IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 18, 2020, accepted April 23, 2020, date of publication April 27, 2020, date of current version May 13, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2990731

A Fast Solution for the Generalized Radial Basis Functions Interpolant

DEYUN ZHONG, LIGUAN WANG, AND LIN BI
School of Resources and Safety Engineering, Central South University, Changsha 410083, China
Research Center of Digital Mine, Central South University, Changsha 410083, China

Corresponding author: Lin Bi (mr.bilin@csu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFC0602905, and in part by the National Natural Science Foundation of China under Grant 41572317.

ABSTRACT In this paper, we propose a fast solution method of the generalized radial basis functions interpolant for global interpolation. The method can be used to efficiently interpolate large numbers of geometry constraints for implicit surface reconstruction. The basic idea of our approach is based on the far field expansion of the kernel and the preconditioned Krylov iteration using the domain decomposition method as a preconditioner. We present a fast evaluation method of the matrix-vector product for the linear system. To minimize the number of iterations for large numbers of constraints, the multi-level domain decomposition method is applied to improve overlap using a nested sequence of levels. The implemented solution algorithm generally achieves $O(NlogN)$ complexity and $O(N)$ storage. It is kernel independent both in the evaluation and solution processes without analytical expansions. It is very convenient to apply various types of RBF kernels in different applications without excessive modifications to the existing process. Numerical results show that the fast evaluation method has a good performance for the evaluation of the matrix-vector product and the preconditioned Krylov subspace iterative method has a good convergence rate with a small number of iterations.

INDEX TERMS Radial basis functions, generalized radial basis functions, far field expansion, domain decomposition method, implicit surface reconstruction.

I. INTRODUCTION

Radial basis functions (RBFs) are widely used in large-scale scattered data interpolation and approximation. Applications include surface reconstruction of dense point clouds, solution of partial differential equations and particle interactions in computational physics.

The generalized radial basis functions (GRBF) interpolant [1]–[3] is initially developed to exactly interpolate the Hermite data (a set of points with normals). Various types of the interpolation constraints, including gradient constraints and tangent constraints, are developed to satisfy different application needs [4]–[6]. As a special case of the GRBF interpolant, the Hermite radial basis functions (HRBF) [7] can exactly interpolate the gradient directions of an implicit surface. However, the direct solution methods of the interpolation equation cost $O(N^3)$ complexity and $O(N^2)$ storage. For a large number of constraints (more than a few thousand points), it is too expensive to solve the interpolation equation using a globally supported basis function.

In recent years, great progress has been made to efficiently solve the large-scale linear system for the RBF-based methods. The domain decomposition-based iteration method (e.g., the Fast RBF method [8]) is a kind of efficient solution method. The optimal domain decomposition strategy is helpful to improve the convergence of iteration. However, as the evaluation process is different, the existing global interpolation methods cannot be applied to the generalized radial basis functions interpolant directly. The previous works focus on solving the radial basis functions interpolant only with domain constraints (the constraints with specified function values). Besides the domain constraints, the directional constraints (e.g., the gradient constraints and the tangent constraints) should be interpolated efficiently. It is necessary to develop efficient methods for the evaluation and solution of GRBF.

In this paper, based on a multilevel domain decomposition method, we develop a fast global solution method for

The associate editor coordinating the review of this manuscript and approving it for publication was Jiju Poovvancheri.

the GRBF interpolant with globally supported radial basis functions. It is an extension of the fast radial basis functions interpolation method. Similar to the fast radial basis functions interpolation method, the idea of this method is based on the far field expansion of the RBF kernel and the preconditioned Krylov iteration using the domain decomposition method as a preconditioner.

To solve the dense linear system efficiently, a fast evaluation method of the matrix-vector product for GRBF is presented. We expand the evaluation of the domain constraints and the difference constrains into several sums with different source points and evaluation points respectively. Each sum is efficiently evaluated using the black-box fast multipole method (FMM) [9] for a wide class of RBF kernels. Then the whole evaluation of the matrix-vector product consists of the evaluations of several sub-matrix blocks. Based on the fast evaluation method, the preconditioned Krylov subspace method is used to solve the linear system iteratively. The flexible generalized minimal residual (FGMRES) method [10] with variable preconditioners is used as the outer iterative method. And the result of the domain decomposition method at each iteration step is used as the variable preconditioners. To minimize the number of iterations for large numbers of constraints, the multi-level domain decomposition method is applied to improve overlap using a nested sequence of levels.

We apply this method to efficiently recover an implicit surface using the GRBF interpolant with large numbers of geometry constraints. The interpolation constraints are converted into geometry constraints by constructing a signed distance field (SDF) [11]. As an example, the Hermite data can be approximated using the domain constraints and the difference constraints of the gradient [5]. Numerical results show that the solution process converges with a small number of iterations by specifying a precision.

The paper is organized as follows. The previous works are studied in Section II. Section III gives a brief description of the GRBF interpolant. In Section IV, the evaluation of GRBF is decomposed into several fast summations with different source points. Based on the fast evaluation method, we use a multilevel domain decomposition method as a preconditioner to efficiently solve the linear system in Section V. In Section VI, the fast solution method is implemented to validate the performance and optimal parameters of the algorithm. The limitations and extensions of the method are discussed in Section VIII.

II. RELATED WORKS

In the past two decades, a number of efficient numerical methods such as radial basis functions have been proposed for scattered data interpolation. One of the common approaches is to adjust the support radius as a function of the data density using the compactly supported basis functions (e.g., Wendland's CSRBF [12]). There are several ways for local interpolation. One of the extensions is to decompose the set of interpolation centers into a nested sequence of subsets using a multilevel method [13]. The advantage of this approach is that the compactly supported basis functions lead to a sparse linear system by ignoring the effects of farther interpolation centers and it is easy for implementation [14]. Moreover, these local interpolation methods can be simply applied to the GRBF interpolant. As an example, Liu *et al.* [15] proposed a fast HRBF interpolation method by automatically adjusting the support sizes of radial basis functions. However, the compact support bases are inferior to the global bases in some applications. The approximation of ignoring the effects of farther interpolation centers may lead to an impact that is difficult to estimate, especially in sparse and uneven data environments. In fact, as a more general approach, the compact support bases can be directly used in the global interpolation method.

For the global bases, it is necessary to implement a fast evaluation for computing matrix-vector product. Based on the idea of low rank approximation, the kernel-based summation can be efficiently evaluated using far field expansion, which is known as the fast multipole method [16], [17]. The fast multipole method was originally developed for the fast summation of the potential fields. At present, a number of expansion methods have been proposed for the RBF kernels, including polyharmonic splines [18], generalized multiquadrics [18] and thin-plate splines [20]. The fast Gauss transform (FGT) [21] is used to compute the summation for Gaussian-type kernels. The expansions of these classical methods depend on the kernel in an analytic way. In recent years, the kernel independent FMM methods are developed to expand a wide class of kernels. Ying [22] proposed a FMM method which can expand various types of RBF kernels in both two and three dimensions. More recently, the black-box FMM method proposed by Fong and Darve [9] can expand the kernel using only the kernel values without analytical expansions.

In the past two decades, several efficient solution methods for the RBF interpolation with globally supported radial basis functions have been proposed. These methods usually utilize the preconditioned Krylov subspace iterative method (e.g., the conjugate gradient method or the generalized minimal residual method). Note that many traditional preconditioners cannot be used directly for GRBF. The preconditioner should satisfy the matrix-free environment as the linear system isn't stored explicitly. The development of some preconditioning strategies for RBF greatly improves the convergence of the iteration. There are two kinds of preconditioning technology for RBF. One is to change for a better basis using the approximate cardinal basis functions (ACBFs) [23], the related researches can be found in [24]–[26]. The other is to decompose the whole domain into subdomains using the alternating projection method. As an example, Beatson *et al.* [27] proposed a two-level domain decomposition method to improve the convergence. Recently, Yokota *et al.* [28] developed a parallel algorithm for radial basis functions interpolation using the restricted additive Schwarz method (RASM) as a preconditioner. The strategy of domain decomposition can be also applied in the meshfree method [29].

