordinary GP emulators [14]. Especially, the condition number of the gradient-enhanced GP emulator will increase much faster than the original GP emulator.

In this work, we approximate $u(Z)$ using a kernel interpolation through a limited number of support points. Over the last four decades, kernel methods using radial basis functions (RBFs) have been successfully applied to scattered data interpolation/approximation in hight dimensions [7,34]. It is known that the kernel interpolation process is equivalent to finding the unbiased estimator for a Gaussian process with covariance from measurements [30,28]. The theoretical connections between kernel methods and Gaussian process regressions for the classical interpolations have recently been highlighted [6,30]. Similar to GP progression, the numerical performance of kernel interpolation is depended on the number of sample points N and the so-called shape parameter ϵ. If ϵ is too large, the interpolant or solution will usually be inaccurate. On the other hand, if ϵ is too small, the condition number of the resultant matrix system will become so bad that linear solvers may perform poorly. Searching for the best shape parameter is still an open problem. On the other hand, for some fixed ϵ, the problem of ill-conditioning can still arise either if the total number of points (commonly called *centers*) is too large or if some points are closely clustered. How to select the optimal centers in the kernel methods is still a challenging work. In this paper, we consider a more practical and solvable problem:

Given a large set of candidate points, how can we choose an "optimal" subset of samples for interpolative kernel approximation?

To this end, we borrow the notion of *Fekete points* in the polynomial interpolation community. The selection strategy we propose only involves linear algebra, and essentially amounts to Cholesky-type decompositions for the design matrix. More precisely, the strategy is data-independent, so it can proceed offline before data is collected. We showed that the selected quasi-optimal points results in a smaller condition number, and thus postpone dramatically the instability of the interpolation procedure when the number of points goes to large. Comparisons with other sampling strategies are presented, and it is noticed that our method demonstrates much improved stability property. Applications to parametric uncertainty quantification are also presented, and it is shown that the proposed kernel interpolation method over perform the sparse grid methods in many interesting cases.

The remained of this paper is organized as follows. Section 2 summaries the problem formulation, where we start with an abstract problem and provide detains on the kernel interpolation. Our approach for choosing the optimal subset of samples is introduced in Section 3. Several numerical examples are considered in Section 4 to study the empirical performance of the new approach. The conclusions are drawn in the final section.

2. Problem setup and kernel interpolation

Here we consider a general setting for PDEs with random input parameters. Let $Z = (Z^{(1)}, \cdots, Z^{(d)}) \in I_Z \subseteq \mathbb{R}^d, d \geq 1$ be a parameter domain representing the uncertain inputs of the system. Consider

$$\begin{cases} u_t(x,t,Z) = \mathcal{L}(u), & \mathcal{D} \times (0,T] \times I_Z \\ \mathcal{B}(u) = 0, & \partial\mathcal{D} \times (0,T] \times I_Z \\ u = u_0 & \mathcal{D} \times \{t=0\} \times I_Z \end{cases} \tag{1}$$

where $\mathcal{D} \in \mathbb{R}^l, l = 1,2,3$, is the physical domain, and $T > 0$ is the terminal time. Here $\mathcal{L}$ is a differential operator and $\mathcal{B}$ a boundary operator that may involve differential operators with respect to the spacial variable x. In a probabilistic framework, Z is modeled as a random variable on a complete probability space $(\Omega, \mathcal{F}, \mathcal{P})$ equipped with the probability distribution function $F_Z(z) = P(Z \leq z)$, with $z \in \mathbb{R}^d$. Note that Z reflects the complete parameterization of uncertain inputs and can include both physical parameters of the system and hyperparameters that characterize certain input processes.

Throughout the paper, we assume that conditioned on the ith independent sample of Z, denoted by z_i, a numerical solution to this problem may be identified by a fixed deterministic solver, such as finite element solves, finite difference solvers, ect. Let $\Xi = \{z_1, \cdots, z_N\} \subset I_Z, N \geq 1$, be a set of sample points in I_Z. For any fixed $x \in \mathcal{D}$ and $t > 0$, we shall denote $u_j = u(x,t,z_j)$ for simplicity. Hereafter we will suppress the notions of x and t whenever possible, with the understanding that our statements are made for all fixed x and t. Once the pairings $(z_j, u_j), j = 1, \cdots, N$, are obtained, the objective is to construct a function $u_N(Z)$ such that $u_N(Z) \approx u(Z)$ in a proper sense.

2.1. Kernel interpolation in parameter space

In the current work, we shall consider the classic kernel interpolations using RBF to construct u_N based on unstructured meshes. The pairing information will be enforced exactly by requiring $u_N(z_j) = u_j$ for all $j = 1, \cdots, N$. In the kernel inter-

polation framework, we first pick a translation-invariant radial kernel $K = \Phi(\epsilon\|\cdot - \cdot\|) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. Here, $\Phi : \mathbb{R} \to \mathbb{R}$ known as the radial basis functions (RBF), ϵ is a shape parameter and $\|\cdot\|$ is, for example, Euclidean distance. Using the set $\Xi = \{z_j\}_{j=1}^N \subset I_Z$, which is usually referred to as the *trial centers* in kernel interpolation, we can define a finite-dimensional trial space in the form of

$$\mathcal{U}_\Xi = \mathcal{U}_{\Xi,K} := \text{span}\{\Phi(\epsilon\|\cdot - z_j\|)|z_j \in \Xi\}. \tag{2}$$

In 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. In the present work we shall confine ourselves to the case of positive definite kernels/RBFs. The important role of positive definiteness for kernel interpolation was pointed out in [19] and, for example, guarantees existence and uniqueness of the approximation. Widely used positive definite kernels/RBFs are the Gaussians, with $\Phi(r) = \exp(-r^2)$, the inverse multiquadrics (IMQ) with $\Phi(r) = 1/\sqrt{1+r^2}$, and the family of compactly supported (CS) RBFs [33,34].

Now, let us consider kernel interpolation problems. Note that any trial function u_N is a linear combination of the basis used in defining (2) and is in the form of

$$u_N(Z) = \sum_{j=1}^N c_j \, \Phi(\epsilon\|Z - z_j\|), \tag{3}$$

for some coefficients $\mathbf{c} = [c_1, \cdots, c_N]^T \in \mathbb{R}^N$. Using the interpolation conditions $u_N(z_i) = u_i, i = 1, \cdots, N$, we obtain the following linear system

$$\begin{bmatrix} \Phi(\epsilon\|z_1 - z_1\|) & \cdots & \Phi(\epsilon\|z_1 - z_N\|) \\ \vdots & \ddots & \vdots \\ \Phi(\epsilon\|z_N - z_1\|) & \cdots & \Phi(\epsilon\|z_N - z_N\|) \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_N \end{bmatrix} = \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix} \quad \text{or} \quad \mathbf{Ac} = \mathbf{u}, \tag{4}$$

where the matrix $\mathbf{A} = K(\Xi, \Xi)$ is symmetric with entries $\mathbf{A}_{ij} = \Phi(\epsilon\|z_i - z_j\|)$ for $z_i, z_j \in \Xi$. The coefficients $\mathbf{c}$ are unique, if the interpolation matrix $\mathbf{A}$ is invertible.

Remark 1. It is known that the kernel interpolation process in trial space (2) is equivalent to finding the unbiased estimator for a Gaussian process with covariance K from realization $\mathbf{u}$ at Ξ, see Appendix A.

Remark 2. If two trial centers $z_i, z_j \in \Xi$ are too closed to each other, identically shaped trial basis functions centered at these centers have nearly equal values at Ξ. This leads to two columns of nearly identical values and the problem of ill-conditioning. Yet, we need to decide what the experimental configuration Ξ is in order to effectively use the kernel interpolation.

2.2. Leave-One-Out Cross Validation (LOOCV)

Noted that the accuracy of the solution of Eq. (4) and the well-conditioning of the matrix $\mathbf{A}$ depends on the shape parameter ϵ. Much work addressese this issue, see, e.g. [7,8,25,29,38], and the references therein. It is out of our scope of this paper to thoroughly analyze choosing optimal shape parameters. In this work, we use the leave-one out cross validation (LOOCV) algorithm [8,25] to select the value of the shape parameter. In the RBF interpolation setting the LOOCV algorithm targets solution of the underlying linear system $\mathbf{Ac} = \mathbf{u}$, where the entries of the matrix $\mathbf{A}$ depend on the shape parameter ϵ which we seek to optimize. The cost function defined as the norm of the error vector $e(\epsilon)$ has entries [8,25],

$$e_i(\epsilon) = \frac{c_i}{\mathbf{A}_{ii}^{-1}},$$

where c_i is the ith component of the vector $\mathbf{c}$ and $\mathbf{A}_{ii}^{-1} := \left(\mathbf{A}^{-1}\right)_{ii}$ is the ith diagonal element of the inverse of the coefficient matrix. Thus, the optimal value of the shape parameter is considered as the one which minimizes the cost function $e(\epsilon)$. We define the optimal shape parameter as the value ϵ^* such that

$$\|e(\epsilon^*)\| = \min_\epsilon \|e(\epsilon)\|. \tag{5}$$

2.3. A Gradient enhanced approach

We consider inclusion of gradient measurements in the kernel interpolation. Consider the availability of the following data: $\{z_i, u(z_i)\}_{i=1}^N$ and $\{z_i, u'_m(z_i)\}_{i=1}^N$ for all $m = 1, 2, \ldots, d$). Here $u'_m(z) = \frac{\partial f(z)}{\partial z^{(m)}}$ stands for the partial derivative with respect to the m-th variable $z^{(m)}$. We try to find an interpolant of the form

$$u_N(Z) = \sum_{j=1}^{N} c_j \Phi(\epsilon \|Z - z_j\|) - \sum_{m=1}^{d} \sum_{j=1}^{N} \beta_{m,j} \Phi'_m(\epsilon \|Z - z_j\|)$$

with appropriate radial basis functions Φ so that u_N satisfies the generalized interpolation conditions

$$\lambda_i u = \lambda_i u_N,\ i = 1, \ldots, N(d+1).$$

Here, λ_i could denote point evaluation at the point z_i, or it could denote evaluation of some derivative at the point z_i.

After enforcing the interpolation conditions the system matrix is given by

$$\mathbf{B} = \begin{bmatrix} \mathbf{A}_{0,0} & \mathbf{A}_{0,1} & \cdots & \mathbf{A}_{0,d} \\ \mathbf{A}_{1,0} & \mathbf{A}_{1,1} & \cdots & \mathbf{A}_{1,d} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{d,0} & \mathbf{A}_{d,1} & \cdots & \mathbf{A}_{d,d} \end{bmatrix}$$

where $\mathbf{A}_{m,n}$ is the $N \times N$ matrix with the (i, j)th element

$$\left(\boldsymbol{A}_{m,n}\right)_{i,j} = \begin{cases} \Phi(\epsilon \|z_i - z_j\|), & m = n = 0, \\ -\Phi'_n(\epsilon \|z_i - z_j\|), & m = 0, n \neq 0, \\ \Phi'_m(\epsilon \|z_i - z_j\|), & m \neq 0, n = 0, \\ -\Phi''_{m,n}(\epsilon \|z_i - z_j\|), & m \neq 0, n \neq 0, \end{cases} \tag{6}$$

and $\mathbf{B}$ is a symmetric $(d+1)N \times (d+1)N$ matrix.

