J Sci Comput (2015) 63:862–884
DOI 10.1007/s10915-014-9919-9

Convexity and Solvability for Compactly Supported Radial Basis Functions with Different Shapes

Shengxin Zhu · Andrew J. Wathen

Received: 11 May 2014 / Revised: 30 August 2014 / Accepted: 4 September 2014 /
Published online: 2 October 2014
© Springer Science+Business Media New York 2014

Abstract It is known that interpolation with radial basis functions of the same shape can guarantee a nonsingular interpolation matrix, whereas little was known when one uses various shapes. In this paper, we prove that functions from a class of compactly supported radial basis functions are convex on a certain region; based on this local convexity and other local geometrical properties of the interpolation points, we construct a sufficient condition which guarantees diagonally dominant interpolation matrices for radial basis functions interpolation with different shapes. The proof is constructive and can be used to design algorithms directly. Numerical examples show that the scheme has a low accuracy but can be implemented efficiently. It can be used for inaccurate models where efficiency is more desirable. Large scale 3D implicit surface reconstruction problems are used to demonstrate the utility and reasonable results can be obtained efficiently.

Keywords Radial basis function · Wendland function · Solvability · RBF · Various shape

Mathematics Subject Classification 65D05

This research is supported by Award no KUK-C1-013-04, made by King Abdullah University of Science of Technology.

S. Zhu (✉)
Oxford Center for Collaborative and Applied Mathematics and Numerical Analysis Group, Mathematical Institute, The University of Oxford, Andrew Wiles Building, Radcliffe Observatory Quarter, Woodstock Road, Oxford OX2 6GG, UK
e-mail: zhus@maths.ox.ac.uk

A. J. Wathen
Numerical Analysis Group, Mathematical Institute, The University of Oxford, Andrew Wiles Building, Radcliffe Observatory Quarter, Woodstock Road, Oxford OX2 6GG, UK
e-mail: wathen@maths.ox.ac.uk

1 Introduction

Recent advances in radial basis function (RBF) theory have demonstrated the usefulness of RBFs for scattered data approximation in high dimensional spaces. RBFs are useful in the design of mesh-free methods [8,18,19], for global optimisation [11] and computer aided design [7,12,14,20,23,27,28]. It has for long been established that certain types of RBFs guarantee non-singular interpolation matrices provided the interpolating points are distinct [21]. There is no doubt that such a guaranteed solvability is one of the convincing reasons why methods based on RBFs continue to attract much attention. As is well known—and as we shall describe—the current solvability of the RBF interpolation problem depends on the symmetry property of the interpolation matrix; it requires that the centres of the basis functions are the interpolating points and that the basis functions are of the same scale. In practice it is often desirable to define/use RBF approximation/interpolation with *different shapes*—different scales or different bases. However, for RBF interpolation with different shape parameters, previous results say *little* on the issue of solvability. To make this more clear, the reader is invited to see how the previous proof works for the case of using radial basis functions of the same scale.

Consider the radial basis function (RBF) interpolation problem in high dimensional spaces: given observations $f_1, f_2, \ldots, f_n$ on a set of points $\mathscr{X} = \{\mathbf{x}_i \in \mathbb{R}^d : i = 1, \ldots n\}$, find an approximation of the form $s(\mathbf{x}) = \sum_{i=1}^{n} \alpha_j \phi_j(\mathbf{x})$ to an unknown function $f(\mathbf{x})$ such that

$$s(\mathbf{x}_k) = f_k, \text{ for } 1 \le k \le n, \tag{1}$$

where $\phi_j(\mathbf{x}) = \Phi(\mathbf{x} - \mathbf{x}_j) = \varphi(\|\mathbf{x} - \mathbf{x}_j\|)$. Equation (1) results in a linear system $A\underline{\alpha} = \underline{f}$, where $\underline{f} = (f_1, f_2, \ldots, f_n)^T$ and

$$A = \begin{pmatrix} \Phi(\mathbf{x}_1 - \mathbf{x}_1) & \Phi(\mathbf{x}_1 - \mathbf{x}_2) & \cdots & \Phi(\mathbf{x}_1 - \mathbf{x}_n) \\ \Phi(\mathbf{x}_2 - \mathbf{x}_1) & \Phi(\mathbf{x}_2 - \mathbf{x}_2) & \cdots & \Phi(\mathbf{x}_2 - \mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi(\mathbf{x}_n - \mathbf{x}_1) & \Phi(\mathbf{x}_n - \mathbf{x}_2) & \cdots & \Phi(\mathbf{x}_n - \mathbf{x}_n) \end{pmatrix}.$$

If Φ has an integrable Fourier transform $\hat{\Phi}$, then Φ can be recovered from $\hat{\Phi}$ by the Fourier inversion formula [28, p. 67]

$$\Phi(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^d}} \int_{\mathbb{R}^d} \hat{\Phi}(\omega) e^{i\mathbf{x}^T\omega} d\omega. \tag{2}$$

For any $\underline{\beta} \in \mathbb{C}^n$, consider the following quadratic form

$$\underline{\beta}^H A \underline{\beta} = \sum_{k=1}^{n}\sum_{j=1}^{n} \beta_j \bar{\beta}_k \Phi(\mathbf{x}_k - \mathbf{x}_j) = \frac{1}{\sqrt{(2\pi)^d}} \sum_{k=1}^{n}\sum_{j=1}^{n} \beta_j \bar{\beta}_k \int_{\mathbb{R}^d} \hat{\Phi}(\omega) e^{i(\mathbf{x}_k - \mathbf{x}_j)^T\omega} d\omega. \tag{3}$$

Separate $e^{i(\mathbf{x}_k - \mathbf{x}_j)^T\omega}$ as $e^{i\mathbf{x}_k^T\omega} e^{-i\mathbf{x}_j^T\omega}$, then

$$\sum_{k=1}^{n}\sum_{j=1}^{n} \beta_j \bar{\beta}_k e^{i(\mathbf{x}_k - \mathbf{x}_j)^T\omega} = \sum_{j=1}^{n} \beta_j e^{-i\mathbf{x}_j^T\omega} \sum_{k=1}^{n} \bar{\beta}_k e^{i\mathbf{x}_k^T\omega} = \left| \sum_{j=1}^{n} \beta_j e^{-i\mathbf{x}_j^T\omega} \right|^2. \tag{4}$$

Therefore it follows that

$$\underline{\beta}^H A \underline{\beta} = \frac{1}{\sqrt{(2\pi)^d}} \int_{\mathbb{R}^d} \hat{\Phi}(\omega) \left| \sum_{j=1}^{n} \beta_j e^{-i\mathbf{x}_j^T \omega} \right|^2 d\omega. \tag{5}$$

If $\hat{\Phi} > 0$, then the interpolation matrix A is symmetric and positive definite. Such functions with positive Fourier transform are called *positive definite functions*. It is obvious that positive definite radial basis functions $\varphi(\|\cdot\|)$ guarantee invertible interpolation matrices. One of the most famous positive definite radial basis functions is the Gaussian $e^{-\epsilon^2 \|\mathbf{x}\|_2^2}$, with Fourier transform $\frac{1}{(\sqrt{2}\epsilon)^d} e^{-\frac{\|\omega\|^2}{4\epsilon^2}}$. Also included in this class are inverse multiquadrics, truncated power functions, Wu and Wendland compactly supported radial basis functions. See [30] [28, p. 76, p. 80, p. 128] for details.

A natural generalization of *a positive definite function* is *a conditional positive definite function*. A conditional positive definite function is a continuous function which guarantees the quadratic form $\underline{\beta}^H A \underline{\beta}$ is positive on a subspace, where $\underline{\beta} \in \mathbb{C}^n / \{0\}$ satisfies $\sum_{j=1}^{n} \beta_j p(\mathbf{x}_j) = 0$ for any distinct point set $\mathscr{X}$ and all complex-valued polynomials, $p(\mathbf{x})$, of degree less than m. For such conditional positive definite functions, one can prove that a modified approximation

$$s(x) = \sum_{j=1}^{n} \alpha_j \phi_j(\mathbf{x}) + \sum_{k=1}^{Q} \gamma_k p_k(\mathbf{x}) \tag{6}$$

with lower order polynomial constraints is solvable under certain conditions;

$$\sum_{j=1}^{n} \alpha_j p_k(\mathbf{x}_j) = 0, \text{ for } 1 \le k \le Q, \tag{7}$$

where p_k, $1 \le k \le Q$ is a linear independent basis of the space of polynomials of degree less than m, $\pi_{m-1}(\mathbb{R}^d)$. The approximation (6) together with the polynomial constraints results in a saddle point system

$$\begin{pmatrix} A & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} \underline{\alpha} \\ \underline{\gamma} \end{pmatrix} = \begin{pmatrix} \underline{f} \\ 0 \end{pmatrix} \tag{8}$$

If P is full rank, and the quadratic form $\underline{\alpha}^H A \underline{\alpha}$ is positive on the null space of P^T, i.e. $\{\underline{\alpha} : P^T \underline{\alpha} = 0\}$, then the saddle point system (8) has a unique solution [28, p. 117]. Whether P is full rank still depends on the point set $\mathscr{X}$ for $\pi_{m-1}(\mathbb{R}^d)$ with $m > 1$, whereas Micchelli proves that $\underline{\alpha}^H A \underline{\alpha}$ is positive on the null space of P^T is always true for all conditional positive definite functions of order m on any data set $\mathscr{X}$ [21]. His proof in fact also depends on the condition that the quadratic form (5) is positive on the null space of P^T, which as a consequence requires that the basis functions are of the same scale according to (4). See [21, p. 17] for details.