As the efficient global evaluation of the GRBF interpolant is difficult, the common approach is to solve the interpolation equation using the compactly supported basis functions. Wendland [30] studied the fast evaluation of large generalized interpolation problems based on the far field expansion. However, the derivation of the kernel is complex and kernel dependent. Considering the difference constraints can be used to approximate the differential constraints, we try to evaluate the GRBF interpolant efficiently by expanding the corresponding constraints into several sums. It provides a novel approach for the fast global solution of the generalized radial basis functions interpolant.

III. GRBF INTERPOLANT

The GRBF interpolant is built upon the theory of Hermite-Birkhoff interpolation with RBFs [3]. To exactly interpolate the different values between unknown function values, the difference operator can be used to construct the difference constraints. We shall first briefly review this method. Then we describe the fast solution method of GRBF interpolant.

To simplify the problem, we only consider the domain constraints and the difference constraints of the gradient. Given a set of scattered data points $\boldsymbol{P} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ with $N = \mu + \sigma$ interpolation constraints (the domain constraints and the difference constraints of the gradient), the GRBF interpolant tries to approximate an unknown function f by the interpolant

$$s(\boldsymbol{x}) = \sum_{j=1}^{\mu} a_j \Phi(\boldsymbol{x}, \boldsymbol{x}_j) + \sum_{k=1}^{\sigma} b_k \Delta'_{\boldsymbol{n}} \Phi\left(\boldsymbol{x}, \boldsymbol{x}_{\mu+k}\right) + p(\boldsymbol{x}) \quad (1)$$

where a_j and b_k are weight coefficients to be determined. $\Phi\left(\boldsymbol{x}, \boldsymbol{x}'\right)$ can be viewed as a common radial basis function $\varphi(\boldsymbol{x}) : R^3 \mapsto R$.

The superscript and subscript are used to distinguish the different effects of the difference operator [5]. $\Delta_{\boldsymbol{n}}$ acts on the first variable $\boldsymbol{x} = (x, y, z)$ of $\Phi\left(\boldsymbol{x}, \boldsymbol{x}'\right)$ and $\Delta'_{\boldsymbol{n}}$ acts on the second variable $\boldsymbol{x}' = (x', y', z')$ of $\Phi\left(\boldsymbol{x}, \boldsymbol{x}'\right)$. The subscript $\boldsymbol{n}$ represents the normal direction at the corresponding variable.

The low degree polynomials $p(\boldsymbol{x})$ consist of monomials

$$p(\boldsymbol{x}) = \sum_{s=1}^{Q} g_s p_s(\boldsymbol{x}) \quad (2)$$

where $g_s, 1 \le s \le Q$ are weight coefficients to be determined and $p_s(\boldsymbol{x}), 1 \le s \le Q$ are monomials. The number of terms Q is determined by the degree of polynomials. All the unknown coefficients can be computed by the interpolation conditions and orthogonality conditions.

For the domain constraints $\{\boldsymbol{x}_i, f(\boldsymbol{x}_i)\}_{i=1}^{i=\mu}$, the GRBF interpolant satisfies

$$\sum_{j=1}^{\mu} a_j \overbrace{\Phi(\boldsymbol{x}_i, \boldsymbol{x}_j)}^{\boldsymbol{A}} + \sum_{k=1}^{\sigma} b_k \overbrace{\Delta'_{\boldsymbol{n}} \Phi(\boldsymbol{x}_i, \boldsymbol{x}_{\mu+k})}^{\boldsymbol{B}} + \overbrace{p(\boldsymbol{x}_i)}^{\boldsymbol{P}} = f_i, \quad 1 \le i \le \mu \quad (3)$$

where $f_i, 1 \le i \le \mu$ are function values of the unknown domain at $\boldsymbol{x}_i$.

For the difference constraints of the gradient $\{\boldsymbol{x}_i, \Delta_{\boldsymbol{n}} f(\boldsymbol{x}_i)\}_{i=\mu+1}^{i=\mu+\sigma}$, the GRBF interpolant satisfies

$$\sum_{j=1}^{\mu} a_j \overbrace{\Delta_{\boldsymbol{n}} \Phi(\boldsymbol{x}_i, \boldsymbol{x}_j)}^{\boldsymbol{B}^T} + \sum_{k=1}^{\sigma} b_k \overbrace{\Delta_{\boldsymbol{n}} \Delta'_{\boldsymbol{n}} \Phi(\boldsymbol{x}_i, \boldsymbol{x}_{\mu+k})}^{\boldsymbol{D}} + \overbrace{\Delta_{\boldsymbol{n}} p(\boldsymbol{x}_i)}^{\boldsymbol{F}} = \delta_i, \quad \mu + 1 \le i \le \mu + \sigma \quad (4)$$

where $\delta_i \mu + 1 \le i \le \mu + \sigma$ are different values of the unknown domain at $\boldsymbol{x}_i$.

For the orthogonality constraints, the GRBF interpolant satisfies

$$\sum_{j=1}^{\mu} a_j \overbrace{p_s(\boldsymbol{x}_j)}^{\boldsymbol{P}^T} + \sum_{k=1}^{\sigma} b_k \overbrace{\Delta_n p_s(\boldsymbol{x}_{\mu+k})}^{\boldsymbol{F}^T} = 0, \quad 1 \le s \le Q \quad (5)$$

The above constraints lead to a linear system

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} & \boldsymbol{P} \\ \boldsymbol{B}^T & \boldsymbol{D} & \boldsymbol{F} \\ \boldsymbol{P}^T & \boldsymbol{F}^T & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \\ \boldsymbol{g} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{\delta} \\ \boldsymbol{0} \end{bmatrix} := \widetilde{\boldsymbol{A}} \widetilde{\boldsymbol{x}} = \widetilde{\boldsymbol{f}}$$

where $\boldsymbol{a} = \left[a_j\right]_{j=1}^{\mu}, \boldsymbol{b} = \left[b_k\right]_{k=1}^{\sigma}, \boldsymbol{g} = \left[g_s\right]_{s=1}^{Q}, \boldsymbol{f} = \left[f_i\right]_{i=1}^{\mu}$ and $\boldsymbol{\delta} = \left[\delta_i\right]_{i=\mu+1}^{\mu+\sigma}$.

The direct solution methods (e.g., LU decomposition) cost $O\left(N^3\right)$ operations and $O\left(N^2\right)$ memory usage. When the number of domain constraints and difference constraints becomes large (more than a few thousand constraints), both the evaluation and solution of the interpolant are unbearable in the case of globally supported basis functions.

IV. FAST EVALUATION

A. FAR FIELD EXPANSION

To solve the linear system efficiently, we first consider a fast evaluation method for GRBF. Taking the 1-D kernel function as an example, the evaluation of the matrix-vector product involves a sum of N kernel function

$$f(x) = \sum_{j=1}^{N} \omega_j K\left(x, y_j\right) \quad (6)$$

where $\omega_j, 1 \le j \le N$ are coefficients of the source/center points and $K(x, y)$ is a kind of kernel function. For convenience, we distinguish between the two variables of the kernel function $K(x, y)$. The first variable x in $K(x, y)$ is viewed as an evaluation point (or target point) and the second variable y is viewed as a source point. Every evaluation of such sum costs $O(N)$ operations and the evaluation of $f(x)$ at N evaluation points obviously costs $O\left(N^2\right)$ operations.

To compute the matrix-vector product efficiently, fast summation methods are required to implement. Among them, the fast multipole method in $O(logN)$ or even constant time is a promising method. It is based on a far field expansion of

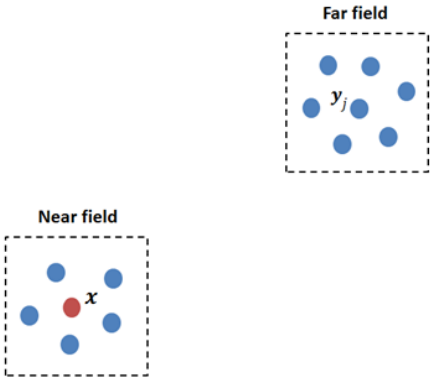

FIGURE 1. A simple case of the far field and near field when evaluating at x. The red point is viewed as an evaluation point and the blue points are viewed as source points.

the kernel function. We can expand the kernel $K(x, y)$ using a low-rank approximation in the form

$$K(x, y) = \sum_{n=1}^{p} \phi_n(x)\psi_n(y) + R_p(x, y) \tag{7}$$

where p is the number of items in the expanded series ϕ_n and ψ_n. $R_p(x, y)$ is a residual that satisfies $R_p(x, y) \longrightarrow 0$ for $\|x - y\|_2 \longrightarrow \infty$ and/or $p \longrightarrow \infty$. This expansion of a finite series serves as a separation of variables. Such kernels are also known as degenerate kernels or finite rank [16].