Note that the matrix $\mathbf{B}$ coincides with the joint covariance matrix of the gradient-enhanced GP emulator using the covariance kernel K, see Appendix B. Similar to the gradient-enhanced GP emulator, the condition numbers of the matrix $\mathbf{B}$ are much larger than those of the ordinary Lagrange interpolation matrix $\mathbf{A}$ [14]. Figs. 1 and 2 present the condition numbers of the system matrix for the design matrixes $\mathbf{A}$ and $\mathbf{B}$. Clearly, the design matrix becomes more singular as the shape parameters are closer to zero or the total number of points becomes large.

3. Methodology

It is obvious that design of the collocation points plays a important role in kernel interpolation. However, selecting optimal locations for trial centers in Ξ is highly non-trivial. These are the motivation of the development of adaptive solvers [15,17,27]. Nevertheless, it is also believed that uniformly distributed interpolation points can lead to better approximation results [29]. Popular choices of such points include the Sobol' points, the low discrepancy points or even the random distributed points. However, it is observed that all these choices have the instability issue when the number of collocation points becomes large, say $N \sim \mathcal{O}(10^2)$, see Fig. 2.

In this work, we shall propose a type of quasi-optimal collocation set for kernel interpolations, by searching from a large set of *candidate* points. The idea is inspired by the computation of "Fekete points" in polynomial approximation.

3.1. Fekete-type interpolation points

We recall the kernel interpolation (3)

$$u_N(Z) = \sum_{j=1}^{N} c_j \, \Phi(\epsilon \|Z - z_j\|) = \sum_{j=1}^{N} d_j \, \ell_j(Z),$$

where the latter equality writes the solution in terms of cardinal Lagrange interpolants, given by

$$\ell_j(Z) = \frac{\det \mathbf{A}(z_1, ..., z_{j-1}, Z, z_{j+1}, ..., z_N)}{\det \mathbf{A}(z_1, ..., z_{j-1}, z_j, z_{j+1}, ..., z_N))}, \qquad j = 1, ..., N. \tag{7}$$

The Lebesgue constant, corresponding to the operator norm of interpolation, is defined as

$$\Lambda_N := \max_{Z \in I_Z} \sum_{j=1}^{N} \left| \ell_j(Z) \right|.$$

Small values of Λ_N indicate that interpolation is a stable operation. The so called Fekete points are a configuration of $(z_1, \ldots, z_N)$ that maximizes the Vandermonde-like matrix determinant, namely,

$$\Xi^* = \max_{\Xi = \{z_1, ..., z_N\} \subset I_Z} \left\{ |\det \mathbf{A}(z_1, ..., z_N)| \right\} \tag{8}$$

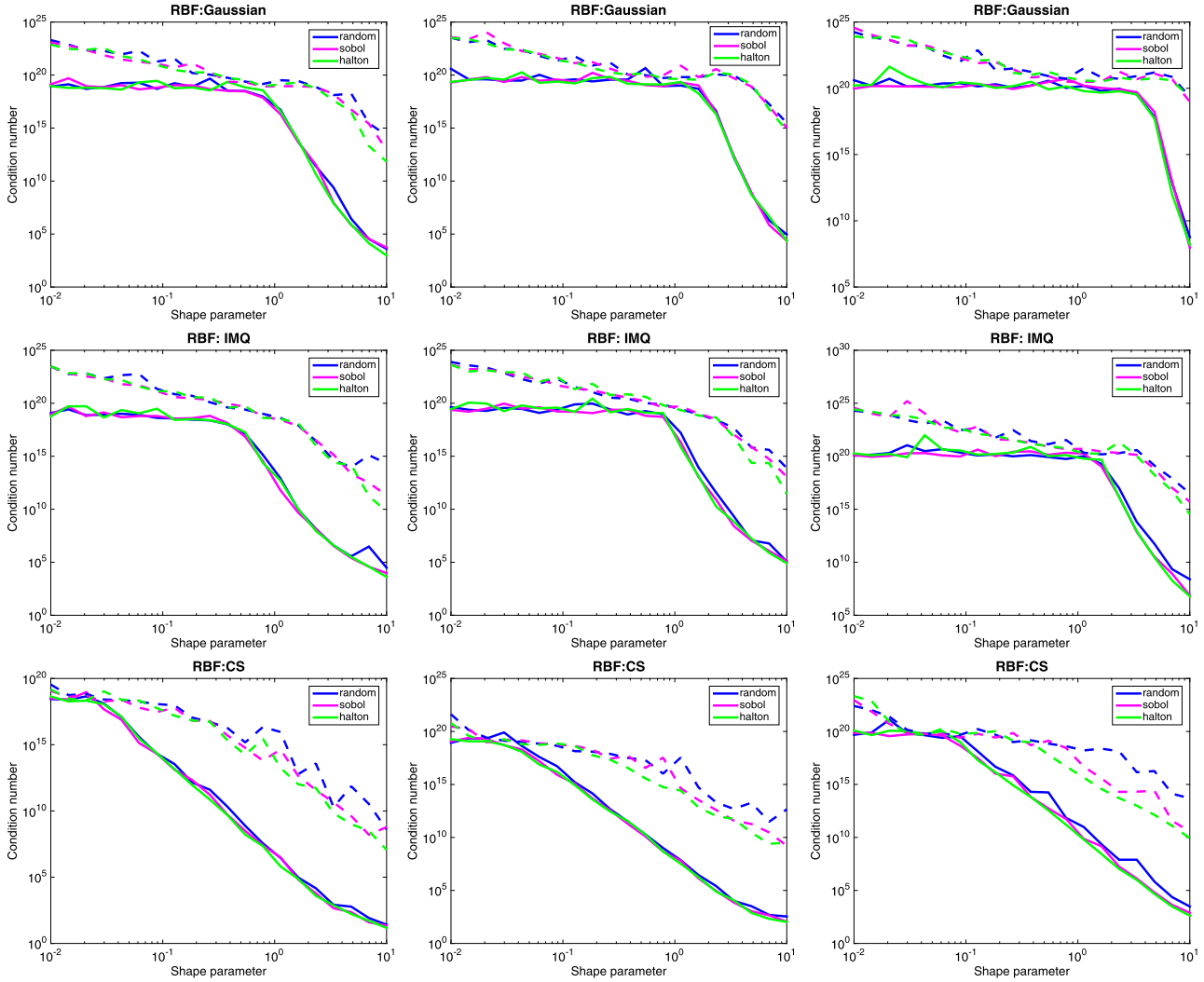

Fig. 1. Curves of the condition numbers for the Lagrange interpolation (solid curves) and the Hermite interpolation (dashed curves) w.r.t. the shape parameter using Gaussian, IMQ and CS, respectively. Left: $N = 50$; middle: $N = 100$; right: $N = 300$; $d = 2$. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

With the relation (7), one observes that the configuration Ξ^* ensures that $|\ell_j(Z)| \leq 1$ for all $Z \in I_Z$, and thus ensures $\Lambda_N \leq N$. In practice, the behavior of Lebesgue constant for Fekete points is frequently sublinear. Thus, Fekete points provide a strategy for stable interpolation. Unfortunately, there is no known way to explicitly characterize or compute these points outside of special polynomial approximation cases. In addition, computationally solving the optimization problem (8) is a daunting task. A typical approach is to relax the optimization problem by seeking a greedy (iterative) solution, i.e.,

$$z_{N+1} = \underset{z \in I_Z}{\arg\max} \left| \det \mathbf{A}(z_1, \ldots, z_N) \right| .$$

In the polynomial approximation literature such an appropriate is titled the "approximate Fekete point" approach, and for kernel interpolation it is a power function maximization approach. We will utilize this procedure in our gradient-enhanced setting to choose centers for approximation.

3.2. Generating quasi-optimal points via greedy selection

In this section, we present a data-independent algorithm to select optimal distribution of centers. The data-independent feature is useful for practical problems, as it allows one to determine, prior to conducting expensive simulations or experimentations, where to collect data samples.

The general goal is to achieve a set of determinant-maximizing (i.e., Fekete) points associated to the gradient-enhanced design matrix $\boldsymbol{B}$. However, direct optimization is likely to be computationally challenging since the optimization problem

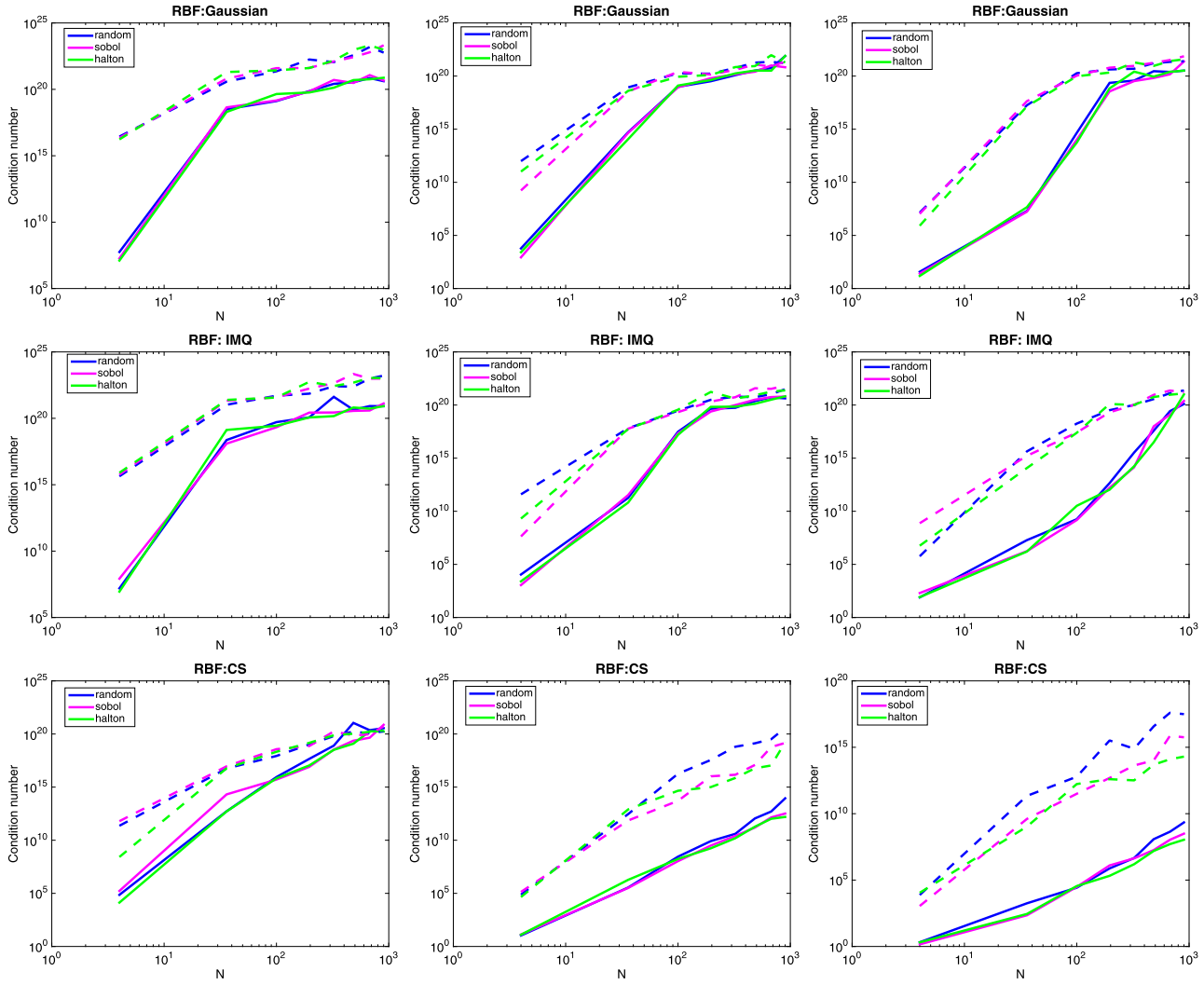

Fig. 2. Curves of the condition numbers for the Lagrange interpolation (solid curves) and the Hermite interpolation (dashed curves) w.r.t. the number of sampling points N using Gaussian, IMQ and CS, respectively. Left: $\epsilon = 0.1$; middle: $\epsilon = 1$; right: $\epsilon = 3$; $d = 2$.