If the interpolation points and the basis function centres are different, or the basis functions in (1) are in different scales or different kinds of functions, the formula (4) does not hold. An example of the latter case, using different kinds of radial basis function, is the unsymmetrical collocation method for solving partial differential equations (PDEs) [16], where the author uses basis functions of different scales or different combinations at each centre. Such an unsymmetrical collocation method works most of the time, but singularity still can occur for some cases because there is no theory to guarantee the unsymmetrical collocation matrix is always invertible. For complicated real applications, many are eager for an adaptive

refinement scheme, arranging more data where it should be, to reduce the total computing complexity. In this case, one also wants to use radial basis functions with different scales for the locally clustered points; if two basis function centres are too close to each other, the corresponding two columns of the system matrix A are nearly linearly dependent, and thus the matrix can be ill-conditioned. Whereas if such two basis functions employ different scales or are different, then the conditioning may be improved [10]. The eagerness of using radial basis functions with different scales dates back to the 1970s [13], whereas few practical results for large data sets were obtained. There are papers which discuss how to choose various shape parameters both numerically [3,6,9,15,17,22,32] and theoretically [1,3]. In particular in [1], Bozzini et al. use matrix perturbation arguments, concluding that if the scales do not vary a lot across the domain, then interpolation with various scales can result in a non-singular interpolation matrix. The threshold of variation of the scales depends on the estimation of the smallest eigenvalue of an interpolation matrix with the same scale [1, Theorem 2]; such a dependence limits the utility of the method, but see the recent paper [2]. Solvability for more cases remains to be established for general radial basis functions.

We focus on compactly supported radial basis functions [28,30]. Sufficient conditions to guarantee a unique solution to the RBF interpolation problem with different shapes are supplied. The conditions only depend on *a local convexity* of the underlying basis functions, and a *local geometric property* of a neighbourhood of each centre. Information on both of these aspects are easy to obtain. Algorithms can be implemented efficiently, whereas the error estimate for such a scheme is unclear at moment and some numerical examples show that the scheme may not guarantee good approximation quality, there is still some posibility to construct a realistic local refinement scheme according to this approach. At moment, the scheme can be used in inaccurate models, like some difficult 3D surface reconstruction model, where efficiency is more important and high resolution is not crucial. When it is applied on the Stanford scanning repository data sets, reasonable results can be obtained.

Here we share our results, so that others may find a better solution and improvement. We first introduce notation and preliminaries in Sect. 2 and then present the main results in Sect. 3. We further discuss several issues related to the main results in Sect. 4 and put algorithms and numerical results in Sects. 5 and 6 respectively. Finally we give some conclusions in Sect. 7.

2 Convexity of Compactly Supported Radial Basis Functions

Let $f(r) \in C^2[a,b]$, if $f'(r) < 0$ and $f''(r) \geq 0$, then $f(r)$ is convex on (a,b). (We suppose a linear function with negative slope is convex in this paper.) This section focuses on one popular class of compactly supported radial basis function—Wendland functions.

A Wendland function $\phi_{d,k}$ is obtained through the following integral operator

$$\mathscr{I}(\phi)(r) := \int_r^\infty t\phi(t)dt. \tag{9}$$

The Wendland function $\phi_{d,k}$ is defined as

$$\phi_{d,k} = \mathscr{I}^k \phi_{\lfloor d/2 \rfloor + k + 1}, \tag{10}$$

where $\phi_\ell(r) = \max\{(1-r)^\ell, 0\}$ and $\lfloor \cdot \rfloor$ is the floor operator. Here, $\phi_{d,k}(\|\mathbf{x}\|)$ is a compactly supported radial basis function and $\phi_{d,k} \in C^{2k}(\mathbb{R}^d)$. The operator $\mathscr{I}$ was introduced to construct compactly supported radial basis functions [30]. An inversion like operator $\mathscr{D}$ was also introduced to simplify computations for high dimensional Fourier transforms of radial functions [30] [28, p. 121]:

$$\mathscr{D}\phi(r) = -\frac{1}{r}\phi'(r) \text{ for } \phi \in C^2(\mathbb{R}) \text{ and } r > 0. \tag{11}$$

It is known that $\mathscr{D}\mathscr{I}\phi = \phi$ and $\mathscr{I}\mathscr{D}\phi = \phi$ [28, p. 121]. Applying the formula (11), we can get the following properties of functions defined in (9).

Lemma 1 *If $\phi \in L_1[0,\infty)$ is a non-negative function, then,*

1. $\frac{d}{dr}\mathscr{I}^k\phi(r) = -r\mathscr{I}^{k-1}\phi(r) \le 0$, *for* $k \ge 1$;
2. $\frac{d^2}{dr^2}\mathscr{I}^k\phi(r) = r^2\mathscr{I}^{k-2}\phi(r) - \mathscr{I}^{k-1}\phi(r)$, *for* $k \ge 2$.

Proof Since ϕ is non-negative, by the definition of the operator $\mathscr{I}$ in (9), $\mathscr{I}^k\phi(r)$ is positive. Employing the formula (11), we can get

$$-\frac{1}{r}\frac{d}{dr}\mathscr{I}^k\phi(r) = \mathscr{D}\mathscr{I}^k\phi(r) = \mathscr{I}^{k-1}\phi(r) \ge 0.$$

Reformulating the equation we can obtain

$$\frac{d}{dr}\mathscr{I}^k\phi(r) = -r\mathscr{I}^{k-1}\phi(r) \le 0,$$

for $k \ge 1$. The second part holds because

$$\frac{d^2}{dr^2}\mathscr{I}^k = \frac{d}{dr}(-r\mathscr{I}^{k-1}\phi(r)) = r^2\mathscr{I}^{k-2}\phi(r) - \mathscr{I}^{k-1}\phi(r).$$

□

The first part of Lemma 1 demonstrates that Wendland functions are non-increasing on $[0, 1]$, and the second part gives a recurrence formula to compute the second derivative of a Wendland function.

Lemma 2 *If $\phi(r)$ is a continuous and monotonically decreasing function on* $[0, 1]$, *which satisfies $\phi(0) > 0$ and $\phi(1) = 0$, then for the function*

$$f(r) = r^2\phi(r) - \int_r^1 t\phi(t)dt,$$

there exists a positive number $\gamma \in (0, 1)$ such that $f(r) \ge 0$ on $[\gamma, 1]$.

Proof Since ϕ is a monotonically decreasing function on $[0, 1]$, then for any $t \in (r, 1)$, $0 < \phi(t) < \phi(r)$. By the Cauchy–Schwarz inequality,

$$\begin{aligned}\int_r^1 t\phi(t)dt &\le \left(\int_r^1 t^2dt\right)^{1/2}\left(\int_r^1 \phi^2(t)dt\right)^{1/2} \\ &\le \sqrt{\frac{1-r^3}{3}}\left(\phi(r)\sqrt{1-r}\right) \le \phi(r)\sqrt{\frac{1-r^3}{3}}.\end{aligned}$$

Therefore

$$f(r) \ge \left(r^2 - \sqrt{\frac{1-r^3}{3}}\right)\phi(r) = h(r)\phi(r). \tag{12}$$

Compute $h'(r) = 2r + \frac{\sqrt{3}r^2}{2\sqrt{1-r^3}} > 0$ on $(0, 1)$. On the other hand, $h(0) = -\sqrt{1/3}$ and $h(1) = 1 > 0$. Therefore, according to the intermediate value theorem for continuous functions, there must exists a positive number $\gamma \in (0, 1)$, such that $h(\gamma) = 0$ and $h(r) > 0$ on $(\gamma, 1)$. Thus $f(r) \ge 0$ on $[\gamma, 1]$. □

Theorem 1 *For any Wendland Function $\phi_{d,k}$ defined above, there exists a positive real number $q \in [0, 1)$, such that $\phi_{d,k}$ is convex on $[q, 1]$.*

Proof If $k = 0$, $\phi_{d,0} = (1-r)^{\lfloor d/2 \rfloor + 1}$, then

$$\phi'_{d,0}(r) = -1(\lfloor d/2 \rfloor + 1)(1-r)^{\lfloor d/2 \rfloor} < 0$$

on $(0, 1)$. If $d = 1$, $\phi''_{1,0}(r) = 0$, otherwise

$$\phi''_{d,0}(r) = (-1)^2(\lfloor d/2 \rfloor + 1)(\lfloor d/2 \rfloor)(1-r)^{\lfloor d/2 \rfloor - 1} \geq 0.$$

Thus $\phi_{d,0}(r)$ is convex on $[0, 1]$.