First, we can pre-compute the moments

$$\Psi_n(y) = \sum_{j=0}^{N} \omega_j \psi_n(y_j) \tag{8}$$

Second, we can approximate $f(x)$ at each evaluation point efficiently using the pre-computed moments

$$f(x) \approx \sum_{n=1}^{p} \phi_n(x) \sum_{j=1}^{N} \omega_j \psi_n(y_j) = \sum_{n=1}^{p} \Psi_n(y)\phi_n(x)$$

Then the evaluation of $f(x)$ at N evaluation points costs $O(pN)$ operations if they are well separated from the source points. Since p is a small constant, the complexity of the evaluation points is linear in the best case. In practical applications, the source points should be decomposed into the well separated points (far field) and not well separated points (near field) using the adaptive multilevel FMM method [31], as shown in Figure 1. The source points in near field will be evaluated using direct summation.

B. KERNEL INDEPENDENT FMM

To expand an arbitrary kernel of the GRBF interpolant conveniently, it is necessary to use a kernel independent or black-box fast multipole method. The black-box FMM proposed by Fong and Darve [9] is a promising method which can be used to expand the low-rank approximation of a kernel function without analytical expansions.

The basic idea of black-box FMM is to interpolate the kernel function using interpolation bases (e.g., Chebyshev polynomials). Taking the 1-D black-box FMM as an example, the first variable of the kernel function can be interpolated by fixing the second variable

$$K(x, y) \approx \sum_{n=1}^{p} K(\bar{x}_n, y)\, S_p(\bar{x}_n, x) \tag{9}$$

where $\{\bar{x}_n\}_{n=1}^{p}$ are the p interpolation nodes corresponding to the first variable and $S_p(\bar{x}_n, x)$ is the interpolating function at the node $\bar{x}_n$. The second variable of $K(\bar{x}_n, y)$ can be interpolated in the same way

$$K(x, y) \approx \sum_{n=1}^{p}\sum_{m=1}^{p} K(\bar{x}_n, \bar{y}_m)\, S_p(\bar{x}_n, x) S_p(\bar{y}_m, y)$$

where $\{\bar{y}_m\}_{m=1}^{p}$ are the p interpolation nodes corresponding to the second variable and $S_p(\bar{x}_n, x)$ is the interpolating function at the node $\bar{y}_m$. Then we can approximate $f(x)$ efficiently by changing the order of summation

$$\begin{aligned} f(x) &\approx \sum_{j=1}^{N} \omega_j \sum_{n=1}^{p}\sum_{m=1}^{p} K(\bar{x}_n, \bar{y}_m)\, S_p(\bar{x}_n, x)\, S_p(\bar{y}_m, y_j) \\ &= \sum_{n=1}^{p} S_p(\bar{x}_n, x) \left[\sum_{m=1}^{p} K(\bar{x}_n, \bar{y}_m) \left[\sum_{j=1}^{N} \omega_j S_p(\bar{y}_m, y_j)\right]\right] \\ &= \sum_{n=1}^{p} \Psi_n(y)\phi_n(x) \end{aligned}$$

where

$$\begin{cases} \Psi_n(y) = \sum_{m=1}^{p} k(\bar{x}_n, \bar{x}_m) \left[\sum_{j=1}^{N} \omega_j S_p(\bar{y}_m, y_i)\right] \\ \phi_n(x) = S_p(\bar{x}_n, x) \end{cases}$$

The interpolation-based scheme can achieve the same effect of far field expansion. As the expansions are defined by the kernel values, this method is very useful to expand kernels with complex analytical expansions.

To utilize the fast summation method, we should first expand the constraints. The evaluation of the domain constraints can be simply decomposed into three sums with different evaluation points and source points

$$\begin{aligned} &\boldsymbol{aA} + \boldsymbol{bB} + \boldsymbol{cP} \\ &= \sum_{j=1}^{\mu} a_j \Phi(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sum_{k=1}^{\sigma} b_k \Delta'_{\boldsymbol{n}} \Phi(\boldsymbol{x}_i, \boldsymbol{x}_{\mu+k}) + p(\boldsymbol{x}_i) \\ &= \sum_{j=1}^{\mu} a_j \Phi(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sum_{k=1}^{\sigma} b_k \Phi\left(\boldsymbol{x}_i, \boldsymbol{x}^{+}_{\mu+k}\right) \\ &\quad - \sum_{k=1}^{\sigma} b_k \Phi\left(\boldsymbol{x}_i, \boldsymbol{x}^{-}_{\mu+k}\right) + p(\boldsymbol{x}_i), \quad 1 \le i \le \mu \end{aligned}$$

where $\boldsymbol{x}^{+}_{\mu+k}$ and $\boldsymbol{x}^{-}_{\mu+k}$ are the two offset points by projecting along the gradient direction at $\boldsymbol{x}_{\mu+k}$ via the definition of the difference operator [5]. The polynomial part can be calculated in linear time.

Similarly, the evaluation of the difference constraints of the gradient can be decomposed into six sums with different

TABLE 1. The expansions of the constraints lead to nine summations with different evaluation points and source points.

kernels	source points	evaluation points	coefficients	numerical ranges
$\Phi(\boldsymbol{x}_i, \boldsymbol{x}_j)$	$\boldsymbol{x}_j$	$\boldsymbol{x}_i$	$+a_j$	$1 \le i \le \mu, 1 \le j \le \mu$
$\Phi(\boldsymbol{x}_i, \boldsymbol{x}^+_{\mu+k})$	$\boldsymbol{x}^+_{\mu+k}$	$\boldsymbol{x}_i$	$+a_j$	$1 \le i \le \mu, 1 \le k \le \sigma$
$\Phi(\boldsymbol{x}_i, \boldsymbol{x}^-_{\mu+k})$	$\boldsymbol{x}^-_{\mu+k}$	$\boldsymbol{x}_i$	$-a_j$	$1 \le i \le \mu, 1 \le k \le \sigma$
$\Phi(\boldsymbol{x}^+_i, \boldsymbol{x}_j)$	$\boldsymbol{x}_j$	$\boldsymbol{x}^+_i$	$+a_j$	$1 \le i \le \mu, 1 \le j \le \mu$
$\Phi(\boldsymbol{x}^-_i, \boldsymbol{x}_j)$	$\boldsymbol{x}_j$	$\boldsymbol{x}^-_i$	$-a_j$	$1 \le i \le \mu, 1 \le j \le \mu$
$\Phi(\boldsymbol{x}^+_i, \boldsymbol{x}^+_{\mu+k})$	$\boldsymbol{x}^+_{\mu+k}$	$\boldsymbol{x}^+_i$	$+b_k$	$1 \le i \le \mu, 1 \le k \le \sigma$
$\Phi(\boldsymbol{x}^-_i, \boldsymbol{x}^+_{\mu+k})$	$\boldsymbol{x}^+_{\mu+k}$	$\boldsymbol{x}^-_i$	$-b_k$	$1 \le i \le \mu, 1 \le k \le \sigma$
$\Phi(\boldsymbol{x}^+_i, \boldsymbol{x}^-_{\mu+k})$	$\boldsymbol{x}^-_{\mu+k}$	$\boldsymbol{x}^+_i$	$-b_k$	$1 \le i \le \mu, 1 \le k \le \sigma$
$\Phi(\boldsymbol{x}^-_i, \boldsymbol{x}^-_{\mu+k})$	$\boldsymbol{x}^-_{\mu+k}$	$\boldsymbol{x}^-_i$	$+b_k$	$1 \le i \le \mu, 1 \le k \le \sigma$

evaluation points and source points $\boldsymbol{B}^T + \boldsymbol{bD} + \boldsymbol{cF}$, as shown at the bottom of this page, where $\boldsymbol{x}^+_i$ and $\boldsymbol{x}^-_i$ are the two offset points by projecting along the gradient direction at $\boldsymbol{x}_i$ via the definition of the difference operator [5].