is non-convex, and the dimension of the optimization variables can be very large when a large interpolation grid is used. Therefore, we opt for a greedy strategy, performing the sequential optimization,

$$z_{N+1} := \arg\max_{z \in I_Z} \det \boldsymbol{B}(z_1, \ldots, z_N, z),$$

where $(z_1, \ldots, z_N) \subset \mathbb{R}^d$ is a given set of N centers. The procedure above defines an iterative scheme. Although in principle computing this determinant requires construction of the matrix $\boldsymbol{B}(z_1, \ldots, z_N, z)$, such an expensive construction can be avoided by exercising low-rank updates of LU factorizations. First we note that given $z_1, \ldots, z_N$, we have

$$\boldsymbol{B}(z_1, \ldots, z_N, z) := \begin{pmatrix} \boldsymbol{A}_{0,0} & \boldsymbol{A}_{0,1} & \cdots & \boldsymbol{A}_{0,d} \\ \boldsymbol{A}_{1,0} & \boldsymbol{A}_{1,1} & \cdots & \boldsymbol{A}_{1,d} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}_{d,0} & \boldsymbol{A}_{d,1} & \cdots & \boldsymbol{A}_{d,d} \end{pmatrix},$$

where above $\boldsymbol{A}_{m,n} = \boldsymbol{A}_{m,n}(z_1, \ldots, z_N, z)$. Then if $\boldsymbol{P}_{N+1} \in \mathbb{R}^{(N+1)(d+1)\times(N+1)(d+1)}$ is a permutation matrix that maps element $(N+1)i$ to element $N(d+1)+i$ for $i = 1, \ldots, d+1$ (and retains the order for the remaining $N(d+1)$ elements in the first $N(d+1)$ entries), then

$$\boldsymbol{B}(z_1, \ldots, z_N, z) = \boldsymbol{P}_{N+1} \boldsymbol{B} \boldsymbol{P}_{N+1}^{-T} = \begin{pmatrix} \boldsymbol{B}_N & \boldsymbol{W}(z) \\ \boldsymbol{W}^T(z) & \boldsymbol{B}(z) \end{pmatrix}$$

Then the determinant can be written in terms of the Schur complement of this latter expression,

$$\det \boldsymbol{B}(z_1,\ldots,z_N,z) = \det\left(\boldsymbol{P}\boldsymbol{B}\boldsymbol{P}^T\right) = \det\begin{pmatrix} \boldsymbol{B}_N & \boldsymbol{W} \\ \boldsymbol{W}^T & \boldsymbol{B}(z) \end{pmatrix}$$
$$= \det \boldsymbol{B}_N \det\left(\boldsymbol{B}(z) - \boldsymbol{W}^T\boldsymbol{B}_N^{-1}\boldsymbol{W}\right),$$

where we have introduced the notation $\boldsymbol{B}_N := \boldsymbol{B}(z_1,\ldots,z_N)$. Thus,

$$\arg\max_{z\in I_Z} \det \boldsymbol{B}(z_1,\ldots,z_N,z) = \arg\max_{z\in I_Z} F(z),$$
$$F(z) := \det\left(\boldsymbol{B}(z) - \boldsymbol{W}^T\boldsymbol{B}_N^{-1}\boldsymbol{W}\right).$$

For a fixed z, the major cost of evaluating F involves computing the solution $\boldsymbol{X} \in \mathbb{R}^{N(d+1)\times(d+1)}$ to the linear system $\boldsymbol{B}_N\boldsymbol{X} = \boldsymbol{W}$. There is a secondary, comparatively negligible, cost of computing the determinant of a $(d+1)\times(d+1)$ matrix. For the linear system, we can write the solution in terms of the Cholesky factor of the permuted version of $\boldsymbol{B}_N$,

$$\boldsymbol{P}_N\boldsymbol{B}_N\boldsymbol{P}_N^T = \boldsymbol{L}_N\boldsymbol{L}_N^T$$

so that, if $\boldsymbol{L}_N^{-1}$ is available and stored, then evaluation of F requires application of $\boldsymbol{L}_N^{-1}$, with a $\mathcal{O}(d^3N^2)$ cost, and we can rewrite the optimization as,

$$z_{N+1} = \arg\max_{z\in I_Z} F(z) = \arg\max_{z\in I_Z} \det\left(\boldsymbol{B}(z) - \boldsymbol{V}^T\boldsymbol{V}\right), \qquad \boldsymbol{V} := \boldsymbol{L}_N^{-1}\boldsymbol{W}(z). \tag{9}$$

To aid in continuing the iteration, an efficient update that transforms $\boldsymbol{L}_N^{-1}$ into $\boldsymbol{L}_{N+1}^{-1}$ (the Cholesky factor for the pivoted $\boldsymbol{B}_{N+1}$) is helpful. Again exercising Schur complements, we find that

$$\boldsymbol{P}_{N+1}\boldsymbol{B}_{N+1}\boldsymbol{P}_{N+1}^T = \boldsymbol{L}_{N+1}\boldsymbol{L}_{N+1}^T \quad\Longrightarrow\quad \boldsymbol{L}_{N+1} = \begin{pmatrix} \boldsymbol{L}_N & \boldsymbol{0}_{N(d+1)\times(d+1)} \\ \boldsymbol{V}^T & \widetilde{\boldsymbol{L}} \end{pmatrix},$$

where $\widetilde{\boldsymbol{L}} = \widetilde{\boldsymbol{L}}(z)$ is the Cholesky factor of the Schur complement of the block $\boldsymbol{B}(z_{N+1})$ of the matrix $\boldsymbol{B}_{N+1}$,

$$\widetilde{\boldsymbol{L}}\widetilde{\boldsymbol{L}}^T = \boldsymbol{B}(z_{N+1}) - \boldsymbol{V}^T\boldsymbol{V}. \tag{10}$$

Then a computation shows that

$$\boldsymbol{L}_{N+1}^{-1} = \begin{pmatrix} \boldsymbol{L}_N^{-1} & \boldsymbol{0}_{N(d+1)\times(d+1)} \\ -\widetilde{\boldsymbol{L}}^{-1}\boldsymbol{V}^T\boldsymbol{L}_N^{-1} & \widetilde{\boldsymbol{L}}^{-1} \end{pmatrix}. \tag{11}$$

In summary, given $\boldsymbol{L}_N^{-1}$ that identifies z_{N+1} from the optimization (9), then we compute $\boldsymbol{L}_{N+1}$ by (i) computing $\boldsymbol{V}(z_{N+1})$ from (9), (ii) computing the inverse of the $(d+1)\times(d+1)$ Cholesky factor $\widetilde{\boldsymbol{L}}(z_{N+1})$ of the Schur complement from (10), (iii) assembling the matrix in (11), which requires an additional application of $\boldsymbol{L}_N^{-1}$.

Finally, we emphasize again that in practice we do not compute the solution to (9), but rather we solve the easier optimization problem that replaces the continuum I_Z by a discrete candidate set.

4. Numerical tests

In this section we use several numerical tests to demonstrate the benefit of the quasi-optimal interpolation points. In order to implement our proposed method, we first pick M random centers on I_Z as a candidate set, then we select N optimal points from Section 3.2 as RBF centers. In the following examples, we choose $M = 10^4$.

For comparisons against other methods, we employ the following sampling methods.

- Random sampling: N samples are taken randomly in I_Z;
- Sobol sampling: choose N values of the Sobol sequence;
- Halton sampling: choose N values of the Halton sequence.

To measure the performance of an approximation, we will use the l_2 error (RMSE). Specifically given a set of $Q = 1000$ random samples $\Xi_{test} = \{z_i\}_{i=1}^Q \in I_Z$ and samples of the true function $u(z_i)$ and the kernel approximation $s(z_i)$ we compute

$$E_{l_2} = \left(\frac{1}{Q}\sum_{i=1}^{Q} |s(z_i) - u(z_i)|^2\right)^{1/2}. \tag{12}$$

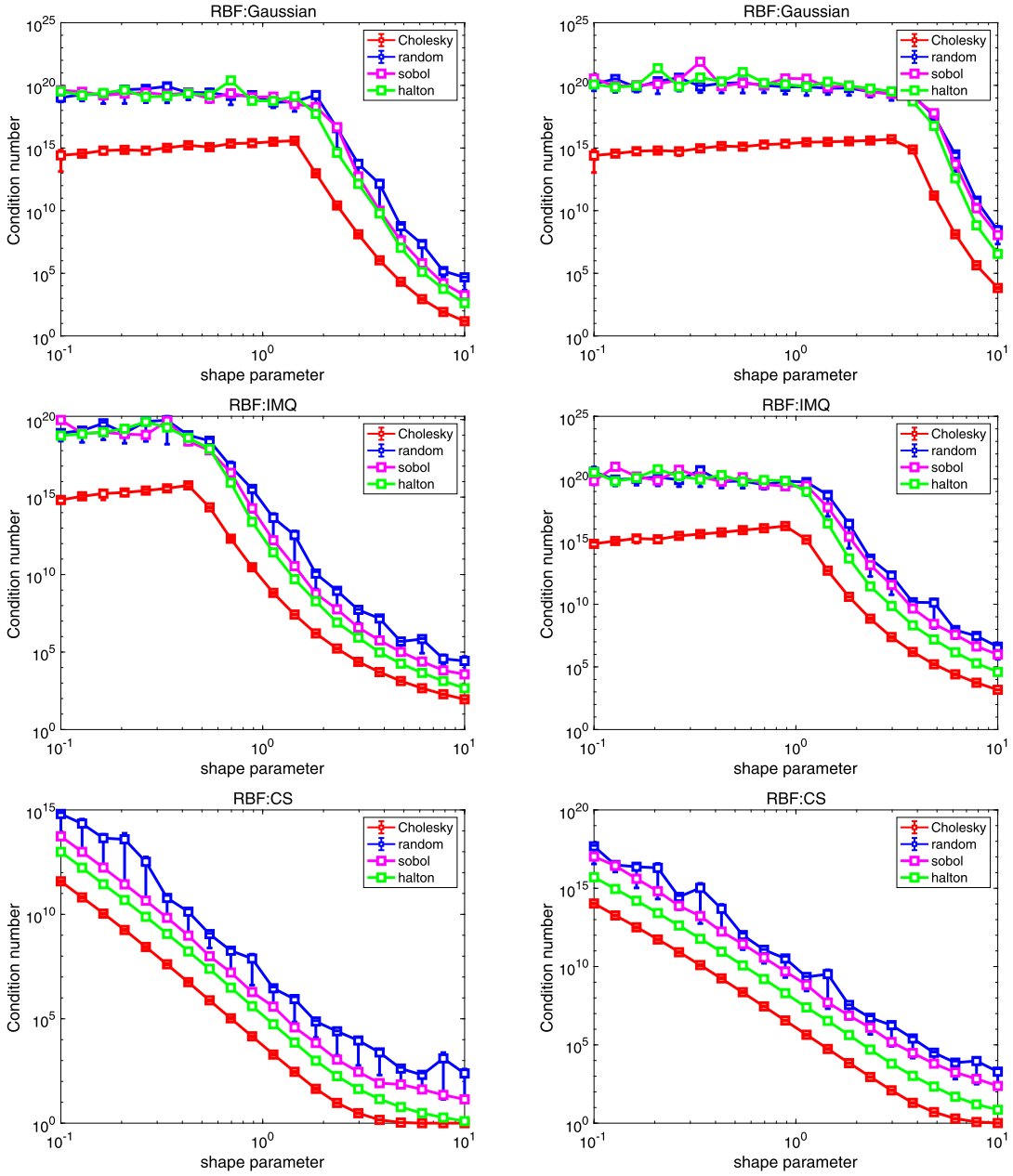

Fig. 3. Condition numbers with respect to shape parameters using Gaussian, IMQ and CS. Left: $N = 100$; right: $N = 300$; $d = 2$.