For $k = 1$, $\phi_{d,1}(r) = \int_r^1 t(1-t)^{\lfloor d/2 \rfloor + 1 + 1} dt$,

$$\phi'_{d,1}(r) = -r(1-r)^{\lfloor d/2 \rfloor + 2} \leq 0 \text{ in} (0, 1),$$
$$\phi''_{d,1}(r) = (1-r)^{\lfloor d/2 \rfloor + 1} \left((\lfloor d/2 \rfloor + 3)r - 1\right).$$

Let $q = \frac{1}{\lfloor d/2 \rfloor + 3}$, then $\phi''_{d,1}(r) \geq 0$ on $[q, 1]$.

For $k \geq 2$, according to Lemma 1, $\phi'_{d,k}(r) \leq 0$ and

$$\phi''_{d,k}(r) = r^2 \mathscr{I}^{k-2}\phi_\ell(r) - \mathscr{I}^{k-1}\phi_\ell(r),$$

where $\ell = \lfloor d/2 \rfloor + k + 1$, and $\phi_\ell(r) = (1-r)_+^\ell$. Define $\varphi(r) = \mathscr{I}^{k-2}\phi_\ell(r)$, then $\varphi(r)$ is a decreasing and non-negative function on $[0, 1]$. Note that

$$\mathscr{I}^{k-1}\phi_\ell(r) = \int_r^1 t\varphi(t)dt.$$

Employing Lemma 2, there exists a positive $0 < \gamma < 1$, such that $\phi''_{d,k}(r)$ is positive on $[\gamma, 1]$. Let $q = \gamma$, which finishes the proof. □

Lemma 3 *For any Wendland function $\phi_{d,k}(r)$, there exists a positive number m and a positive constant c such that for $r \in (\delta_1, 1)$, $\phi_{d,k}(r) \leq c(1-r)^m$.*

Proof Since $\phi_{d,k}$ vanishes at 1, thus $\phi_{d,k} = (1-r)^m p(r)$ on $[0, 1]$, where $p(r)$ is a polynomial with non-zeros on $[0, 1]$. Let $c = \max_{r\in[0,1]} p(r)$, since $\phi_{d,k}$ is on $[0, 1)$, thus $c > 0$. □

3 Main Results

Let $\mathscr{N}(\mathbf{x}_j, n_j) \subset \mathscr{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \subset \mathbb{R}^d$, be the set of the n_j nearest neighbouring points to $\mathbf{x}_j$, where $\mathbf{x}_j \notin \mathscr{N}(\mathbf{x}_j, n_j)$. For each $\mathscr{N}(\mathbf{x}_j, n_j)$, define

$$r_j = \min_{\mathbf{x}_k \in \mathscr{N}(\mathbf{x}_j, n_j)} \|\mathbf{x}_k - \mathbf{x}_j\| \quad \text{and} \quad R_j = \max_{\mathbf{x}_k \in \mathscr{N}(\mathbf{x}_j, n_j)} \|\mathbf{x}_k - \mathbf{x}_j\|. \tag{13}$$

It is obvious that $\mathscr{N}(\mathbf{x}_j, n_j) \subset \mathscr{B}(\mathbf{x}_j, R_j) \backslash \mathscr{B}(\mathbf{x}_j, r_j)$, where $\mathscr{B}(\mathbf{x}, R)$ is a Euclidean ball with center $\mathbf{x}$ and radius R. Further, define

$$q_j = \frac{r_j}{R_j}, \text{ and } m_j = \frac{1}{n_j R_j} \sum_{\mathbf{x}_k \in \mathscr{N}(\mathbf{x}_j, n_j)} \|\mathbf{x}_k - \mathbf{x}_j\|. \tag{14}$$

It is easy to find that $0 < q_j \leq m_j \leq 1$, the equalities holding only when all of the points in $\mathscr{N}(\mathbf{x}_j, n_j)$ are located on the same sphere with $\mathbf{x}_j$ as center (including the special case

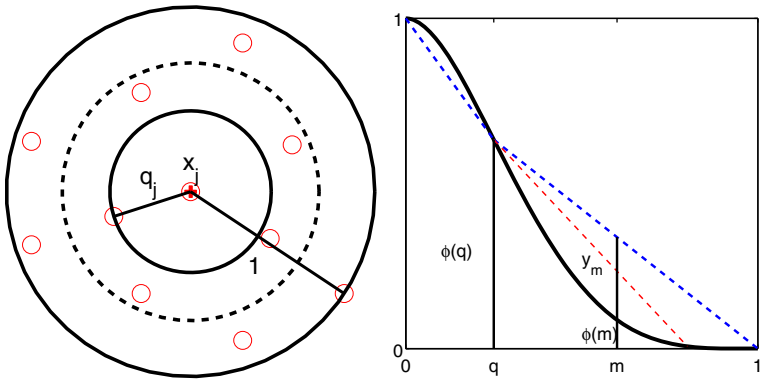


Fig. 1 **a** Illustrates a 'scaled' neighbourhood of $\mathbf{x}_j$ with 10 neighbouring points. **b** Illustrates a Wendland function which is convex in $(q, 1)$. One can see that $\frac{y_m}{\phi(q)} = \frac{1-m}{1-q}$ holds, which is used to prove Theorem 2

$n_j = 1$). The quantities q_j and m_j are associated with the particular geometric property of $\mathcal{N}(\mathbf{x}_j, n_j)$: the points in $\mathcal{N}(\mathbf{x}_j, n_j)$ are located in $\mathcal{B}(\mathbf{x}_j, R_j)\backslash\mathcal{B}(\mathbf{x}_j, r_j)$, and can be viewed as distributed on the sphere $\|\mathbf{x}_j - \mathbf{x}\|_2 = m_j R_j$ on average. Such a local geometric property only depends on the relative distances between $\mathbf{x}_j$ and points in $\mathcal{N}(\mathbf{x}_j, n_j)$ (see Fig. 1a for an illustration).

Denote the radial basis functions $\varphi(\|\mathbf{x}\|)$ with different scales, ϵ_j, as $\phi_{\epsilon_j}(\mathbf{x}) = \varphi(\epsilon_j\|\mathbf{x}\|)$, for $1 \le j \le n$. The corresponding approximation to (1) is written as

$$s(\mathbf{x}) = \sum_{j=1}^{N} \alpha_j \phi_{\epsilon_j}(\mathbf{x}) \tag{15}$$

and the corresponding interpolation matrix on the data set $\mathcal{X}$ with $\{\phi_{\epsilon_j}\}_{j=1}^n$ is denoted as $A_{\phi_{\epsilon_j},\mathcal{X}}$.