Table 1 shows the nine summations corresponding to different kernels. Compared to the differential operator, the difference operator changes the evaluation points and source points instead of the kernel type. This ensures that the RBF kernel retains its original properties (e.g., strictly positive definite or conditionally positive definite). And we can expand the nine kernels using a low-rank approximation respectively.

It is worth noting that the RBF kernel with the same source points has the same pre-computed moments $\Psi_n(\boldsymbol{y})$. According to the analysis of the expansions in Table 1, the source points can be classified into three categories $\boldsymbol{x}_j$, $\boldsymbol{x}^+_{\mu+k}$ and $\boldsymbol{x}^-_{\mu+k}$. Therefore, we can represent the nine summations as three kinds of RBF kernels with different evaluation points, which can simplify the amount of calculation.

V. FAST SOLUTION

A. PRECONDITIONING

We have solved the fast evaluation problem of matrix-vector product for GRBF, and we can now solve the linear system efficiently using the Krylov subspace iterative methods. The common iterative methods used for RBF are the conjugate gradient method and the generalized minimum residual method. There are several possible preconditioning technologies that can be used to improve the rate of convergence of the linear system. Since the evaluation process still takes a lot of time per iteration for large problems, the number of iterations must be reduced as much as possible and the preconditioning technology is required to be implemented. The preconditioning strategy is crucial for satisfactory convergence.

Different from the existing methods, we use the Flexible GMRES iterative method with variable preconditioners to solve the linear system. The FGMRES method also stores an orthonormal basis of the Krylov subspace to form the conjugate vectors at each iteration step. One difference between GMRES and FGMRES is that the latter is represented by a linear combination of the different preconditioned residual vectors. For a linear system $\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{x}} = \widetilde{\boldsymbol{f}}$, taking the right preconditioned GMRES as an example, the Arnoldi iteration is used to recursively construct an orthonormal basis of the right preconditioned Krylov space

$$span\left\{\boldsymbol{r}_0, \widetilde{\boldsymbol{A}}\boldsymbol{M}^{-1}\boldsymbol{r}_0, \ldots, \left(\widetilde{\boldsymbol{A}}\boldsymbol{M}^{-1}\right)^m \boldsymbol{r}_0\right\}$$

where $\boldsymbol{r}_0 = \widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{x}}_0 - \widetilde{\boldsymbol{f}}$ is the initial residual vector, $\widetilde{\boldsymbol{x}}_0$ is the first trial vector (usually zero), m is the number of iteration and $\boldsymbol{M}$ is the right preconditioner. The FGMRES method allows changes for the preconditioner to enhance robustness at each iteration step, and its right preconditioned Krylov space becomes

$$span\left\{\boldsymbol{r}_0, \widetilde{\boldsymbol{A}}\boldsymbol{M}_1^{-1}\boldsymbol{r}_0, \ldots, \left(\widetilde{\boldsymbol{A}}\boldsymbol{M}_m^{-1}\right)^m \boldsymbol{r}_0\right\}$$

where $\boldsymbol{M}_k, k = 1, \ldots m$ are the different preconditioners.

One of the benefits is that any iterative method can be used as a preconditioner, which is known as an inner-outer

$$\begin{aligned}
\boldsymbol{B}^T + \boldsymbol{bD} + \boldsymbol{cF} &= \sum_{j=1}^{\mu} a_j \Delta_{\boldsymbol{n}} \Phi\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) + \sum_{k=1}^{\sigma} b_k \Delta_{\boldsymbol{n}} \Delta'_{\boldsymbol{n}} \Phi\left(\boldsymbol{x}_i, \boldsymbol{x}_{\mu+k}\right) + \Delta_{\boldsymbol{n}} p\left(\boldsymbol{x}_i\right) \\
&= \sum_{j=1}^{\mu} a_j \Phi\left(\boldsymbol{x}^+_i, \boldsymbol{x}_j\right) - \sum_{j=1}^{\mu} a_j \Phi\left(\boldsymbol{x}^-_i, \boldsymbol{x}_j\right) + \sum_{k=1}^{\sigma} b_k \Phi\left(\boldsymbol{x}^+_i, \boldsymbol{x}^+_{\mu+k}\right) - \sum_{k=1}^{\sigma} b_k \Phi\left(\boldsymbol{x}^-_i, \boldsymbol{x}^+_{\mu+k}\right) - \sum_{k=1}^{\sigma} b_k \Phi\left(\boldsymbol{x}^+_i, \boldsymbol{x}^-_{\mu+k}\right) \\
&\quad + \sum_{k=1}^{\sigma} b_k \Phi\left(\boldsymbol{x}^-_i, \boldsymbol{x}^-_{\mu+k}\right) + p\left(\boldsymbol{x}^+_i\right) - p\left(\boldsymbol{x}^-_i\right), \quad \mu + 1 \le i \le \mu + \sigma
\end{aligned}$$

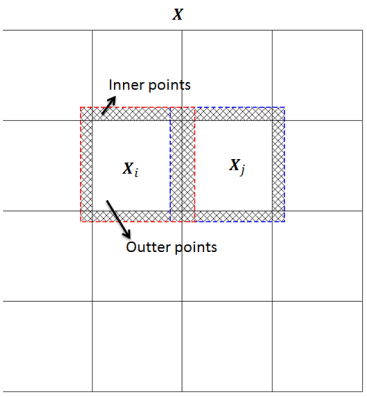

FIGURE 2. A schematic diagram of the overlapping subdomains. For each subdomain X_i, there is an inner point that is not contained in any of the other subdomains.

preconditioned GMRES method. Then we can take the result of each iteration step as a different preconditioner in the inner iteration. And we can expect a more accurate preconditioner along with the process of outer iteration. It is a useful method to reduce the number of iterations for the Krylov subspace method.

B. DOMAIN DECOMPOSITION METHOD

Now we can use an iterative algorithm that is more suitable for radial basis functions as a preconditioner. As a variant of the two-level domain decomposition method for RBF proposed by Beatson *et al.* [27], a multilevel domain decomposition method is analyzed to improve overlap using a nested sequence of levels.

The basic idea of the domain decomposition method is to divide the whole domain $\boldsymbol{X}$ into overlapping subdomains $\boldsymbol{X}_i, i = 1, \ldots, D$ such that $\boldsymbol{X} = \cup_{i=1}^{D} \boldsymbol{X}_i$ and $\boldsymbol{X}_i \cap \boldsymbol{X}_j \neq \emptyset$, until the subdomains become simple enough to be solved directly. Note that the domain refers to a set of interpolation centers. The number of interpolation centers in $\boldsymbol{X}_i$ is denoted as $N_i = |\boldsymbol{X}_i|$. The interpolation centers in each subdomain are classified as inner points (non-overlapping part) and outer points (overlapping part), as shown in Figure 2. The solutions to the subdomains are then combined to give a solution to the whole domain using the alternating projection method [3]. The alternating projection method interpolates a subdomain to correct the associated local residual and then interpolates the next subdomain with the corrected residual in sequence. The global residual will converge within a specified accuracy ϵ in finite cycles only if the subdomains are weakly distinct. Details about the convergence theory of the alternating projection method can be found in [3], [27].

The number of iteration is strongly affected by the size of the inner points and outer points in each subdomain. To further improve the convergence, a two-level method [27] can be employed to reduce the spectral radius of the linear system. The divided subdomains $\boldsymbol{X}_i$ are viewed as a fine level. In addition to the correction of the fine level, this method adds a correction of coarse level $\boldsymbol{Y}$ by randomly choosing some inner points from each subdomain $\boldsymbol{X}_i$ in a certain ratio ρ_1.