In what follows we will use the term "Cholesky" for our proposed algorithm, "random" to denote random sampling points, "sobol" to denote Sobol points, and "halton" to denote Halton points. Furthermore, in all examples that follow we perform 50 trials of each procedure and report the mean results along with 20% and 80% quantiles. For CS RBFs, we use the following Wendlands compactly supported function $\Phi_d(r)$, which has the form

$$\Phi_d(r) = (1-r)_+^{l+3}\Big((l^3+9l^2+23l+15)r^3+(6l^2+36l+45)r^2+(15l+45)r+15\Big)/15$$

with $l = \left\lfloor \frac{d}{2} \right\rfloor + 4$.

4.1. Kernel interpolation for UQ

In this section, some numerical examples are shown to demonstrate the stability and convergent behaviors of our proposed algorithms for UQ.

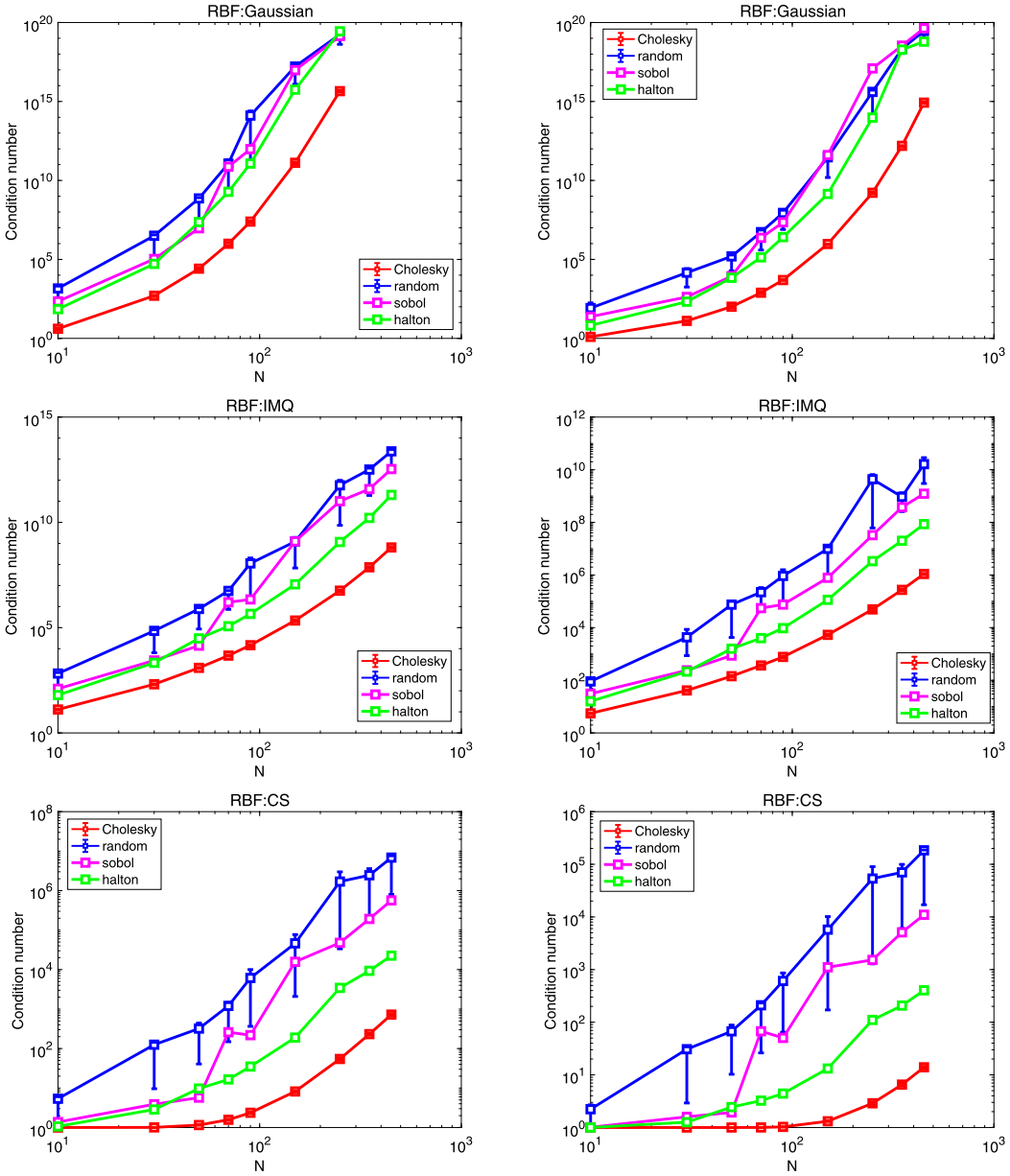

Fig. 4. Condition numbers with respect to shape parameters with respect to the number of sample points N using Gaussian, IMQ and CS. Left: $\epsilon = 3$; right: $\epsilon = 5$; $d = 2$.

4.1.1. Matrix stability

We first investigate the condition number $Cond(\mathbf{A}) = \frac{\sigma_{max}(\mathbf{A})}{\sigma_{min}(\mathbf{A})}$ of the design matrix $\mathbf{A}$. In Fig. 3, we show the condition number with respect to the different shape parameters. In Fig. 4, we show the condition number with respect to the number of the selected RBF centers N. For both Gaussian and IMQ, our algorithm is much more stable compared to the other sampling methods. In fact, the other choice of the candidates points do not dramatically affect the performance of the proposed method. Fig. 5 show the corresponding results. In the figure, "uniform" utilizes the evenly spaced points in I_Z.

4.1.2. Numerical accuracy

In this section, we will compare Cholesky, random, Sobol and Halton in terms of their ability to approximate test function.

We first consider the benchmark Franke's function $u(z), z = (z_1, z_2) \in [0, 1]^2$, given by

$$u(z) = \frac{3}{4}e^{-((9z^{(1)}-2)^2+(9z^{(2)}-2)^2)/4} + \frac{3}{4}e^{-(9z^{(1)}+1)^2/49-(9z^{(2)}+1)^2)/10} + \frac{1}{2}e^{-((9z^{(1)}-7)^2+(9z^{(2)}-3)^2)/4} - \frac{1}{5}e^{-(9z^{(1)}-4)^2-(9z^{(2)}-7)^2}. \tag{13}$$

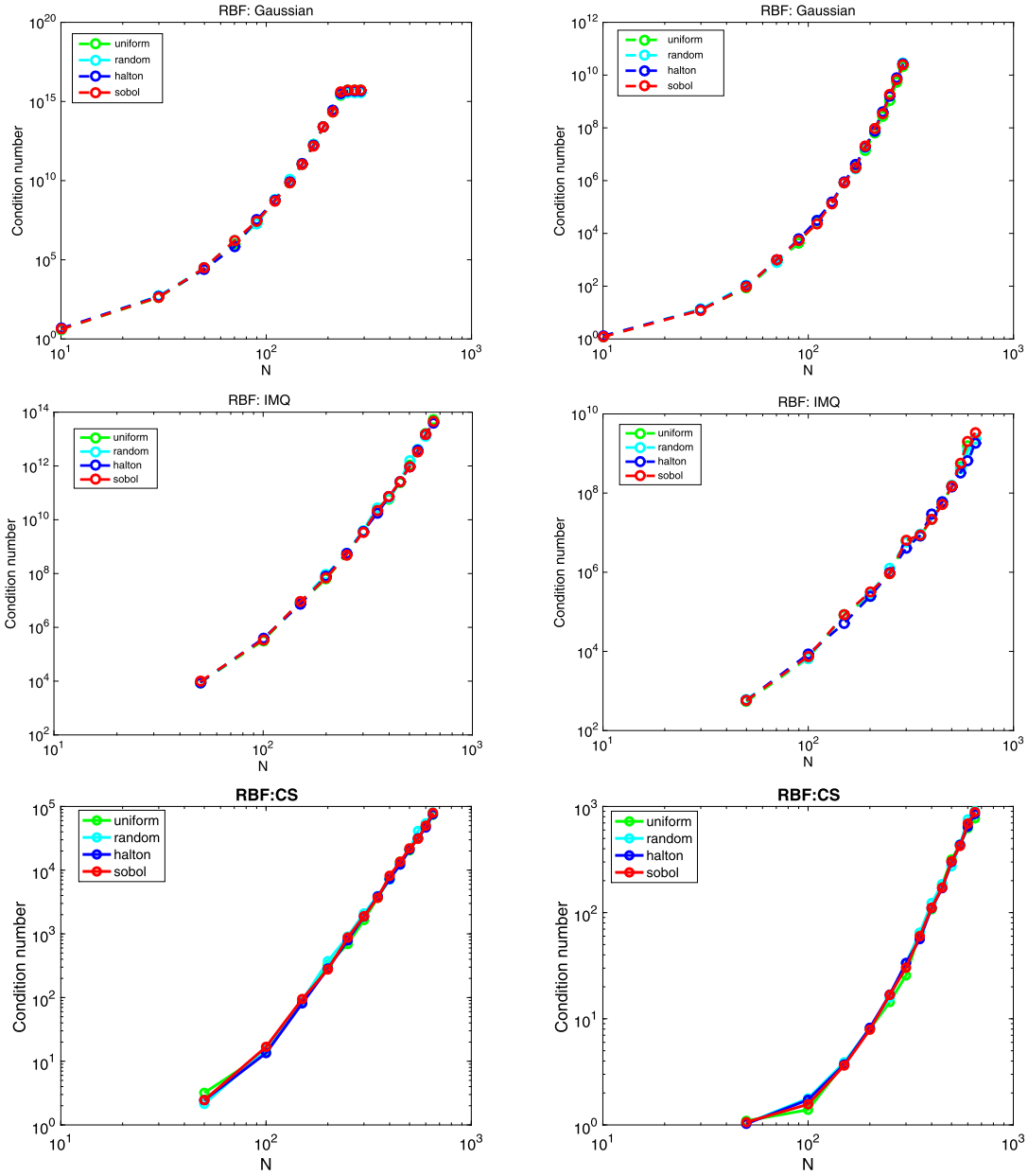

Fig. 5. Condition number with respect to the number of sample points N for different choice of the candidates. Left: $\epsilon = 3$; right: $\epsilon = 5$.