Theorem 2 *Let $\phi(r)$ be a univariate compactly supported radial basis function on $[0, 1]$ which is convex on $[q, 1]$ and satisfies $\phi(0) = 1$. $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ is a scattered data set in $\mathbb{R}^d$. For a set $\mathcal{N}(\mathbf{x}_j, n_j)$ of n_j nearest neighbouring points, R_j is defined by (13) and q_j and m_j are defined by (14). If for every $\mathbf{x}_j \in \mathcal{X}$, there is a set $\mathcal{N}(\mathbf{x}_j, n_j)$ of n_j nearest neighbouring points which satisfies $q_j \ge q$ and $m_j > 1 - \frac{1-q}{\phi(q)n_j}$, then the interpolation matrix $A_{\phi_{\epsilon_j},\mathcal{X}} = \phi(\epsilon_j\|\mathbf{x}_i - \mathbf{x}_j\|)_{1\le i,j\le N}$ is non-singular when $\epsilon_j \ge 1/R_j$.*

Proof For $\mathbf{x}_k \in \mathcal{N}(\mathbf{x}_j, n_j)$, define $d_{kj} = \frac{\|\mathbf{x}_k - \mathbf{x}_j\|}{R_j}$. Since $q_j = \frac{r_j}{R_j} \ge q$, then for each k and j, $d_{kj} \ge q_j \ge q$. Suppose now y_{kj} is defined so that the points (d_{kj}, y_{kj}) are located on the straight line which passes through $(q, \phi(q))$, and $(1, 0)$, then $y_{kj} = \frac{1-d_{kj}}{1-q}\phi(q)$ (see Fig. 1 for an illustration). Since $d_{kj} \in (q_j, 1) \subset [q, 1]$ and ϕ is convex on $[q, 1]$, then $y_{kj} \ge \phi(d_{kj})$. It follows that

$$\sum_{\mathbf{x}_k\in\mathcal{N}(\mathbf{x}_j,n_j)} y_{kj} = \frac{\phi(q)}{1-q} \sum_{\mathbf{x}_k\in\mathcal{N}(\mathbf{x}_j,n_j)} (1 - d_{kj}) = \frac{n_j\phi(q)}{1-q}(1 - m_j) \le 1, \tag{16}$$

under the stated assumption that $m_j > 1 - \frac{1-q}{\phi(q)n_j}$. This proves that when the scale parameter ϵ_j is taken as $\epsilon_j = 1/R_j$ then for each j we must have

$$1 - \sum_{\mathbf{x}_k \in \mathcal{N}(\mathbf{x}_j, n_j)} \phi(d_{kj}) = \phi(0) - \sum_{\mathbf{x}_k \in \mathcal{N}(\mathbf{x}_j, n_j)} \phi(\epsilon_j \|\mathbf{x}_k - \mathbf{x}_j\|_2) > 0.$$

That is: the interpolation matrix $A_{\phi_{\epsilon_j}, \mathscr{X}}$ is diagonally dominant and thus non-singular. For the case $\epsilon_j \geq 1/R_j$, it follows because

$$g(\epsilon) := \phi(0) - \sum_{\mathbf{x}_k \in \mathcal{N}(\mathbf{x}_j, n_j)} \phi(\epsilon \|\mathbf{x}_k - \mathbf{x}_j\|_2) \tag{17}$$

is a decreasing function on ϵ. □

Remark 1 If $n_j = 1$, then $q_j = m_j = 1 > 1 - \frac{1-q}{\phi(q)n_j}$, Theorem 2 always holds. In this case the interpolation matrix is a diagonal matrix, and of course non-singular. However, this case is not of interest. For $n_j = 2$, then the non-zero off-diagonal elements are always smaller than 1 if $\epsilon_j = 1/R_j$, even if $\mathcal{N}(\mathbf{x}_j, 2)$ does not satisfy the condition. When $q_j < q$, the nearest point is too close to $\mathbf{x}_j$ and the two basis functions are more likely to correlate with each other.

Example 1 Consider the Wendland function $\phi_{3,1}(r) = (1-r)_+^4(4r+1)$; its convex interval is $[0.25, 1]$. Consider a point $\mathbf{x}_j$ on the equally spaced mesh in $\mathbb{R}^2$; see Fig. 2a for illustration. Let $R_j = 1$ for $\mathcal{N}(\mathbf{x}_j, n_j)$, $n_j = 5, 6, 7, 8$. $q_j = \frac{\sqrt{2}}{2} > 0.25$, $m_j = \frac{\sqrt{2}}{4} + \frac{1}{2} \approx 0.8536$ for $n_j = 8$.

$$1 - \frac{1 - 0.25}{8\phi_{3,1}(0.25)} \approx 0.8519 < m_j.$$

Therefore on the equally spaced mesh in $\mathbb{R}^2$, $\mathcal{N}(\mathbf{x}_j, n_j)$ for $n_j \leq 8$ satisfy Theorem 2. This also illustrates that on a equally spaced mesh, the radii for a compact support basis function which satisfies Theorem 2 is proportional to the underlying grid size.

Theorem 3 *Let ϕ be a compactly supported Wendland radial basis function, for any set of distinct points $\mathscr{X} = \{\mathbf{x}_k \in \mathbb{R}^d, k = 1, 2, \ldots, n.\}$, there exist scale parameters ϵ_j, such that each column of the interpolation matrix $A_{\phi_{\epsilon_j}, \mathscr{X}} = \phi(\epsilon_j \|\mathbf{x}_i - \mathbf{x}_j\|_2)_{1 \leq i,j \leq n}$ is strictly column diagonally dominant and each column has at least 3 non-zero elements.*

Proof For given $\mathbf{x}_j$, let

$$\rho_1 = \min_{i \neq j} \|\mathbf{x}_i - \mathbf{x}_j\|, \rho_{m+1} = \min_{\|\mathbf{x}_k - \mathbf{x}_j\| > \rho_m} \|\mathbf{x}_k - \mathbf{x}_j\|, \text{ for } m > 1,$$

and

$$N(\rho) := \#\{\mathbf{x}_k : 0 < \|\mathbf{x}_k - \mathbf{x}_j\| < \rho\}.$$

be the number of points whose distance to $\mathbf{x}_j$ smaller than ρ, then for any $\delta > 0$ such that $\rho_1 + \delta < \rho_2$, we have $N(\rho_1) = N(\rho_1 + \delta)$. According to Lemma 3, we have

$$\sum_{\mathbf{x}_k \in \mathcal{N}(\mathbf{x}_j, N(\rho))} \phi\left(\frac{\|\mathbf{x}_k - \mathbf{x}_j\|}{\rho}\right) \leq cN(\rho)\left(1 - \frac{\rho_1}{\rho}\right)^m. \tag{18}$$

If $N(\rho_1) \geq 2$, let $\rho = \delta + \rho_1 \leq \rho_2$, and force the right hand side in (16) to be less than one, we get

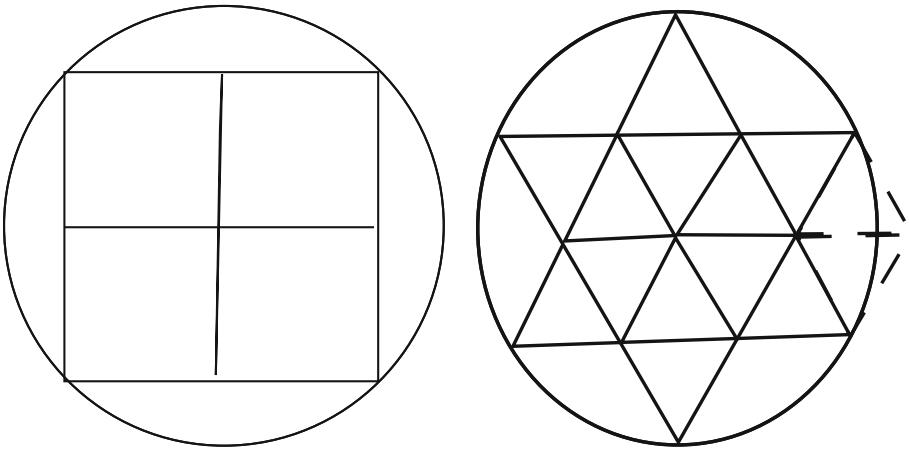


Fig. 2 Two examples for the supports of basis functions which satisfy a diagonal dominant condition. **a** On an equally spaced regular mesh. **b** On a mesh filled by isosceles triangles

$$\delta \le \frac{\rho_1 T}{1-T} = \delta^*, \text{ where } T = \left(\frac{1}{cN(\rho_1)}\right)^{1/m}.$$

Then for any $0 < \delta < \min\{\delta^*, \rho_2 - \rho_1\}$, the shape parameter $\epsilon_j = \frac{1}{\rho_1+\delta}$ can make the jth column have $N(\rho_1)+1$ non-zero elements and column diagonally dominant.