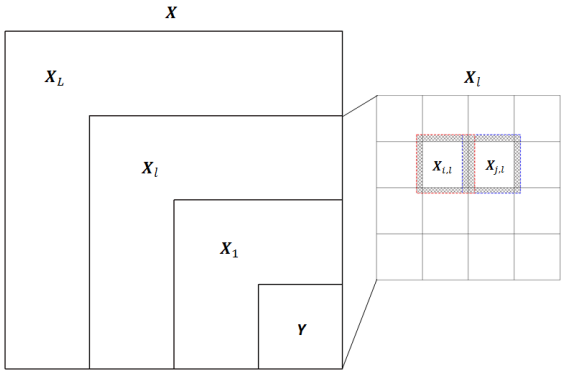

FIGURE 3. A schematic diagram of the multilevel domain decomposition scheme. The whole domain X is decomposed into a nested sequence of levels $X_l, l = 1, \ldots, L$.

Both the results of fine level and coarse level are combined to give a solution to the whole domain using the alternating projection method. The coarse level improves the overlap between subdomains globally. Numerical experiments show that the added coarse level correction can significantly improve the convergence rate which becomes better as the ratio of the coarse level points, or the amount of overlap, is increased.

Since the coarse level is selected in a certain ratio over the whole domain $\boldsymbol{X}$, the interpolation centers in coarse level would be too large to solve directly. If the coarse level is viewed as a new whole domain, it is natural to apply the two-level method to the multiple levels recursively. This allows us to turn a larger domain into subdomains that are easy to solve. The multilevel domain decomposition method decomposes the whole domain $\boldsymbol{X}$ into a nested sequence of levels $\boldsymbol{X}_l, l = 1, \ldots, L$ such that $\boldsymbol{X}_l \subset \boldsymbol{X}_{l+1}, l = 1, \ldots, L-1$ and $\boldsymbol{X}_L = \boldsymbol{X}$. Each level $\boldsymbol{X}_l$ is divided into overlapping subdomains $\boldsymbol{X}_{i,l}, i = 1, \ldots, D_l$ such that $\boldsymbol{X}_l = \cup_{i=1}^{D_l} \boldsymbol{X}_{i,l}$ and $\boldsymbol{X}_{i,l} \cap \boldsymbol{X}_{j,l} \neq \emptyset$. $\boldsymbol{X}_l$ is constructed by randomly choosing some inner points from each subdomain $\boldsymbol{X}_{i,l+1}$ in a certain ratio ρ_1 recursively. The coarse level $\boldsymbol{Y}$ is constructed by randomly choosing some inner points from each subdomain $\boldsymbol{X}_{i,1}$ in a certain ratio ρ_1. The number of interpolation centers in $\boldsymbol{X}_{i,l}$ is denoted as $N_{i,l} = |\boldsymbol{X}_{i,l}|$ and in $\boldsymbol{X}_l$ is denoted as $N_l = |\boldsymbol{X}_l|$. Then the results of multiple levels are combined to give a solution to the whole domain using the alternating projection method.

The simplified pseudo code of the extension to the multilevel domain decomposition method is given below. s_g, $\widetilde{\boldsymbol{x}}_g$ and $\widetilde{\boldsymbol{f}}_g$ will denote the current approximate GRBF interpolant, the current solution coefficients and the current residual. s_l, $\widetilde{\boldsymbol{x}}_l$ and $\widetilde{\boldsymbol{f}}_l$ denote the same meaning at each level.

Input: The whole domain $\boldsymbol{X}$ and the divided subdomains $\boldsymbol{X}_l, l = 1, \ldots, L$, accuracy $\epsilon > 0$, right-hand side of the linear system $\widetilde{\boldsymbol{f}}$.

C. SOLUTION PROCEDURE

According to the above analyses, the fast solution method of GRBF includes three key parts: a fast evaluation method for computing the matrix-vector product and the residuals,

$s_g \leftarrow 0, \widetilde{\boldsymbol{x}}_g \leftarrow 0, \widetilde{\boldsymbol{f}}_g \leftarrow \widetilde{\boldsymbol{f}}$
while $\|\widetilde{\boldsymbol{f}}_g\| \geq \epsilon$
  for $l = L, \ldots, 1$ ***do***
    $s_l \leftarrow 0, \widetilde{\boldsymbol{x}}_l \leftarrow 0, \widetilde{\boldsymbol{f}}_l \leftarrow \widetilde{\boldsymbol{f}}_g$
    for $i = 1, \ldots, D_l$ ***do***
      Solve the subdomain $\boldsymbol{X}_{i,l}$ *corresponding to the residual* $\widetilde{\boldsymbol{f}}_l$ *and set the solution coefficients of inner points to* $\widetilde{\boldsymbol{x}}_l$. *The divided subdomains can be solved using a direct solution method in parallel.*
    end for
      To satisfy the orthogonal condition, correct the coefficients $\widetilde{\boldsymbol{x}}_l$ *such that it is orthogonal to the polynomial space.*
      Form the fine level interpolant $s_{1,l}$ *using the corrected coefficients* $\widetilde{\boldsymbol{x}}_l$.
      Solve the coarse level $\boldsymbol{Y}$ *corresponding to the residual* $\widetilde{\boldsymbol{f}}_l - s_{1,l}$ *and form the coarse level interpolant* $s_{2,l}$ *using the solution results.*
      $s_l \leftarrow s_{1,l} + s_{2,l}$
      $s_g \leftarrow s_g + s_l$
      $\widetilde{\boldsymbol{f}}_g \leftarrow \widetilde{\boldsymbol{f}}_g - s_l$
  end for
end while
Output the final solution coefficients $\widetilde{\boldsymbol{x}}_g$.

for $l = L, \ldots, 1$ ***do***
  Divide the space of the subdomain $\boldsymbol{X}_l$ *into rectangular boxes using a balanced kd-tree data structure. Form overlapping subdomains* $\boldsymbol{X}_{i,l}$ *via the rectangular boxes.*
end for
Form the coarse level $\boldsymbol{Y}$ *via the subdomains* $\boldsymbol{X}_{i,1}$.
Expand the kernels for each level using the fast evaluation method.
Compute the initial residual vector $\boldsymbol{r}_0 = \widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{x}}_0 - \widetilde{\boldsymbol{f}}, \boldsymbol{v}_1 = \boldsymbol{r}_0 / \|\boldsymbol{r}_0\|_2$.
for $k = 1, 2, \ldots$ ***do***
  Compute the preconditioned vector $\boldsymbol{z}_j = \boldsymbol{M}_k^{-1}\boldsymbol{v}_k$. *Note that the preconditioner* $\boldsymbol{M}_k$ *isn't stored explicitly.* $\boldsymbol{z}_j$ *is the solution result of one cycle of the multilevel domain decomposition method.*
  Compute $\boldsymbol{v}_{k+1}$ *via Arnoldi process.*
  Define $\boldsymbol{Z}_k = [z_1, \ldots, z_k]$.
  Compute $\widetilde{\boldsymbol{x}}_k = \widetilde{\boldsymbol{x}}_0 + \boldsymbol{Z}_k\boldsymbol{y}_k$ *where* $\boldsymbol{y}_k$ *can be computed by minimizing the residual.*
  if $\|\boldsymbol{r}_k\|_2 / \|\boldsymbol{r}_0\|_2 < \epsilon$ ***then***
    stop.
  end if
  end for
Output the final solution coefficients $\widetilde{\boldsymbol{x}}_k$.

an inner iteration method for preconditioning and an outer iteration method. In this paper, we use the black-box FMM as the fast evaluation method, the multilevel domain decomposition method as the inner iteration method and the Flexible GMRES iterative method as the outer iteration method. As the number of interpolation centers at each level is large, the fast evaluation method is also used to evaluate the residuals in the process of inner iteration.

For a given linear system of GRBF and the desired accuracy ϵ, a simplified procedure of the fast solution method is given below. $\boldsymbol{r}_k$ and $\widetilde{\boldsymbol{x}}_k$ are a sequence of the residual vectors and solution vectors for FGMRES.

VI. NUMERICAL RESULTS

We have implemented the algorithm of the fast solution method using some open source libraries, especially the ScalFMM library [32]. The black-box FMM in ScalFMM was used to implement the fast evaluation process. To speed up the process of iteration, the solution of subdomains was implemented with Intel Math Kernel Library (Intel MKL) in parallel.