The corresponding numerical errors for the test function in (13) are shown in Fig. 6. We observe that our proposed algorithm produces superior results than the other sampling methods. Moreover, the new algorithm shows clear convergence patterns as N increases. This is considered to be an improvement over the other sampling methods: it is clear in the error profile of the other sampling methods that providing more sampling points does not always result in better accuracy. Again, the choice of the candidates points do not dramatically affect the performance of the proposed method. Fig. 7 show the corresponding results.

In Fig. 8 we show the numerical results as a function of the sample points, N, obtained using the Cholesky algorithm with various values of the shape parameter ϵ. The left plot shows the condition number while the right plot shows the corresponding errors. It can be seen from this figure that as ϵ gets smaller, the resulting matrix becomes ill-conditioned. The numerical results when the LOOCV algorithm is used are also presented in Fig. 8. From these figures, we observe that the Cholesky algorithm using LOOCV to choose the good value of shape parameter can provide a very good numerical results.

We now consider the stochastic elliptic equation, one of the most used benchmark problems in UQ, in one spatial dimension,

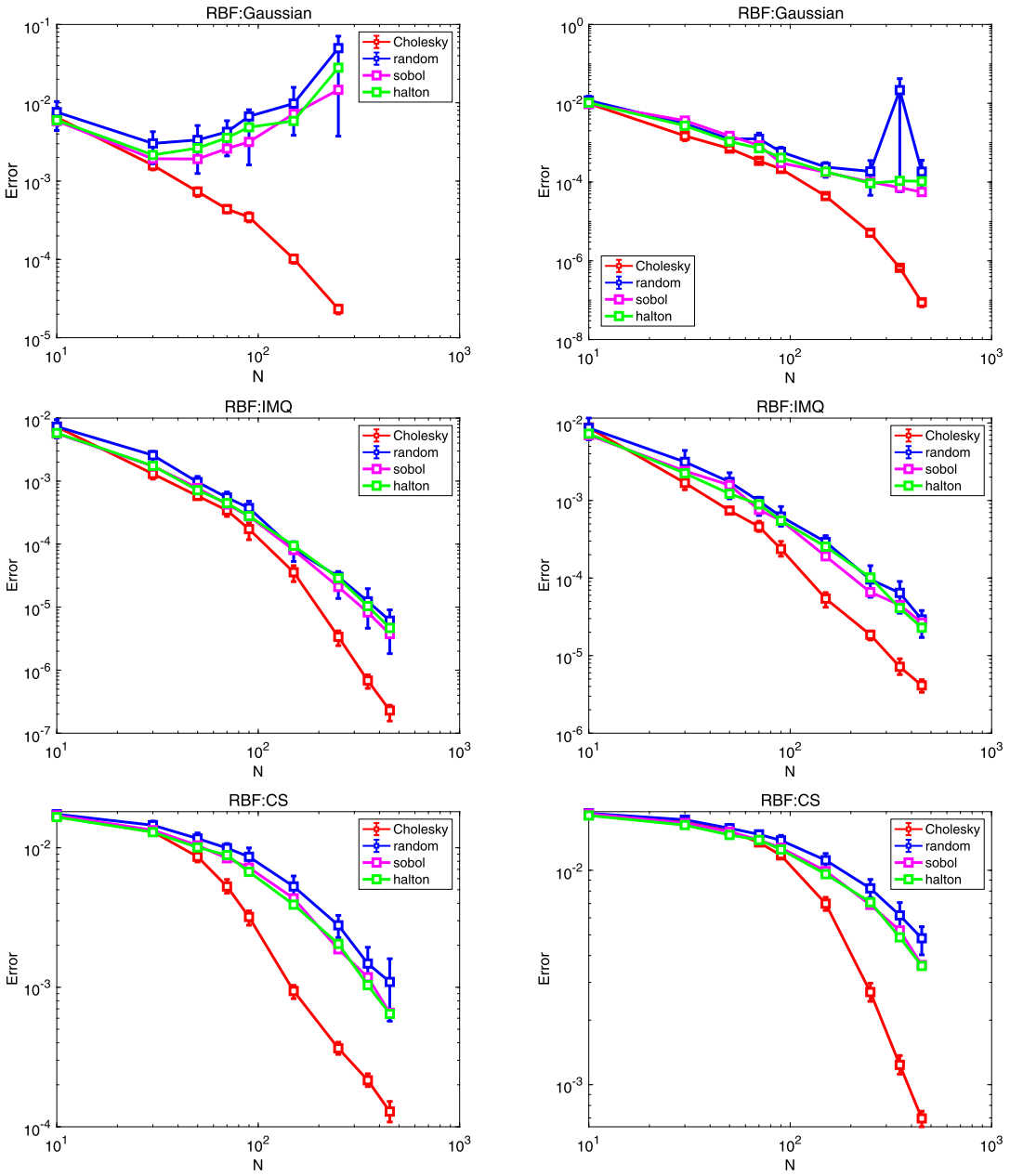

Fig. 6. Approximation error with respect to the number of sample points N using Gaussian, IMQ and CS. Left: $\epsilon = 3$; right: $\epsilon = 5$.

$$-\frac{d}{dx}[\kappa(x,z)\frac{du}{dx}(x,z)] = f, \quad (x,z) \in (0,1) \times \mathbb{R}^d, \tag{14}$$

with boundary conditions

$$u(0,z) = 0, \quad u(1,z) = 0,$$

and $f = 2$. The random diffusivity takes the following form

$$\kappa(x,z) = 1 + \sigma \sum_{k=1}^{d} \frac{1}{k^2\pi^2} \cos(2\pi kx) z^{(k)}, \tag{15}$$

where $z = (z^{(1)}, \ldots, z^{(d)})$ is a random vector with independent and identically distributed components.

Here we approximate the solution $u(z) = u(0.5, z)$, while the uncertain inputs $z^{(i)} \sim U[-1,1], k = 1, \cdots, d$. For further comparison, we also employ the sparse grids stochastic collocation method using Legendre polynomial of total order

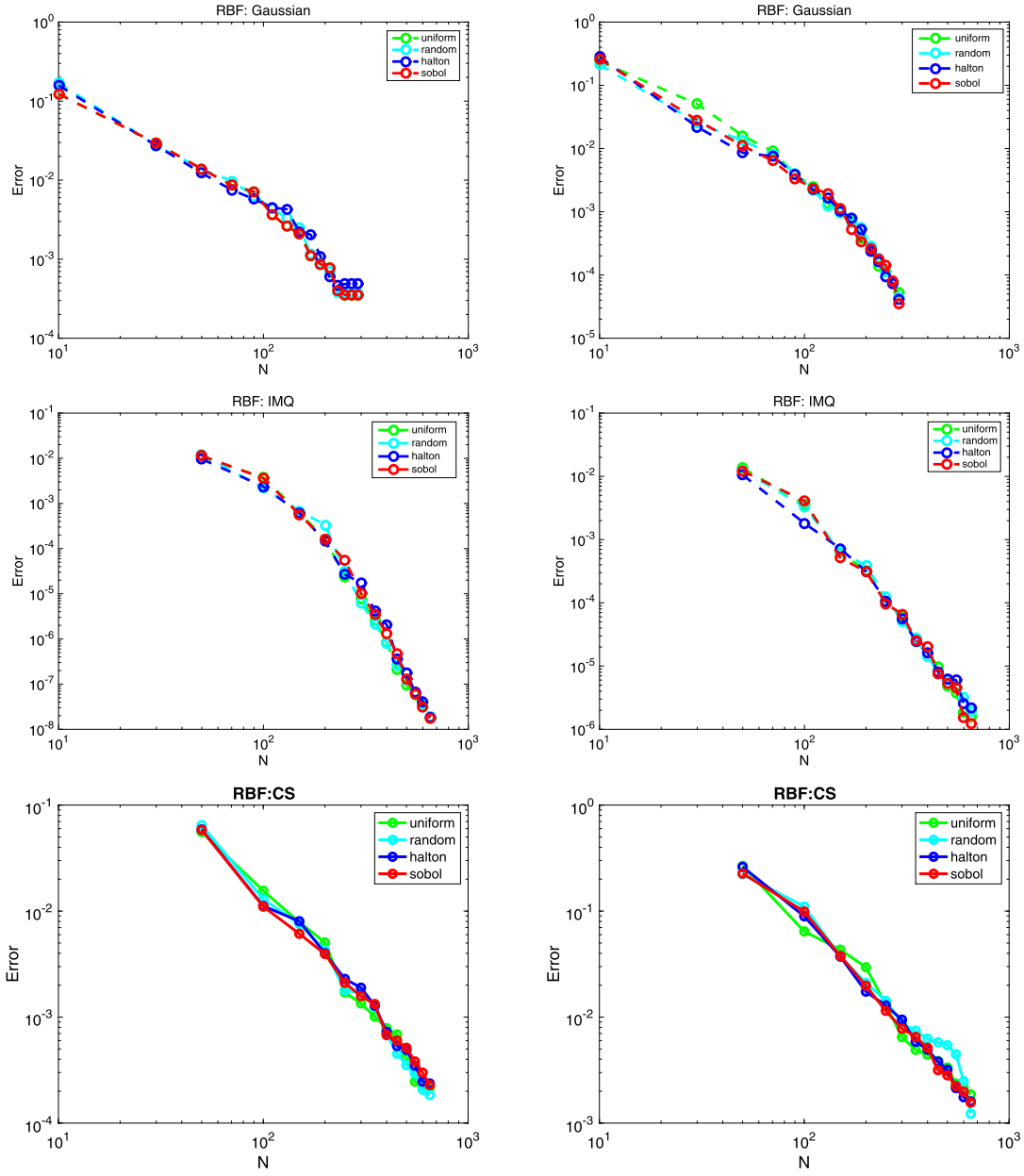

Fig. 7. Numerical results with respect to the number of sample points N for different choice of the candidates: Left: $\epsilon = 3$; right: $\epsilon = 5$.

$k = 8(d = 3)$ and $k = 4(d = 6)$ to solve the problem. The numerical results are presented in Figs. 9 and 10. The RBFs approximation methods, are notably superior to the sparse grids method.

4.2. Gradient-enhanced approach for UQ

We now illustrate the symmetric approach to Hermite interpolation with a set of numerical experiments.

Example 1. We use the corner peak

$$u(z) = (1 + \sum_{i=1}^{d} \omega_i z^{(i)})^{-(d+1)}, \; z \in \Omega = [0, 1]^d$$

to generate the data. In this example, we use $d = 2, \omega_i = \frac{1}{i^2}$.