For the case $N(\rho_1) = 1$, let $\rho := \rho_2 + \delta < \rho_3$, then

$$\begin{aligned}\sum_{\mathbf{x}_k \in \mathcal{N}(\mathbf{x}_j, N(\rho))} \phi\left(\frac{\|\mathbf{x}_k - \mathbf{x}_j\|}{\rho}\right) &\le \phi\left(\frac{\rho_1}{\rho_1+\delta}\right) + c(N(\rho_2)-1)\phi\left(\frac{\rho_2}{\rho_2+\delta}\right) \\ &\le \phi\left(\frac{\rho_1}{\rho_1+\delta}\right) + c(N(\rho_2)-1)\left(\frac{\delta}{\rho_2+\delta}\right)^m \\ &\le \phi\left(\frac{\rho_1}{\rho_1+\delta}\right) + c(N(\rho_2)-1)\left(\frac{\delta}{\rho_2+\delta}\right) := f(\delta).\end{aligned}$$

Thus $f(0) = \phi(\frac{\rho_1}{\rho_2}) < 1$ and $f(\delta)$ is a monotonically increasing function. Suppose $f(\delta) \le 1$ has solutions on $(0, \delta^*)$ for some δ^*. Then for any $0 < \delta \min\{\delta^*, \rho_3 - \rho_2\}$, the choice $\epsilon_j = \frac{1}{\rho_2+\delta}$ makes the column have $N(\rho_2)+1$ non-zero elements whilst keeping the diagonal dominance condition. □

Remark 2 Theorem 3 indicates that a point set $\mathscr{X}$ satisfying the Theorem is a sufficient but not necessary condition to guarantee a non-singular interpolation matrix with RBFs of different shapes.

Example 2 Consider a mesh which consists of isosceles triangles as illustrated in Fig. 2b. Since $N(\rho_1) = 6$, $\mathscr{N}(\mathbf{x}_j, n_j)$ for $n_j \le 6$ satisfy a diagonal dominant condition but they result in interpolation matrices with only one non-zero element in column j. For $N(\rho_2) = 12$, let $\rho_2 = R_j = 1$, where R_j corresponds to $\mathscr{N}(\mathbf{x}_j, n_j)$ for $n_j = 7$ to 12, then $q_j = \frac{\sqrt{3}}{3} > 0.25$ and $1 - 6 * \phi_{3,1}(q_j) \approx 0.3664 > 0$. Therefore, when using $\phi_{3,1}$ with radius as ρ_2, a diagonal dominant condition can also be guaranteed. The largest n_j such that $\mathscr{N}(\mathbf{x}_j, n_j)$ satisfies Theorem 2 is 8.

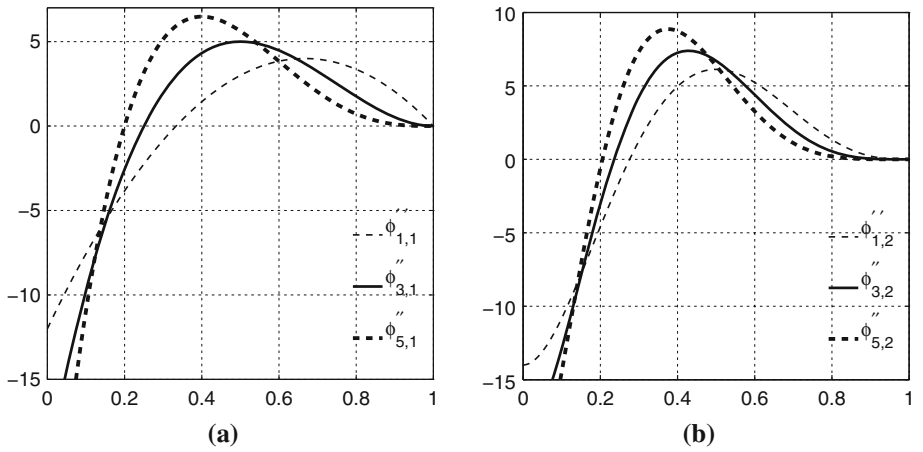


Fig. 3 The second derivatives of Wendland function $\phi_{d,k}(r)$ for $d = 1, 3, 5$ and $k = 1, 2$. The root of $\phi''_{d,k}(r)$ in (0,1) is the exactly tight bound of q

Remark 3 Both Example 1 and Example 2 illustrate that the radii of the basis functions which can guarantee a diagonal dominant interpolation matrix is a relative value to the mesh size.

Finally, we conclude that interpolation matrices constructed from Theorems 2 and 3 have an nice property which can guarantee (incomplete) LU factorizations. Matrices with such a property are called H-matrices. An H-matrix is a generalization of an M-matrix. A real matrix $A = (a_{i,j})_{1\le,i,j\le n}$ with $a_{i,j} \le 0$ for all $i \ne j$ is an M-matrix if A is non-singular and A^{-1} is a non-negative matrix. A matrix A is an H-matrix if its comparison matrix $\mathscr{M}(A)$ is an M-matrix, where the comparison matrix $\mathscr{M}(A) = (\alpha_{i,j})_{1\le i,j,\le n}$ is defined by

$$\alpha_{i,i} := |a_{i,i}|, \text{ and } \alpha_{i,j} := -|a_{i,j}| \text{ for } i \ne j, (1 \le i, j \le n).$$

H-matrices can guarantee stable incomplete factorization pre-conditioners [26].

Theorem 4 *The interpolations matrices in Theorems 2 and 3 are H-matrices.*

Proof Theorems 2 and 3 guarantee the underlying interpolation matrix strictly diagonally dominant. The result follows due to an established result: if a matrix is is either a strictly diagonally dominant or an irreducibly diagonally dominant matrix, then A is an H-matrix [25, p. 92, Theorem 3.27]. □

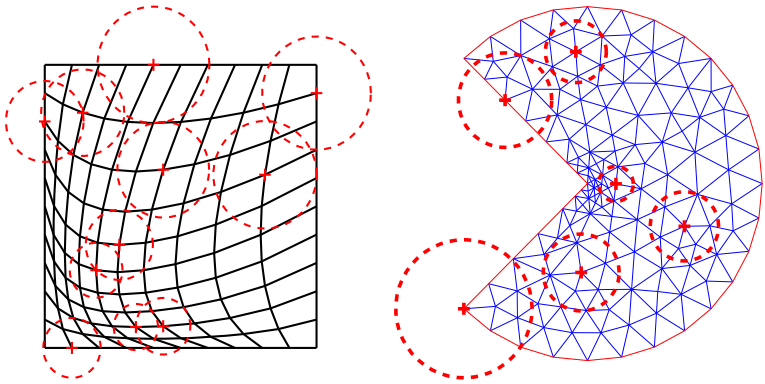
4 Further Discussion

4.1 Estimation of the Lower Bound of the Convex Interval

A more accurate estimation on the threshold q for the convex interval $[q, 1]$ can supply a more tight lower bound on m_j. For Wendland function $\phi_{d,0}$, we can take $q = 0$; for $\phi_{d,1}$, $q = \frac{1}{\lfloor d/2 \rfloor + 3}$. In both cases q is minimal. It is easy to see that the lower bound of the convex interval $\phi_{d,1}$ becomes smaller as d increases. This is also true for $\phi_{d,2}$ when $d = 1, 3, 5$. See Fig. 3. Though neither Theorem 2 nor Lemma 2 indicates there is only one root for $\phi''_{d,k}(r) = 0$ for $k \ge 2$, this is true for the case $k = 2$ and $d = 1, 3, 5$. It is possible that $\phi''_{d,k}$

Table 1 Estimations of the convex interval for Wendland functions

Smoothness	Function	Convex interval
C^0	$\phi_{1,0} = (1-r)_+$	$[0, 1]$
C^2	$\phi_{1,1} = (1-r)_+^3(3r+1)$	$[\frac{1}{3}, 1]$
C^4	$\phi_{1,2} = (1-r)_+^5(8r^2+5r+1)$	$[0.2760, 1]$
C^0	$\phi_{3,0} = (1-r)_+^2$	$[0, 1]$
C^2	$\phi_{3,1} = (1-r)_+^4(4r+1)$	$[\frac{1}{4}, 1]$
C^4	$\phi_{3,2} = (1-r)_+^6(35r^2+18r+3)$	$[0.2356, 1]$
C^0	$\phi_{5,0} = (1-r)_+^3$	$[0, 1]$
C^2	$\phi_{5,1} = (1-r)_+^5(5r+1)$	$[\frac{1}{5}, 1]$
C^4	$\phi_{5,2} = (1-r)_+^7(16r^2+7r+1)$	$[0.2056, 1]$



(a) **(b)**

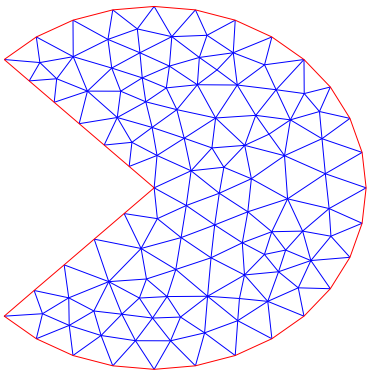
Fig. 4 A demonstration of radial basis function with different supports for non-uniform grid. **a** $\mathcal{N}(\mathbf{x}_j, 6)$ for a distort grid. **b** $\mathcal{N}(\mathbf{x}_j, 7)$ for a FEM mesh

at most has only one zero in $(0, 1)$. The estimations of the threshold q for several Wendland function in Table 1 are computed by Mathematica.