We tested the solution method on several data sets composed of Hermite points. The data sets were randomly sampled from several real objects, as shown in Figure 4. Based on the method of implicit surface reconstruction [5], the sampling points can be converted into the domain constraints and the sampling normals can be converted into the difference constraints of the gradient. To recover the implicit surface, we can efficiently interpolate these constraints (including the domain constrains and the difference constraints of the gradient) for the solution process of implicit surface reconstruction.

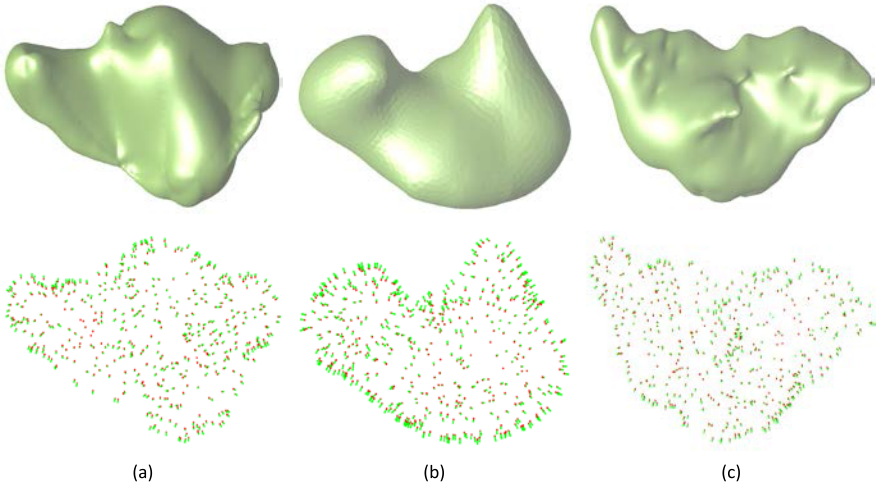

FIGURE 4. The red points were randomly sampled from several real objects and the green lines were the estimated normals.

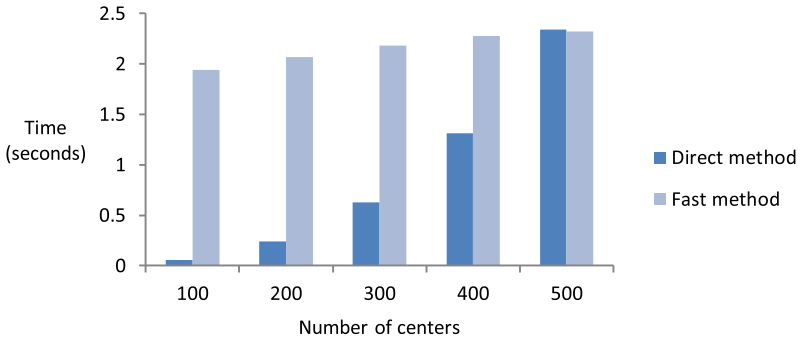

FIGURE 5. Comparison of the performance using the direct method with the fast method. Both the number of domain constraints and difference constraints varied from 100 to 500 ($\mu = \sigma$). The desired accuracy for the fast method is 10^{-4}.

The solution method utilizes several parameters, the number of levels L, the overlapping ratio ρ_1, the number of interpolation centers $N_{\boldsymbol{Y}}$ in coarse level, the max number of interpolation centers $N_{i,l}$ in $\boldsymbol{X}_{i,l}$ and the ratio of inner points in one subdomain ρ_2. Most of them are used to divide the multilevel subdomains. To ensure each subdomain is solved efficiently using a direct method, the value of $N_{i,l}$ should be small enough. However, a small $N_{i,l}$ leads to a large number of subdomains. Under the premise of equalization efficiency and convergence, a good choice is to divide the subdomains automatically. We use the following set of parameter values for all the numerical examples: $\rho_1 = 0.1 \sim 0.2$, $\rho_2 = 0.2 \sim 0.3$, $N_{\boldsymbol{Y}} = 1024 \sim 2048$ and $N_{i,l} = 128 \sim 256$. The number of levels L can be approximated by the value of ρ_1, ρ_2 and $N_{\boldsymbol{Y}}$. The actual values are adjusted according to the distribution of the data, which is related to the uniformity of the data distribution. In the following examples, we used the triharmonic spline as the basic function. For the far field expansion, the number of items in the expanded series was set to 10.

To validate the viability and performance of this method, we compared the results with the direct solution method (LU decomposition method) and tested on a Windows 64-bit PC with 3.00 GHz Intel(R) Core(TM) i5-7400 and 8GB RAM. The performance of the solution method mainly depends on the number of constraints and the desired accuracy. Figure 5 and Figure 6 reported the timings of the solution process of the direct method and the fast method. As expected, the improved efficiency is significant both in the evaluation and solution processes. For the problems with larger than 10,000 interpolation centers, the direct solution method will be very time consuming. Though the direct solution method cannot be used to solve large-scale problems, it has better performance for smaller data sets, especially the problems with less than 1000 interpolation centers. This is also the basis for the choice of subdomain size $N_{i,l}$.

The convergence rate of the fast solution method mainly depends on the division of the whole domain $\boldsymbol{X}$. The method converges with a small number of iterations as long as a sufficient overlap is created, as shown in Figure 7 and Figure 8. More importantly, the solution failure of a subdomain will result in the interruption of the whole solution process. Therefore, the method requires that each subdomain is solvable. The failure example occurs in trivial solutions when the

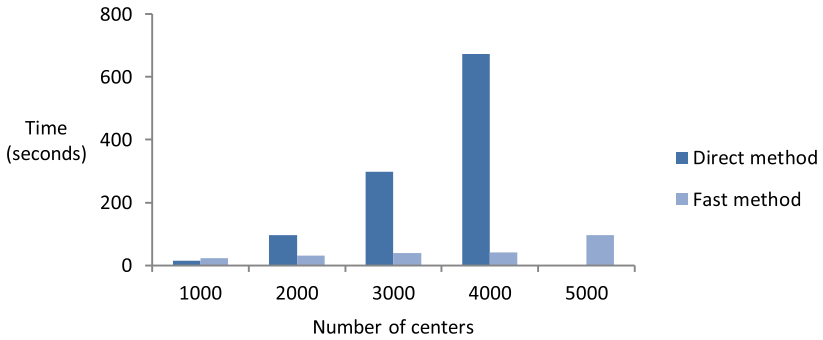

FIGURE 6. Comparison of the performance using the direct method with the fast method. Both the number of domain constraints and difference constraints varied from 1000 to 5000 ($\mu = \sigma$). The direct method was out of memory when $\mu = \sigma = 5000$. The desired accuracy for the fast method is 10^{-4}.

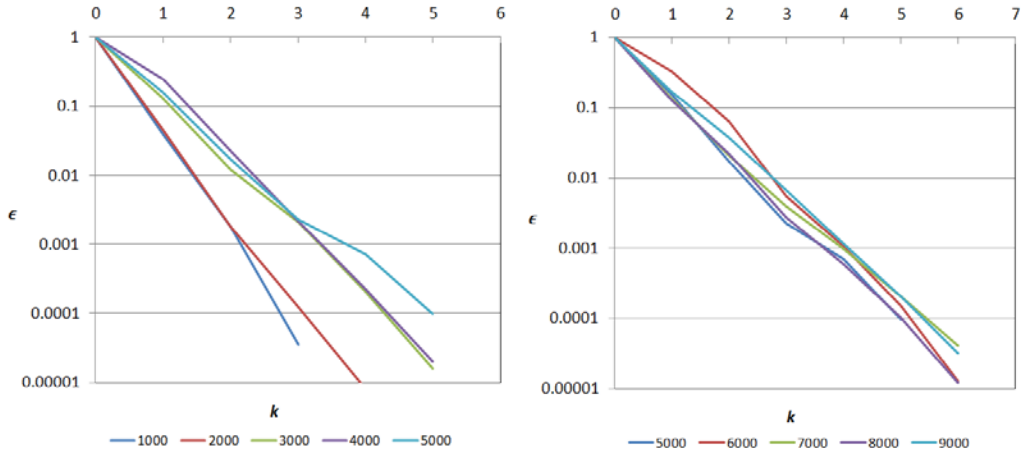

FIGURE 7. Convergence rate of the solution method using the two-level domain decomposition method. Both the number of domain constraints and difference constraints varied from 1000 to 9000 ($\mu = \sigma$).