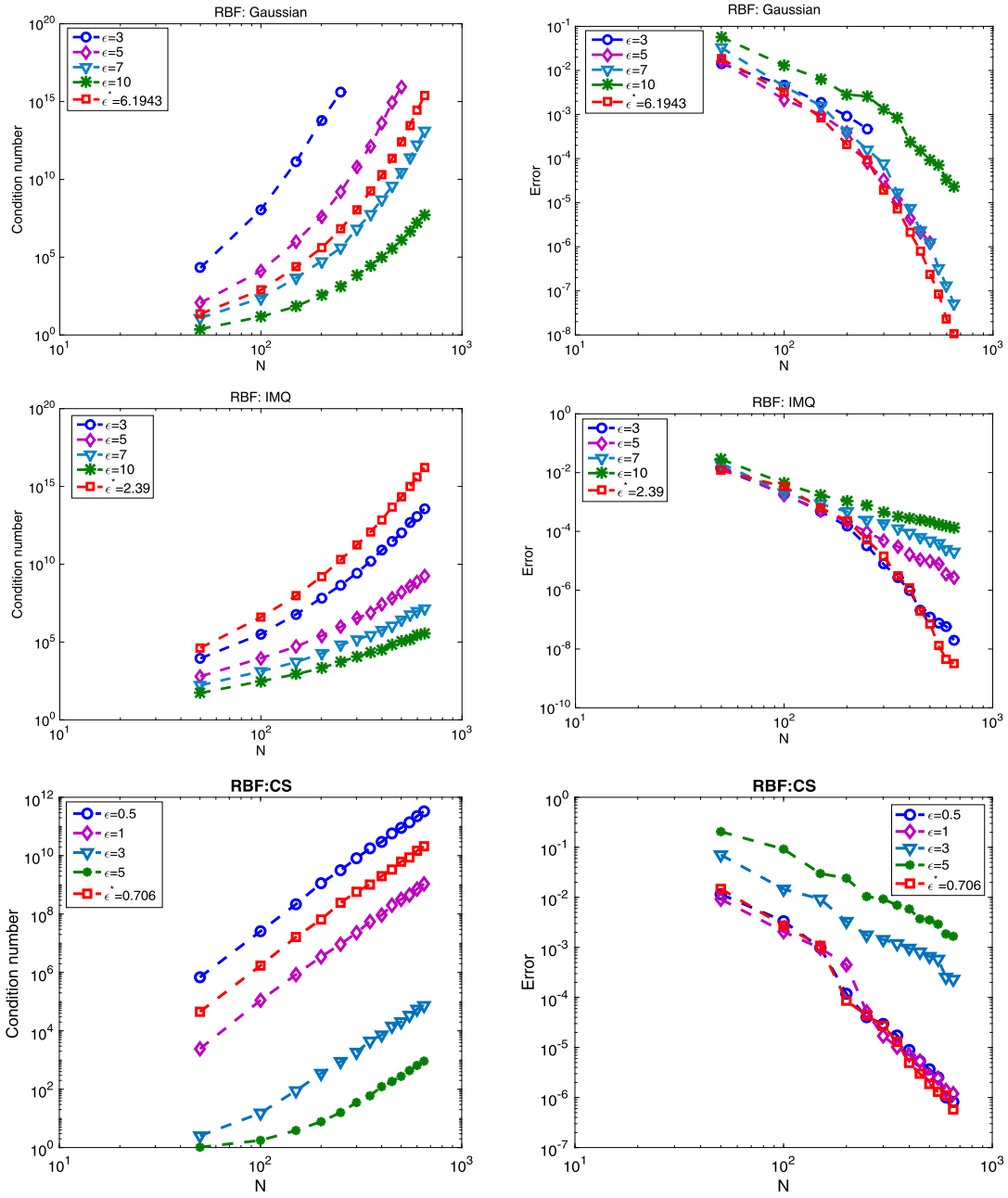

Fig. 8. Numerical results with respect to the number of sample points N for different shape parameters. Left: condition number; right: errors.

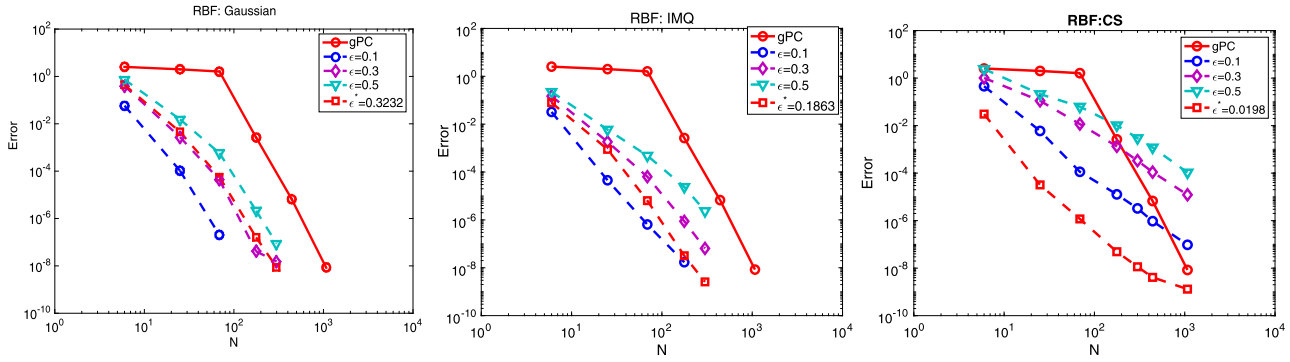

Fig. 9. Approximation error against the number of sample points N (d=3). Left: Gaussian; Middle: IMQ; right: CS.

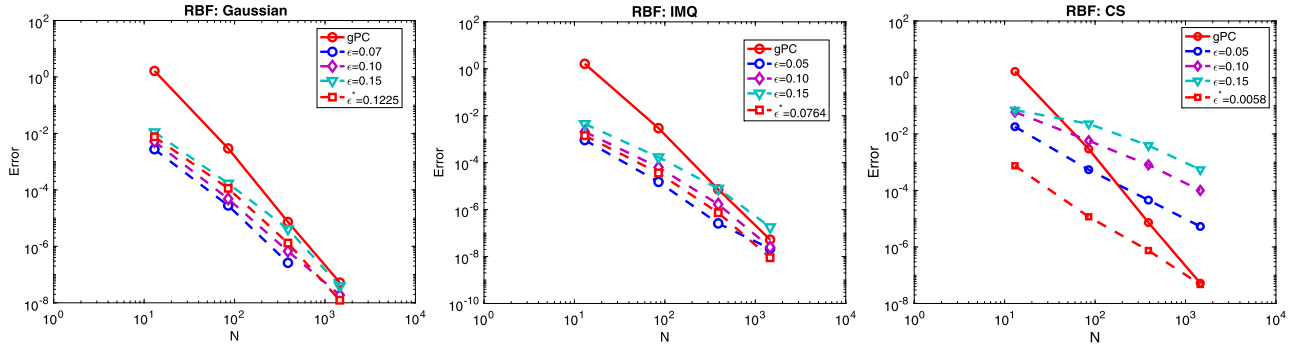

Fig. 10. Approximation error against the number of sample points N (d=6). Left: Gaussian; middle: IMQ; right: CS.

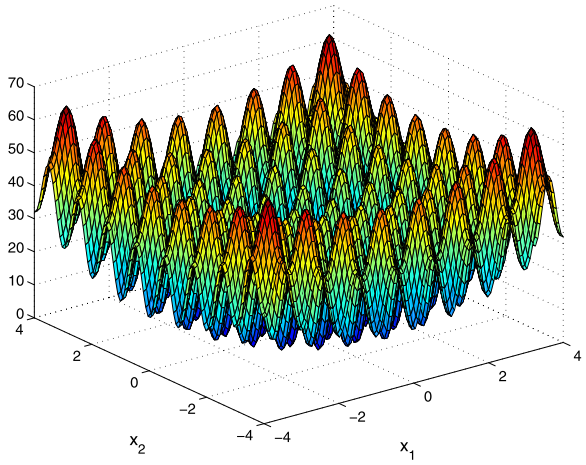

Fig. 11. 2-dimensional Rastrigin function.

Example 2. We consider the following 2-dimensional Rastrigin function:

$$u(z) = 20 + \sum_{i=1}^{2} \left((z^{(i)})^2 - 10\cos(2\pi z^{(i)}) \right), \quad z^{(i)} \in [-4, 4].$$

The 2-D Rastrigin function is plotted in Fig. 11.

Example 3. Consider the following 5-dimensional Friedman function

$$u(z) = 10\sin(\pi z^{(1)} z^{(2)}) + 20(z^{(3)} - 0.5)^2 + 10z^{(4)} + 5z^{(5)}.$$

The corresponding numerical results are shown in Figs. 12–14. We can conclude that the design matrix **B** can be well conditioned under the proposed algorithm. Again, our proposed algorithm produces superior results than the other sampling methods.

5. Summary

In this work, we presented a strategy for selecting a quasi-optimal sample points for kernel interpolation. We first demonstrate that the traditional sampling methods suffer from the numerical instability in the sense that the condition number will increase fast as the total number of points becomes large. To improve this, we propose to use the efficient Cholesky decomposition with pivoting for the Vandermonde-like interpolation matrix to choose the sample points. Then the quasi-optimal sample points are used to control the condition number of the design matrices. It is demonstrated that with the new approach the stability can be much improved. On the other hand, for problems inclusion of gradient measurements in the kernel interpolation are also considered. Applications to parametric UQ problems are illustrated.

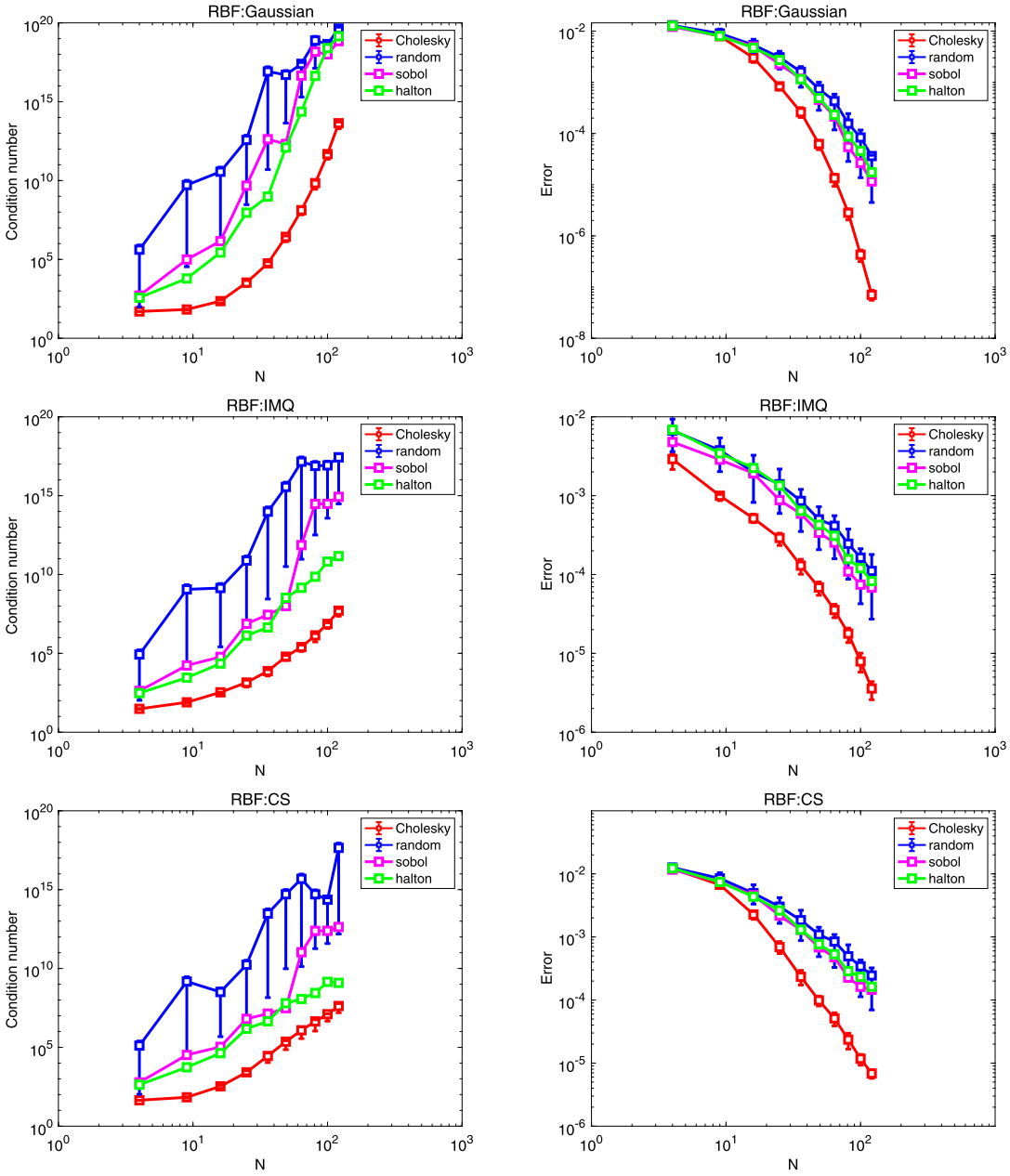

Fig. 12. Numerical results with respect to the number of sample points N for 2-dimensional corner peak.