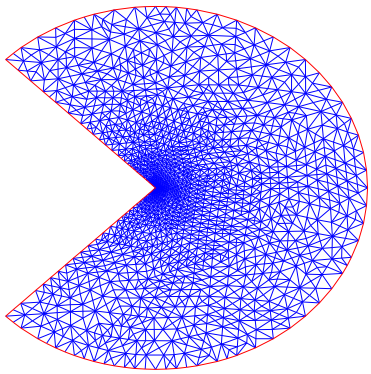
4.2 Sharper Bound for m_j and Eigenvalue Distribution

From Fig. 1, as one might predict, the lower bound of m_j in Theorem 2, $1 - \frac{1-q}{n_j q}$, can be far from tight in most cases. For a tighter bound of m_j, one can use $\frac{1-q_j}{n_j q_j}$ in (16). One even can verify the diagonal dominance by "brute force"—evaluate $\phi(\epsilon_j \|\mathbf{x}_k - \mathbf{x}_j\|)$ directly, but this involves more computations, every time changing the support or the size of the neighbourhood means that one has to re-evaluate the function. However, even if only a part of the centres are checked according to Theorem 2, it can save a lot of computations when adaptively choosing the support.

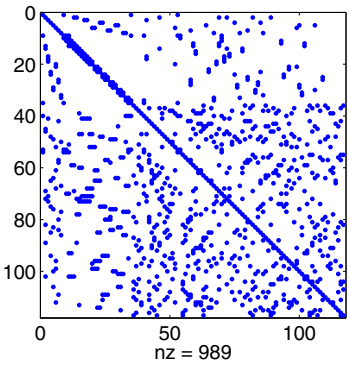
In either case, one can guarantee that the smallest eigenvalue of the interpolation matrix is bounded significantly away from the origin. The diagonal dominance indicates that $\mu = \|A_{\phi_{\epsilon_j}, \mathscr{X}}\|_1 < 2$. According to the classical Gerschgorin disc theorem [25, p. 16], all the



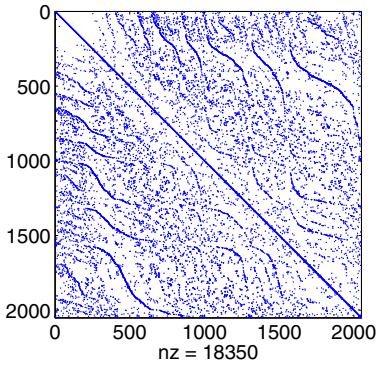
(a) TriDisc1 data set



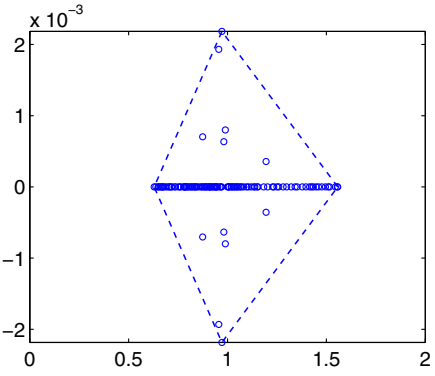
(b) TriDisc2 data set



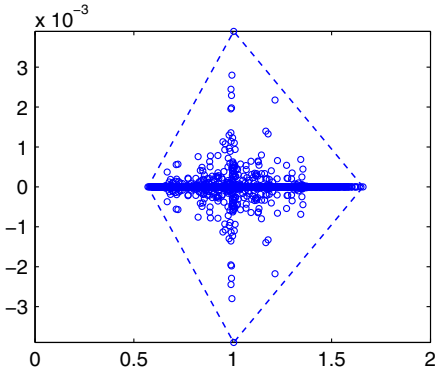
(c) TriDisc1 interpolation matrix



(d) TriDisc2 interpolation matrix



(e) TriDisc1 eigenvalues



(f) TriDisc2 eigenvalues

Fig. 5 Data sets, interpolation matrices, and eigenvalues distribution for `TriDisc` problem. The *blue circles* in (**e**) and (**f**) are eigenvalues, and the *blue dashed lines* are the boundaries of the convex hull of all the eigenvalues. **a** TriDisc1 data set. **b** TriDisc2 data set. **c** TriDisc1 interpolation matrix. **d** TriDisc2 interpolation matrix. **e** TriDisc1 eigenvalues. **f** TriDisc2 eigenvalues

Table 2 Information of several test problems by Algorithm 2

Prob	A		Eig Bounds		n_j			Scale range (R_j)			
	size	nnz	$\frac{\lambda_{\max}}{\lvert\lambda\rvert_{\min}} \le$	$\mathrm{Re}(\lambda) \ge$	min	max	mean	min	max	mean	max/min
HC	252	1,704	4.0	0.45	6	8	6.7	0.1095	0.3568	0.2275	3.3
T1	117	989	7.6	0.23	6	14	8.5	0.2163	0.4	0.2824	1.8
T2	2,050	18,350	10.8	0.17	4	14	9.0	3.8e−4	0.1515	5.0e−2	391.9
B4	1,351	9,911	10.7	0.17	3	16	7.3	3.5e−3	1.6e−2	1.0e−2	4.6
B3	5,643	42,825	8.8	0.20	3	17	7.6	1.6e−3	8.7e−3	5.3e−3	5.6
B2	24,425	183,988	12.0	0.15	3	21	7.5	1.3e−4	4.2e−3	2.6e−3	32.2
B1	105,615	740,493	18.9	0.10	3	21	7.0	5.8e−6	2.9e−3	1.2e−3	510.3
D3	15,563	117,362	10.1	0.18	3	24	7.5	2.6e−4	5.8e−3	3.51-3	22.0
D2	68,830	512,376	14.1	0.13	3	19	7.4	1.5e−4	3.4e−3	1.7e−3	22.2
D1	300,298	2,136,353	13.3	0.14	2	24	7.1	2.1e−5	2.3e−3	7.9e−4	109.8

eigenvalues of the interpolation matrix are located in a disc with center $(1, 0)$ and radius $\mu - 1$. The upper bound on the ratio of the largest eigenvalue to the smallest is $\frac{\mu}{2-\mu}$.

4.3 The Range of R_j and n_j

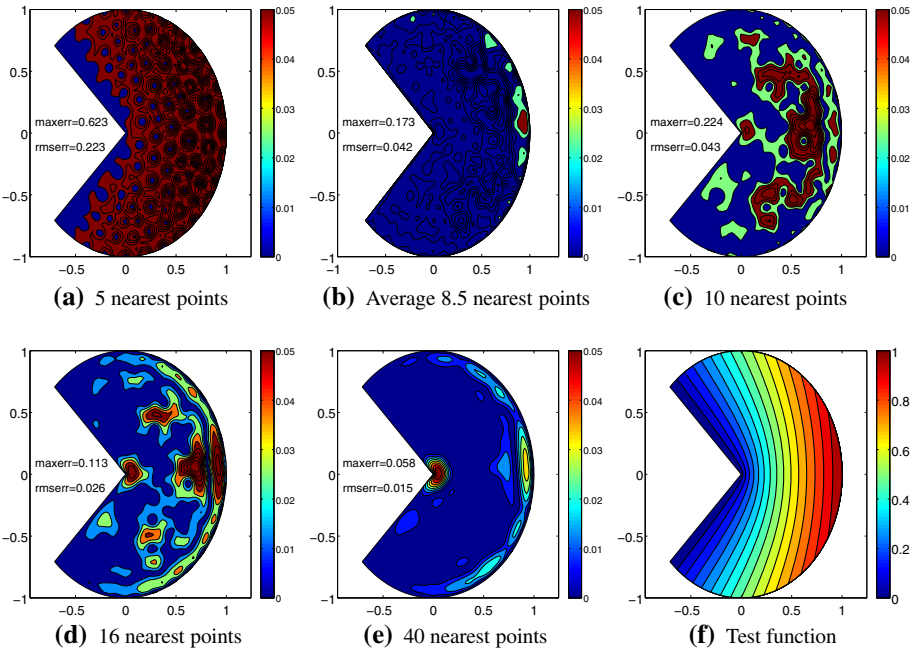
For local refinements, the value R_j for $\mathcal{N}(\mathbf{x}_j, n_j)$ can vary a lot (see Fig. 4b for an illustration). The ratio of the largest support to the smallest support can be on a order of hundred, for example, for the mesh in Fig. 5b. Table 2 shows that the range of the value of R_j can vary by up to hundreds, while the mean of n_j may have a pattern.