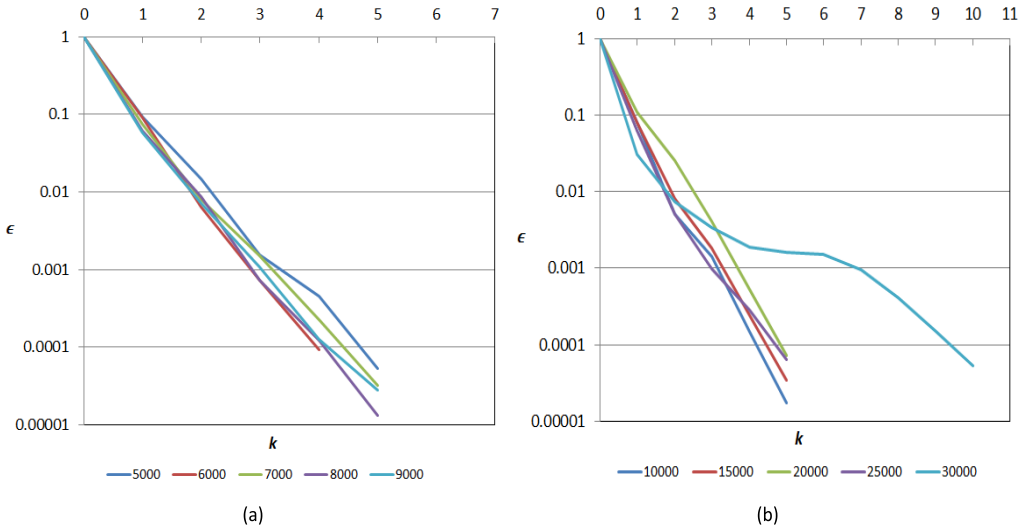

FIGURE 8. Convergence rate of the solution method using the multi-level domain decomposition method. Both the number of domain constraints and difference constraints varied from 5000 to 30000 ($\mu = \sigma$).

right-hand side of the linear system is zero. For example, a subdomain only contains the domain constraints with zero function values. To avoid trivial solutions, we should ensure that the domain constraints and the difference constraints are uniformly distributed in subdomains at each level.

Table 2 showed that the correction of coarse level has a great impact on the convergence for the two-level domain decomposition method. The number of iterations increases remarkably as the ratio of the coarse level points is reduced. However, a large number of the interpolation centers in coarse

TABLE 2. Number of iterations and performance of the FGMRES iteration method using the two-level domain decomposition method. The number of interpolation centers in coarse level was 1024, 2048 and 4096 respectively.

N	μ	σ	ϵ	Number of iterations			Time (seconds)		
				1024	2048	4096	1024	2048	4096
20,000	10,000	10,000	10^{-4}	8	6	5	159.3	134.7	131.6
30,000	15,000	15,000	10^{-4}	12	7	6	237.3	159.6	161.8
40,000	20,000	20,000	10^{-4}	12	9	6	247.1	208.7	168.0
50,000	25,000	25,000	10^{-4}	23	11	9	455.1	258.9	240.2
60,000	30,000	30,000	10^{-4}	34	16	12	647.3	359.3	335.7

TABLE 3. Number of iterations and performance of the FGMRES iteration method using the multi-level domain decomposition method. The number of the levels was 2.

N	μ	σ	ϵ	L	$N_{l=1}$	N_Y	Number of iterations	Time (seconds)
10,000	5,000	5,000	10^{-4}	2	1308	1022	5	119.4
12,000	6,000	6,000	10^{-4}	2	1586	1026	4	114.4
14,000	7,000	7,000	10^{-4}	2	1834	1018	5	127.4
16,000	8,000	8,000	10^{-4}	2	2076	1004	5	132.3
18,000	9,000	9,000	10^{-4}	2	2366	1018	5	137.8

TABLE 4. Number of iterations and performance of the FGMRES iteration method using the multi-level domain decomposition method. The number of the levels was 2.

N	μ	σ	ϵ	L	$N_{l=1}$	N_Y	Number of iterations	Time (seconds)
20,000	10,000	10,000	10^{-4}	2	2628	1022	5	151.3
30,000	15,000	15,000	10^{-4}	2	3882	1040	5	154.0
40,000	20,000	20,000	10^{-4}	2	5190	1056	5	179.1
50,000	25,000	25,000	10^{-4}	2	6514	1054	7	223.0
60,000	30,000	30,000	10^{-4}	2	7646	2034	10	362.4

level would be too large to solve directly. The multilevel domain decomposition method converges with a coarse level of roughly $1024 \sim 2048$ points, as shown in Table 3 and Table 4. The multilevel method improves the overlap using a nested sequence of levels instead of the ratio of the coarse level points. It avoids the number of interpolation centers in the coarse level to be too large to solve. As mentioned earlier, the solution method is kernel independent both in the evaluation and solution processes. It is very convenient to implement the solution method with different RBF kernels for the selection of most adequate kernel [33]. Table 5 shows the performance of the solution method with different RBF kernels, including thin-plate spline (TPS), biharmonic spline (BIS) and triharmonic spline (TRS).

VII. DISCUSSION

The method still has several limitations that await further investigation and improvement. One of the main limitations is that the subdomains are solved using the natural basis instead of a better basis. For the radial basis function interpolation problem, it is well known that the interpolation equation is frequently ill-conditioned [34], even when the number of interpolation centers is small. For larger data sets, it is worth noting that the ill-conditioning can influence the accuracy of evaluation and reduce the convergence rate of the iterative method. For example, the algorithm will not converge with a small number of iterations. To improve the robustness of the iterative method, the natural basis should be changed to reduce the condition number of the interpolation equation. The approximate cardinal basis functions preconditioning technology are recommended to be extended for the GRBF interpolant. Another limitation of the fast solution method is that it can only be used to interpolate the domain constraints and difference constraints.

An important extension to the fast solution method is to improve the performance of the evaluation process. The same

TABLE 5. Number of iterations and performance of the FGMRES iteration method using different RBF kernels. The number of the levels was 2.

N	μ	σ	ϵ	Number of iterations			Time (seconds)		
				TPS	BIS	TRS	TPS	BIS	TRS
20,000	10,000	10,000	10^{-4}	4	4	5	120.7	126.9	151.3
30,000	15,000	15,000	10^{-4}	4	4	5	133.2	131.1	154.0
40,000	20,000	20,000	10^{-4}	4	4	5	143.9	148.1	179.1
50,000	25,000	25,000	10^{-4}	4	4	7	152.3	154.1	223.0
60,000	30,000	30,000	10^{-4}	6	7	10	209.3	251.1	362.4

number of the difference constrains costs more time than the domain constraints. In the evaluation process, we evaluate the several summations respectively. In fact, there is a certain relationship between these summations. If we can evaluate these summations in a more general way, we can further improve the efficiency of the evaluation. To further improve the solution performance, the optimum division of subdomains can be investigated to reduce the number of iterations.

Although we only consider the fast solution of the GRBF interpolant with domain constraints and difference constraints, the solution of the differential constraints (including the gradient constraints and the tangent constraints) can be studied in the same way. As the derivation of the RBF kernel hinders the fast summation, the main problem that needs to be solved is to evaluate the differential constraints efficiently.

VIII. CONCLUSION

We have presented a fast solution method of the generalized radial basis functions interpolant for global interpolation. The method can be used in the GRBF interpolant with large numbers of domain constraints and difference constraints. One of the main features is that the implemented solution algorithm generally achieves $O(NlogN)$ complexity and $O(N)$ storage. Moreover, it is kernel independent both in the evaluation and solution processes. It is very convenient to apply various types of RBF kernels in different applications without excessive modifications to the existing process. The preconditioning technology is the key to the numerical stability of solution. The optimum division of subdomains and the appropriate overlap rate of subdomains can improve the convergence significantly. Compared to the two-level method, the multilevel domain decomposition method has a better convergence rate for larger data sets by improving the overlap rate of coarse level. Numerical results showed that the fast evaluation method has a good performance for the evaluation of GRBF and the preconditioned Krylov subspace iterative method has a good convergence rate with a small number of iterations.