CRediT authorship contribution statement

Akil Narayan: Conceptualization, Formal analysis, Investigation, Methodology, Validation, Writing – review & editing. **Liang Yan:** Conceptualization, Investigation, Methodology, Software, Writing – original draft, Writing – review & editing. **Tao Zhou:** Conceptualization, Investigation, Methodology, Validation, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

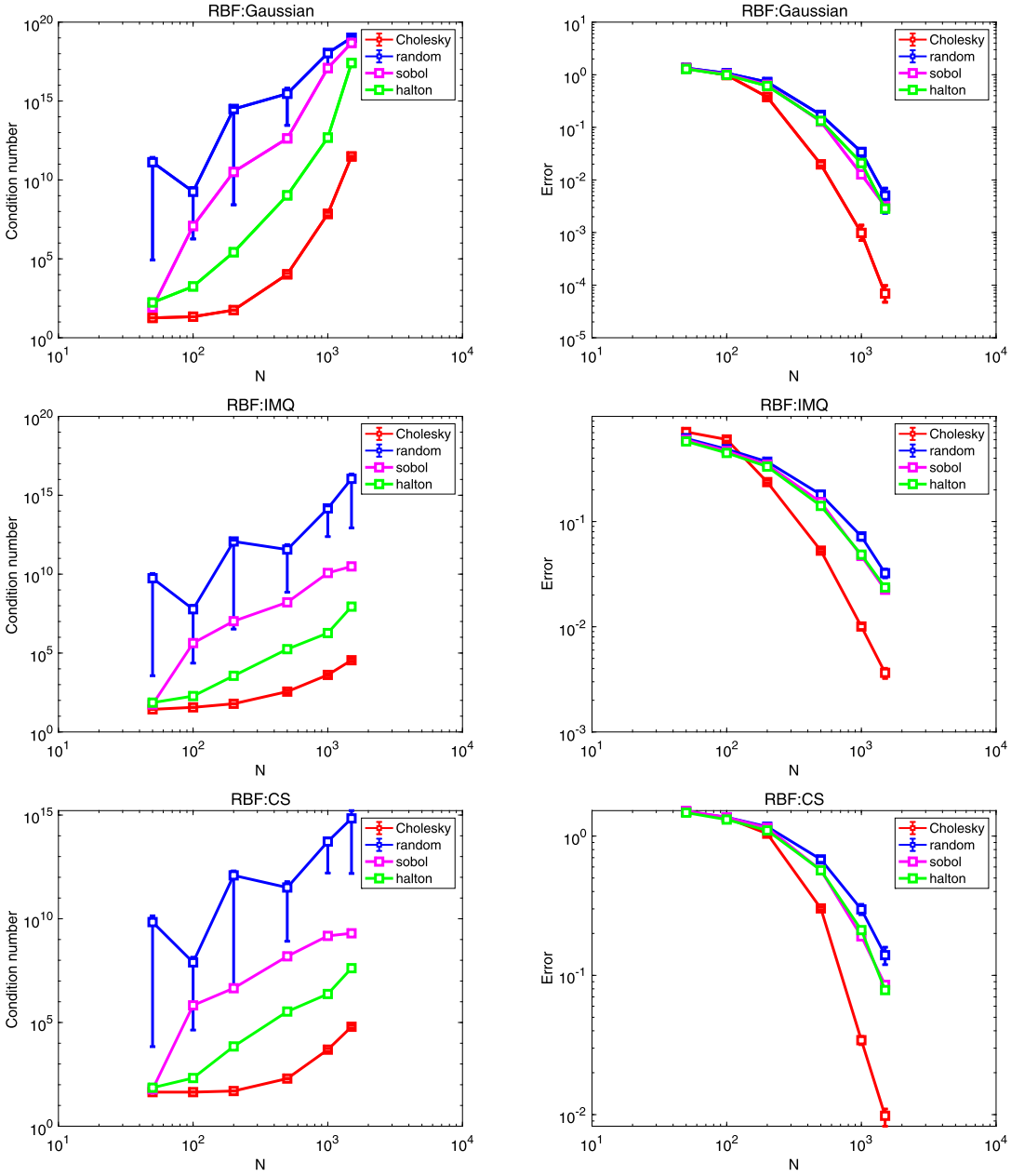

Fig. 13. Numerical results with respect to the number of sample points N for 2-dimensional Rastrigin function.

Acknowledgements

The first author is partially supported by NSF DMS-1848508 and AFOSR FA9550-20-1-0338. The second author's work is partially supported by NSF of China (No. 11771081) and the Southeast University Zhishan Young Scholars Program. The last author is partially supported by the National Key R&D Program of China (No. 2020YFA0712000), NSF of China (under grant numbers 11822111, 11688101 and 11731006), Science Challenge Project (No. TZ2018001), the Strategic Priority Research Program of Chinese Academy of Sciences (Grant No. XDA25000404) and Youth Innovation Promotion Association, CAS.

Appendix A. Gaussian process and kernel interpolation

In order to help readers get deeper insights into the connections between kernel methods and Gaussian process regressions, we give a simple example in this section. More detains can be found in [30].

Suppose that we have a set of observed data values $\{(z_j, u_j)_{j=1}^N | z_j \in \Xi\}$. We aim to estimate the unknown value at some other locations Z based on the observed data at Ξ. In the Gaussian process regression [24,30], a.k.a. the simple kriging

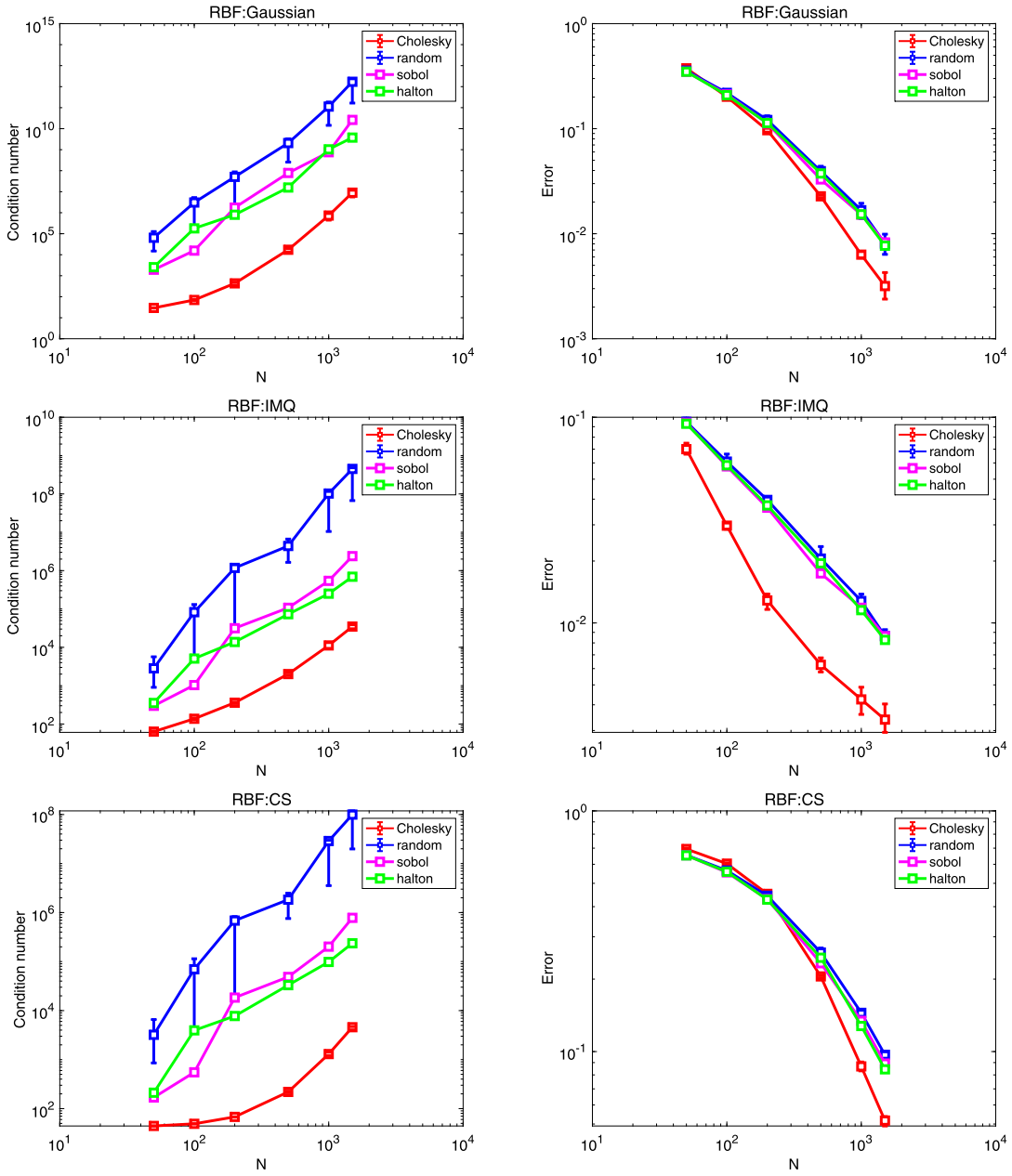

Fig. 14. Numerical results with respect to the number of sample points N for 5-dimensional Friedman function.

method, u is assumed as a realization of a random field $\mathbf{Y}$, which is a collection $\{\mathbf{Y}(Z) : Z \in I_Z\}$ of random variables over the probability space $(\Omega, \mathcal{F}, \mathcal{P})$. The observations $\{u(z_j)\}$ are then realizations of the random variables $\mathbf{Y}(z_j)$. To predict $\mathbf{Y}$ at some location $Z \in I_Z$, one considers all linear predictors of the form

$$\mathbf{Y}(Z) = \sum_{i=1}^{N} \omega_i(Z)\mathbf{Y}(z_i)$$

which are themselves random variables.