4.4 Accuracy

It is known that *stationary interpolation* with compactly supported positive definite kernels does not provide uniform convergence, because it violates the Strang-Fix condition [29,31]. The Strang-Fix condition is a necessary and sufficient condition for stationary interpolation with compactly supported function to achieve uniform local L_p convergence [24, Thm 1]. (Stationary interpolation keeps the support size propositional to the mesh size; on regular mesh this corresponds to the basis function covers a fixed stencil). Making each basis function cover as many as the neighbouring points will do better than using a smaller support. Keeping the diagonal dominance leads to efficiency but also leads to a smaller support than usual cases when each basis function covers more than 25 to 50 nearest neighbouring points. The approximation quality would not be high. See an example in Fig. 6.

5 Algorithm and Implementation

One might think that it is cumbersome to find a set of nearest neighbouring points satisfying the local geometric property described above. In fact a very brief code can achieve the aim: one possible Matlab function to find the largest n_j satisfying Theorem 2 with a set of k ($k \ge n_j$) nearest points is given in Algorithm 1. Together with the k nearest neighbouring points searching algorithm (eg. `knnsearch` in the Matlab statistics toolbox), a



(a) 5 nearest points **(b)** Average 8.5 nearest points **(c)** 10 nearest points

(d) 16 nearest points **(e)** 40 nearest points **(f)** Test function

Fig. 6 Error functions on an 85×85 mesh on polar coordinates and the test function (**f**). **b** Uses the diagonal dominant condition for each basis center, other case use k nearest neighbouring points for each center. Where `maxerr` and `rmserr` are the infinity norm and the root-mean-square error. **a** 5 Nearest points, **b** average 8.5 nearest points, **c** 10 nearest points, **d** 16 nearest points, **e** 40 nearest points, **f** Test function

program to generate the interpolation matrix with differently scaled RBFs can be written as in Algorithm 2.

Note that the `cumsum` takes k operations, calculating `mj` and `qj` takes k and $3k$ operations, calculating `RHS` takes $4k$ operations plus k evaluations; the `find` steps only need $2k$ comparisons and k logical operations. Suppose k is a number less than 100, the total operation count is at most $\mathcal{O}(k^2)$. If an initial set of k neighbouring points is supplied for each point, then the total operation count to calculate the entries for the interpolation matrix is at most $\mathcal{O}(Nk^2)$. Further, for each point, this procedure can run independently which is suitable for parallel computing(by replacing the serial `for` loop in Algorithm 2). The set up of the kd-tree structure takes takes $\mathcal{O}(dN \log(N))$ time [28, p. 240], and finding the k nearest neighbouring points can be done in $\mathcal{O}((\log N + k)N)$ time [28, p. 248] by a generalized kd-tree or bd-tree search. Such causal analysis shows that when N is large, the timing for generating the interpolation matrix is shorter than the time for nearest neighbourhood search. A comparison of the time for the `knnsearch` (knn) and that for checking the diagonal dominance condition is listed in Table 3.

6 Numerical Verification

Two data sets from adaptive finite element refinement on three quarters of a disc are used as the first examples to verify the theory; the data sets are referred to as `TriDisc1` (T1) and `TriDisc2` (T2) respectively. We then apply the procedures to implicit curve/surface

Algorithm 1 A Matlab function to check Theorem 2

```
function[dsto,mnj,G]=checkt(dst,q,rbf)
%% A Matlab function to find a neighbourhood satisfying theorem
%IN: dst:distance of k nearest neighbouring points in increasing order
%     rbf: a function handle
%       q: the lower bound of the convex interval of rbf
%OUT:dsto: the elements of the interpolation matrix within current center
%     mnj: the largest number of nj satisfying the theorem
%       G: G=[r_j, R_j, q_j, mj, lower bounds of eigenvalue]
k=length(dst)-1;              % the number of neighbours
dst2=dst(2:end);              % remove the center itself;
cs=cumsum(dst2);              % cumsum
n1j=[1:k]';                   % column vector of nj
mj=cs./(n1j.*dst2);           % mj as a vector of k
qj=dst2(1)./dst2;             % qj as a vector of k
RHS=1-(1-qj)./(n1j.*rbf(qj));
mask=find(mj>RHS &qj>q);      % satisfy mj> RHS and qj>q
if ~isempty(mask)
     mnj=max(mask);           % find the maximum nj
else
     mnj=2;                   % otherwise mnj=2
end
dsto=[1; rbf(dst2(1:mnj)/dst2(mnj))];
G=[dst2(1), dst2(idx),qj(mnj), mj(mnj), 1-sum(dsto(2:end))];
```

Algorithm 2 A Matlab function to generate the interpolation matrix

```
function[A,R,mnj]=mkA(dsites,rbf,npts,q)
% IN    :   dsites: the interpolation points
%           rbf: the radial basis functions
%           npts: the initial number of neighbour points
%           q: the lower bound of the convex interval of rbf
% OUT   :   A:  the interpolation matrix
%           R:  The radials of each basis functions
%           nj: the numbers of neighbouring points of the jth basis
%                 function
% Requires package: statistics toolbox:knnsearch
NN=length(dsites);
[idx,dist]=knnsearch(dsites,dsites,'k',npts,'NSMethod','kdtree');
idx=idx';dist=dist';
 A=spalloc(NN,NN,NN*npts);s=zeros(NN,5);mnj=uint32(ones(NN,1));
 for k=1:NN
      [dst,mnj(k),G]=checkt(dist(:,k),q,rbf);
      R(k)=G(2);A(idx(1:mnj(k)+1,k),k)=dst;
 end
```

construction problems, demonstrating that the presented algorithms can solve very large scale real applications. For the 3D surface construction problem, we consider the Stanford bunny model and Stanford dragon model in the Stanford3D scanning repository.[1] The Stanford bunny (`bunny.tar.gz`) consists of 35947, 8171, 1889 and 453 points. The Stanford dragon (`dragon_recon.tar.gz`) is a larger model; only three low resolution data sets are used, the data sets with 100250, 22998 and 5205 vertices. The test problems are named as `Bunny453` (B4), `Bunny1889` (B3), `Bunny8171` (B2), `Bunny35947` (B1),

[1] http://graphics.stanford.edu/data/3Dscanrep/.

Table 3 Timing results (in seconds) for C/C++ implementation

Prob.	N	knn	Check	Sparse	ilu	gmres	R1 (‰)	R2 (‰)
B4	1,351	0.0130	0.0002	0.0015	0.0007	0.0054	3.28	1.05
B3	5,643	0.0543	0.0009	0.0068	0.0031	0.0169	4.50	1.21
B2	24,425	0.2576	0.0040	0.0342	0.0162	0.0726	4.50	1.15
B1	105,615	1.2357	0.0180	0.1546	0.0801	0.4029	3.73	1.05
D3	15,563	0.1641	0.0027	0.0213	0.0087	0.0543	4.29	1.19
D2	68,830	0.7776	0.0119	0.1023	0.0495	0.2118	4.55	1.15
D1	300,298	3.5881	0.0521	0.5041	0.2335	0.9574	4.37	1.09

The results illustrate that checking the diagonal dominant condition takes little time in the whole process of scattered data approximation. Column knn is the data querying time by `knnsearch` in Matlab; column sparse is the time for generating sparse matrices by `sparse` in Matlab, this process is not necessary in a pure C/C++ environment; column ilu and column gmres is the time for linear solver; column check is the time for the C/C++ implementation to check the diagonal dominance condition; column R1 is the ratio of the time for check over the time for linear solver; column R2 is the ratio of the time for check over the time for linear solver and data querying

Table 4 Error for the test function with increasing support

Error	Methods					
	I	II				
	8.5	5	10	16	32	40
MAX error	0.173	0.623	0.224	0.113	0.059	0.058
RMS-error	0.042	0.223	0.043	0.020	0.016	0.015

`Dragon5205` (D3), `Dragon22998` (D2) and `Dragon-100250` (D1) respectively. To illustrate the method for 3D surface construction with radial basis functions, a 2D parametric heart curve (HC) is also considered. In all of the problems, the Wendland function $\phi_{3,1} = (1-r)_+^4(4r+1)$ is employed. A collection of the results related to our theorems are put together in Table 2, more information is detailed as follows.