REFERENCES

[1] W. Zongmin, "Hermite-Birkhoff interpolation of scattered data by radial basis functions," *Approximation Theory Appl.*, vol. 8, no. 2, pp. 1–10, Jun. 1992.
[2] G. E. Fasshauer, *Meshfree Approximation Methods With MATLAB*. London, U.K.: World Scientific, 2007.
[3] H. Wendland, *Scattered Data Approximation*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[4] J. P. Gois, D. F. Trevisan, H. C. Batagelo, and I. Macêdo, "Generalized hermitian radial basis functions implicits from polygonal mesh constraints," *Vis. Comput.*, vol. 29, nos. 6–8, pp. 651–661, Jun. 2013.
[5] D.-Y. Zhong, L.-G. Wang, and L. Bi, "Implicit surface reconstruction based on generalized radial basis functions interpolant with distinct constraints," *Appl. Math. Model.*, vol. 71, pp. 408–420, Jul. 2019.
[6] M. J. Hillier, E. M. Schetselaar, E. A. de Kemp, and G. Perron, "Three-dimensional modelling of geological surfaces using generalized interpolation with radial basis functions," *Math. Geosci.*, vol. 46, no. 8, pp. 931–953, Nov. 2014.
[7] I. Macêdo, J. P. Gois, and L. Velho, "Hermite radial basis functions implicits," *Comput. Graph. Forum*, vol. 30, no. 1, pp. 27–42, Mar. 2011.
[8] D. Zhong and L. Wang, "Solution optimization of RBF interpolation for implicit modeling of orebody," *IEEE Access*, vol. 8, pp. 13781–13791, Jan. 2020.
[9] W. Fong and E. Darve, "The black-box fast multipole method," *J. Comput. Phys.*, vol. 228, no. 23, pp. 8712–8725, Dec. 2009.
[10] Y. Saad, "A flexible inner-outer preconditioned GMRES algorithm," *SIAM J. Sci. Comput.*, vol. 14, no. 2, pp. 461–469, Mar. 1993.
[11] F. Calakli and G. Taubin, "SSD: Smooth signed distance surface reconstruction," *Comput. Graph. Forum*, vol. 30, no. 7, pp. 1993–2002, Sep. 2011.
[12] H. Wendland, "Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree," *Adv. Comput. Math.*, vol. 4, no. 1, pp. 389–396, Dec. 1995.
[13] M. S. Floater and A. Iske, "Multistep scattered data interpolation using compactly supported radial basis functions," *J. Comput. Appl. Math.*, vol. 73, nos. 1–2, pp. 65–78, Oct. 1996.
[14] E. Shivanian, "Local radial basis function interpolation method to simulate 2D fractional-time convection-diffusion-reaction equations with error analysis," *Numer. Methods Partial Differ. Equ.*, vol. 33, no. 3, pp. 974–994, May 2017.
[15] S. Liu, C. C. L. Wang, G. Brunnett, and J. Wang, "A closed-form formulation of HRBF-based surface reconstruction by approximate solution," *Comput.-Aided Des.*, vol. 78, pp. 147–157, Sep. 2016.
[16] R. Beatson and L. Greengard, "A short course on fast multipole methods," *Wavelets, Multilevel Methods Elliptic PDEs*, vol. 1, pp. 1–37, 1997.
[17] V. C. Raykar, "A short primer on the fast multipole method," Tech. Rep., 2005, vol. 1.
[18] R. K. Beatson, M. J. D. Powell, and A. M. Tan, "Fast evaluation of polyharmonic splines in three dimensions," *IMA J. Numer. Anal.*, vol. 27, no. 3, pp. 427–450, 2006.
[19] J. B. Cherrie, R. K. Beatson, and G. N. Newsam, "Fast evaluation of radial basis functions: Methods for generalized multiquadrics in R^n," *SIAM J. Sci. Comput.*, vol. 23, no. 5, pp. 1549–1571, Jan. 2002.
[20] R. K. Beatson, W. E. Ong, and I. Rychkov, "Faster fast evaluation of thin plate splines in two dimensions," *J. Comput. Appl. Math.*, vol. 261, pp. 201–212, May 2014.
[21] M. Spivak, S. K. Veerapaneni, and L. Greengard, "The fast generalized gauss transform," *SIAM J. Sci. Comput.*, vol. 32, no. 5, pp. 3092–3107, Jan. 2010.
[22] L. Ying, "A kernel independent fast multipole algorithm for radial basis functions," *J. Comput. Phys.*, vol. 213, no. 2, pp. 451–457, Apr. 2006.

[23] D. Brown, L. Ling, E. Kansa, and J. Levesley, "On approximate cardinal preconditioning methods for solving PDEs with radial basis functions," *Eng. Anal. Boundary Elements*, vol. 29, no. 4, pp. 343–353, Apr. 2005.
[24] R. K. Beatson, J. B. Cherrie, and C. T. Mouat, "Fast fitting of radial basis functions: Methods based on preconditioned GMRES iteration," *Adv. Comput. Math.*, vol. 11, nos. 2–3, pp. 253–270, Nov. 1999.
[25] N. A. Gumerov and R. Duraiswami, "Fast radial basis function interpolation via preconditioned Krylov iteration," *SIAM J. Sci. Comput.*, vol. 29, no. 5, pp. 1876–1899, Jan. 2007.
[26] A. Faul and M. Powell, "Krylov subspace methods for radial basis function interpolation," *Numer. Anal.*, vol. 2000, pp. 115–141, Jan. 1999.
[27] R. K. Beatson, W. A. Light, and S. Billings, "Fast solution of the radial basis function interpolation equations: Domain decomposition methods," *SIAM J. Sci. Comput.*, vol. 22, no. 5, pp. 1717–1740, Jan. 2001.
[28] R. Yokota, L. A. Barba, and M. G. Knepley, "PetRBF—A parallel O (N) algorithm for radial basis function interpolation with Gaussians," *Comput. Methods Appl. Mech. Eng.*, vol. 199, nos. 25–28, pp. 1793–1804, May 2010.
[29] M. Kamranian, M. Dehghan, and M. Tatari, "An image denoising approach based on a meshfree method and the domain decomposition technique," *Eng. Anal. Boundary Elements*, vol. 39, pp. 101–110, Feb. 2014.
[30] H. Wendland, "Solving large generalized interpolation problems efficiently," in *Advances in Constructive Approximation*, M. Neamtu and E. B. Saff, Eds. Brentwood, TN, USA: Nashboro Press, 2004, pp. 509–518.
[31] N. A. Gumerov, R. Duraiswami, and E. A. Borovikov, "Data structures, optimal choice of parameters, and complexity results for generalized multilevel fast multipole methods in *d* dimensions," Univ. Maryland, College Park, Maryland, USA, Tech. Rep., 2003.
[32] P. Blanchard, B. Bramas, O. Coulaud, E. Darve, L. Dupuy, and A. Etcheverry, "ScalFMM: A generic parallel fast multipole library," in *Proc. SIAM Conf. Comput. Sci. Eng.*, Salt Lake City, UT, USA, Mar. 2015.
[33] H. Rocha, "On the selection of the most adequate radial basis function," *Appl. Math. Model.*, vol. 33, no. 3, pp. 1573–1583, Mar. 2009.
[34] R. K. Beatson, J. Levesley, and C. T. Mouat, "Better bases for radial basis function interpolation problems," *J. Comput. Appl. Math.*, vol. 236, no. 4, pp. 434–446, Sep. 2011.

DEYUN ZHONG received the bachelor's degree from Central South University, in 2013, and the M.S. degree from Fuzhou University, in 2016. He is currently pursuing the Ph.D. degree with Central South University. His current research interests include geological statistics, geology modeling, surface reconstruction, geometry processing, and their applications.

LIGUAN WANG received the M.S. degree from Central South University, China, in 1988, and the Ph.D. degree from Akita University, Japan, in 2002. He is currently a Professor with Central South University. His current research interests mainly focused on digital mine, geology modeling, rock mechanics, and their applications.

LIN BI received the M.S. degree from the China University of Geosciences, in 2006, and the Ph.D. degree from Central South University, in 2010. He is currently an Associate Professor with Central South University. His research areas include digital mine, geology modeling, simultaneous localization and mapping, and their applications.