To determine optimal weights $\omega(\mathbf{Z}) := (\omega_1(Z), \cdots, \omega_N(Z))^T$, additional structural assumptions on $\mathbf{Y}$ are needed. In this work, we only discuss the simple kriging, due to their close connection to kernel interpolation. Suppose $\mathbf{Y}$ is a Gaussian process with a mean 0 and symmetric covariance kernel K, i.e. $\mathbb{E}(\mathbf{Y}(Z)) = 0$, $Cov(\mathbf{Y}(z_i), \mathbf{Y}(z_j)) = K(z_i, z_j) := \phi(z_i - z_j)$. We usually assume that such covariance kernel K is positive definite. The simple kriging method provides the best linear unbiased estimator $\hat{f}$ of $\mathbf{Y}$ in the form of

$$\hat{f} := \omega_1(Z)u_1 + \cdots + \omega_N(Z)u_N$$

where the weighting $\omega(\mathbf{Z})$ is uniquely determined by the linear system ([30])

$$\begin{bmatrix} K(z_1,z_1) & \cdots & K(z_1,z_N) \\ \vdots & \ddots & \vdots \\ K(z_N,z_1) & \cdots & K(z_N,z_N) \end{bmatrix} \begin{bmatrix} \omega_1(Z) \\ \vdots \\ \omega_N(Z) \end{bmatrix} = \begin{bmatrix} K(Z,z_1) \\ \vdots \\ K(Z,z_N) \end{bmatrix}$$

or in matrix form $K(\Xi,\Xi)\omega(\mathbf{Z}) = K(Z,\Xi)^T$. Thus, the estimator $\hat{f}(Z)$ can be written as

$$\hat{f}(Z) := \omega(\mathbf{Z})\mathbf{u} \tag{A.1}$$

Note that if the RBF kernel and the covariance kernel coincide, then (3) and (A.1) suggest that $u_N(Z) = K(Z,\Xi)^T K(\Xi,\Xi)^{-1}\mathbf{u} = (K(\Xi,\Xi)^{-1}K(Z,\Xi))^T\mathbf{u} = \hat{f}(Z)$. It means that the RBF estimator and the unbiased estimator are indeed identical.

Appendix B. Gradient-enhanced Gaussian process

If the gradient information $\{z_i, u'_m(z_i)\}_{i=1}^N$ $(m = 1, 2, \ldots, d)$ is included in the ordinary GP model, the covariance matrix becomes a block matrix that includes the covariance between derivative observations in addition to the covariance between function observations. The block matrix can be represented as [18]:

$$\mathbf{K} = \begin{bmatrix} Cov(\mathbf{Y},\mathbf{Y}) & Cov(\mathbf{Y},\nabla\mathbf{Y}) \\ Cov(\nabla\mathbf{Y},\mathbf{Y}) & Cov(\nabla\mathbf{Y},\nabla\mathbf{Y}) \end{bmatrix},$$

where $Cov(\mathbf{Y},\mathbf{Y})$ represents an $N\times N$ covariance matrix of the function values at the sample points, $Cov(\mathbf{Y},\nabla\mathbf{Y})$ is an $Nd\times N$ cross covariance matrix between gradient components and the function values at the sample points, and $Cov(\nabla\mathbf{Y},\nabla\mathbf{Y})$ is an $Nd\times Nd$ covariance matrix of the gradients. If $\mathbf{Y}$ is a Gaussian process with a mean 0 and symmetric covariance kernel K (or ϕ), then the joint covariance matrix of the outputs is given by

$$\mathbf{K} = \begin{bmatrix} \mathbf{P}_{0,0} & \mathbf{P}_{0,1} & \cdots & \mathbf{P}_{0,d} \\ \mathbf{P}_{1,0} & \mathbf{P}_{1,1} & \cdots & \mathbf{P}_{1,d} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{d,0} & \mathbf{P}_{d,1} & \cdots & \mathbf{P}_{d,d} \end{bmatrix}$$

where $\mathbf{P}_{k,l}$ is the $N\times N$ matrix with the (i,j)th element

$$\begin{cases} \phi(z_i - z_j), & k = l = 0, \\ -\phi'_l(z_i - z_j), & k = 0, l \neq 0, \\ \phi'_k(z_i - z_j), & k \neq 0, l = 0, \\ -\phi''_{k,l}(z_i - z_j), & k \neq 0, l \neq 0, \end{cases} \tag{B.1}$$

and $\mathbf{K}$ is a symmetric $(d+1)N\times(d+1)N$ matrix.

Note that if the RBF kernel Φ and the covariance kernel ϕ coincide, then the joint covariance matrix $\mathbf{K}$ and the design matrix $\mathbf{B}$ are indeed identical.

References

[1] Ilias Bilionis, Nicholas Zabaras, Multi-output local Gaussian process regression: applications to uncertainty quantification, J. Comput. Phys. 231 (17) (2012) 5718–5746.
[2] Ilias Bilionis, Nicholas Zabaras, Bayesian uncertainty propagation using Gaussian processes, in: Handbook of Uncertainty Quantification, 2016, pp. 1–45.
[3] A. Chkifa, A. Cohen, C. Schwab, High-dimensional adaptive sparse polynomial interpolation and applications to parametric PDEs, Found. Comput. Math. 14 (4) (2014) 601–633.
[4] Jouke H.S. de Baar, Richard P. Dwight, Hester Bijl, Improvements to gradient-enhanced kriging using a bayesian interpretation, Int. J. Uncertain. Quantificat. 4 (3) (2014).
[5] A. Doostan, H. Owhadi, A non-adaptive sparse approximation for PDEs with stochastic inputs, J. Comput. Phys. 230 (8) (2011) 3015–3034.
[6] Gregory Fasshauer, Michael McCourt, Kernel-Based Approximation Methods Using MATLAB, vol. 19, World Scientific Publishing Company, 2015.
[7] Gregory E. Fasshauer, Meshfree Approximation Methods with MATLAB, Interdisciplinary Mathematical Sciences., vol. 6, World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2007.
[8] Gregory E. Fasshauer, Jack G. Zhang, On choosing "optimal" shape parameters for RBF approximation, Numer. Algorithms 45 (1–4) (2007) 345–368.
[9] B. Ganapathysubramanian, N. Zabaras, Sparse grid collocation methods for stochastic natural convection problems, J. Comput. Phys. 225 (1) (2007) 652–685.
[10] R.G. Ghanem, P.D. Spanos, Stochastic Finite Elements: A Spectral Approach, Springer-Verlag, Inc., New York, 1991.
[11] L. Guo, A. Narayan, L. Yan, T. Zhou, Weighted approximate Fekete points: sampling for least-squares polynomial approximation, SIAM J. Sci. Comput. 40 (1) (2018) A366–A387.

[12] Ling Guo, Yongle Liu, Akil Narayan, Tao Zhou, Sparse approximation of data-driven PCEs: an induced sampling approach, Commun. Math. Res. 36 (2020) 128–153.
[13] Ling Guo, Akil Narayan, Tao Zhou, Constructing least-squares polynomial approximations, SIAM Rev. 62 (2) (2020) 483–508.
[14] Xu He, Peter Chien, On the instability issue of gradient-enhanced Gaussian process emulators for computer experiments, SIAM/ASA J. Uncertain. Quantificat. 6 (2) (2018) 627–644.
[15] Y.C. Hon, R. Schaback, X. Zhou, An adaptive greedy algorithm for solving large RBF collocation problems, Numer. Algorithms 32 (1) (2003) 13–25.
[16] J. Jackman, A. Narayan, T. Zhou, A generalized sampling and preconditioner scheme for sparse approximation of polynomial chaos expansions, SIAM J. Sci. Comput. A 39 (3) (2017) 1114–1144.
[17] Leevan Ling, Robert Schaback, An improved subspace selection algorithm for meshless collocation methods, Int. J. Numer. Methods Eng. 80 (13) (2009) 1623–1639.
[18] Brian A. Lockwood, Mihai Anitescu, Gradient-enhanced universal kriging for uncertainty propagation, Nucl. Sci. Eng. 170 (2) (2012) 168–195.
[19] Charles A. Micchelli, Interpolation of scattered data: distance matrices and conditionally positive definite functions, Constr. Approx. 2 (1) (1986) 11–22.
[20] Max D. Morris, Toby J. Mitchell, Donald Ylvisaker, Bayesian design and analysis of computer experiments: use of derivatives in surface prediction, Technometrics 35 (3) (1993) 243–255.
[21] A. Narayan, J. Jakeman, Adaptive Leja sparse grid constructions for stochastic collocation and high-dimensional approximation, SIAM J. Sci. Comput. 36 (6) (January 2014) A2952–A2983.
[22] A. Narayan, J.D. Jakeman, T. Zhou, A Christoffel function weighted least squares algorithm for collocation approximations, Math. Comput. 86 (2017) 1913–1947.
[23] F. Nobile, R. Tempone, C.G. Webster, An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data, SIAM J. Numer. Anal. 46 (5) (January 2008) 2411–2442.
[24] Carl Edward Rasmussen, Christopher K.I. Williams, Gaussian Process for Machine Learning, MIT Press, 2006.
[25] Shmuel Rippa, An algorithm for selecting a good value for the parameter c in radial basis function interpolation, Adv. Comput. Math. 11 (2–3) (1999) 193–210.
[26] Jerome Sacks, Susannah B. Schiller, William J. Welch, Designs for computer experiments, Technometrics 31 (1) (1989) 41–47.
[27] Robert Schaback, Holger Wendland, Adaptive greedy techniques for approximate solution of large RBF systems, Numer. Algorithms 24 (3) (2000) 239–254.
[28] Robert Schaback, Holger Wendland, Kernel techniques: from machine learning to meshless methods, Acta Numer. 15 (2006) 543–639.
[29] Michael Scheuerer, An alternative procedure for selecting a good value for the parameter c in RBF-interpolation, Adv. Comput. Math. 34 (1) (2011) 105–126.
[30] Michael Scheuerer, Robert Schaback, Martin Schlather, Interpolation of spatial data – a stochastic or a deterministic problem?, Eur. J. Appl. Math. 24 (4) (2013) 601–629.
[31] Rohit Tripathy, Ilias Bilionis, Marcial Gonzalez, Gaussian processes with built-in dimensionality reduction: applications to high-dimensional uncertainty propagation, J. Comput. Phys. 321 (2016) 191–223.
[32] Selvakumar Ulaganathan, Ivo Couckuyt, Tom Dhaene, Joris Degroote, Eric Laermans, Performance study of gradient-enhanced kriging, Eng. Comput. 32 (1) (2016) 15–34.
[33] Holger Wendland, Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, Adv. Comput. Math. 4 (4) (1995) 389–396.
[34] Holger Wendland, Scattered Data Approximation, Cambridge Monographs on Applied and Computational Mathematics, vol. 17, Cambridge University Press, Cambridge, 2005.
[35] D. Xiu, J.S. Hesthaven, High-order collocation methods for differential equations with random inputs, SIAM J. Sci. Comput. 27 (3) (2005) 1118–1139.
[36] D. Xiu, G.E. Karniadakis, The Wiener-Askey polynomial chaos for stochastic differential equations, SIAM J. Sci. Comput. 24 (2) (2002) 619–644.
[37] L. Yan, L. Guo, D. Xiu, Stochastic collocation algorithms using ℓ_1-minimization, Int. J. Uncertain. Quantificat. 2 (3) (2012) 279–293.
[38] F.L. Yang, L. Yan, L. Ling, Doubly stochastic radial basis function methods, J. Comput. Phys. 363 (2018) 87–97.