6.1 `TriDisc` Problem

The data set `TriDisc1` is generated by the following Matlab code

```
[~,p,e,t]=adaptmesh('cirsg','cirsb',1,0,0,'maxt',100,...
        'tripick','pdeadworst','ngen',inf);TriDisc1=p';
```

The data set `TriDisc2` is generated similarly by replacing the parameter 100 by 3,000. Figure 5 illustrates the data sets, the structure of the interpolation matrices, and the eigenvalue distribution of the two interpolation matrices.

We interpolate the following test function

$$\left(x^2+y^2\right)^{1/3}\cos\left(\frac{2\,\mathrm{atan2}(y,x)}{3}\right) \tag{19}$$

on the data set `TriDisc1` with two approaches and investigate the error behavior. One approach is to use the diagonally dominant condition described here, which results in an interpolation matrix with average non-zero elements 8.5 per column. In the other approach,

we just use make each basis function cover its k-nearest points, where $k = 5, 10, 16, 32, 40$. The error function is evaluation on $N = 85 \times 85$ points in the domain. Both the infinity norm and the root-mean-square error (RMS-error) are listed in Table 4. The root-mean-square error is defined as

$$\text{RMS-error} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} |s(\mathbf{x}_i) - f(\mathbf{x}_i)|^2}.$$

For the case the case when more than 10 nearest points for each basis function are used in Table 4, the underlying interpolation matrix is not diagonally dominant and could be singular. The method which keeps the diagonal dominance performance slightly better than the case when using 10 nearest points, but its approximation quality is low.

6.2 Implicit Curve/Surface Reconstruction

Here we consider a realistic problem which allows low resolution schemes: constructing an implicit surface. An implicit surface/curve can be viewed as a zero level set of a function $f(\mathbf{x})$, where $f(\mathbf{x})$ satisfies

$$\begin{cases} f(\mathbf{x}) = 0 & \text{if } \mathbf{x} \text{ on the curve/surface;} \\ f(\mathbf{x}^+) > 0 & \text{if } \mathbf{x}^+ \text{ off the curve/surface (outside);} \\ f(\mathbf{x}^-) < 0 & \text{if } \mathbf{x}^- \text{ off the curve/surface (inside).} \end{cases}$$

Given a point set on the curve/surface and associated normal information, the artificial off-surface points $\mathbf{x}^+$ and $\mathbf{x}^-$ are easily obtained. Such a scheme is popular in reconstructing complicated 3D objects, see [4]. Because $\mathbf{x}^+$ and $\mathbf{x}^-$ are artificial and $f(\mathbf{x}^+)$ and $f(\mathbf{x}^-)$ can be any positive and negative number, and usually the supplied normal directions are approximated, inaccurate or measured with some noise. So a low resolution interpolant is enough and efficiency is of more concern in practice.

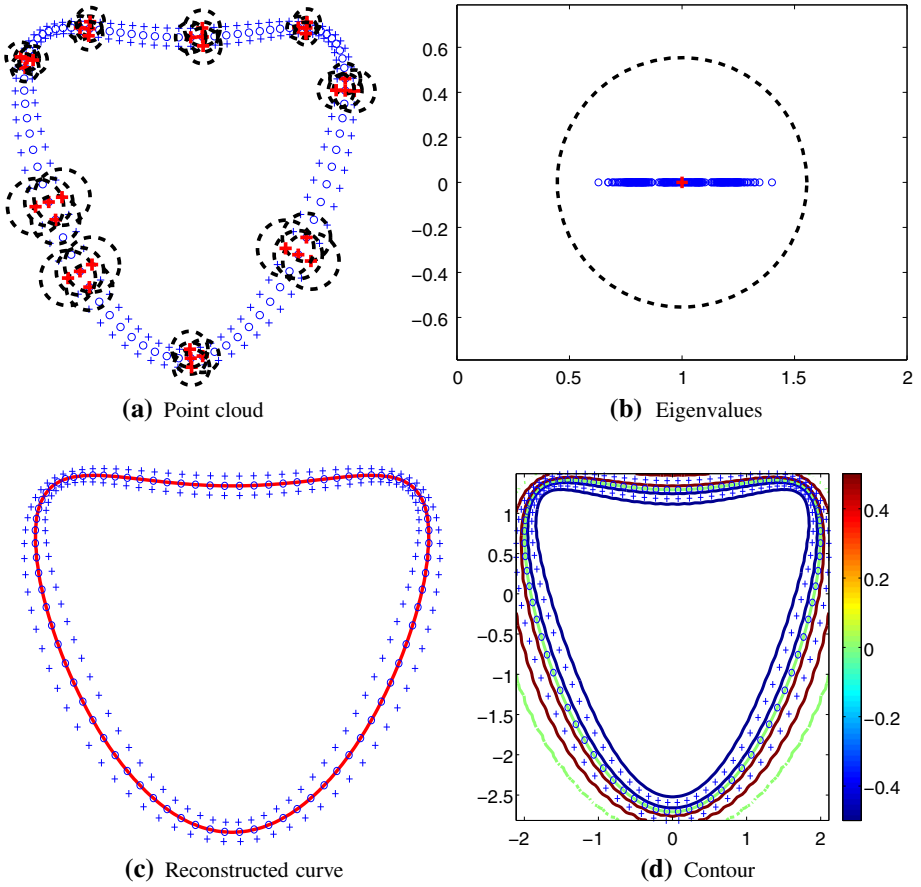
We first illustrate the procedure by an implicit curve construction problem and then show the method which guarantees diagonal dominance condition can solve large scale 3D surface reconstruction problems very efficiently with satisfactory results.

6.2.1 Illustration in 2D

Consider the following heart curve

$$\begin{cases} x(t) = \sqrt{2}(\sin(t) - \cos(t)); \\ y(t) = -\sqrt{2}(\sin(t) + \cos(t)(1 + \sin(t))), \end{cases} \tag{20}$$

which is modified from an example in [8, p. 257]. 84 points on the curve are sampled by setting $\{t_i\}_{i=1}^{84}$ equally spaced in $[0, 2\pi]$. These points themselves are not equally spaced, see the small blue circles in Fig. 7a, c. The off curve artificial points are constructed as follows $\mathbf{x}^{\pm} = \mathbf{x} \pm .75\delta_{\mathbf{x}}\mathbf{n}_{\mathbf{x}}$, where $\mathbf{n}_{\mathbf{x}}$ is the normalised normal direction at $\mathbf{x}$ and $\delta_{\mathbf{x}}$ is the distance from $\mathbf{x}$ to its nearest neighbour on the curve; the ratio .75 can be any scalar not far from 1. Figure 7a illustrates the support of each each basis function, the support is found by Algorithm 1. Figure 7b demonstrates that the eigenvalues of the interpolation matrix are located in the black circle by the diagonal dominance condition. The red curve in Fig. 7c and the zero level set in Fig. 7d is the reconstructed result.



(a) Point cloud **(b)** Eigenvalues

(c) Reconstructed curve **(d)** Contour

Fig. 7 Reconstruction of an implicit curve. The *blue* + *points* are artificial off-curve point constructed by related normal information. The *black circles* in (**a**) are the supports for the basis functions corresponding to *red* + *centres*. **b** Demonstrates that the eigenvalues of the interpolation matrix are located in the *black circle*. The *red curve* in (**c**) and the 0 level set in (**d**) is the reconstructed curve

6.2.2 3D Surface Reconstruction

The normal information are computed by the `normalsply` from the package `ply.tar.gz` provided by Greg Turk.[2] The artificial computed points are constructed by $\mathbf{x}^{\pm} = \mathbf{x} \pm .5\delta_{\mathbf{x}}\mathbf{n}_{\mathbf{x}}$. Suppose `dsites` are the points in the cloud and `normals` is the corresponding calculated normal information. The 3D surface construction problem can be solved by the following framework described in Matlab code Algorithm 3, in which the function name and linear solver are chosen randomly; basically, it includes four steps: preparing data, generating linear systems, solving linear systems and post-precessing. The details of the post-precessing part can be find in [33].

[2] http://www.cc.gatech.edu/projects/large_models/ply.html.

Algorithm 3 A framework for 3D surface reconstruction with Radial basis functions

```
rbf=@(r) (1-r).^4.*(4*r+1);
[dsites,rhs]=scan3Drhs(dsites,normals,.5); % prepare data
[A,R]=mkA(dsites,rbf,npts,.25);  % generate interpolation  matrix
[sol,flag,reves,iter, resvec]=gmres(A,rhs,50,1e$-$13); % A*sol=rhs
%%%%% ---- post-precessing----%%%%
neval=200;[epoints,xe,ye,ze]=eva_pts(dsites,neval);
Pf=spevaluation(dsites,epoints,R,rbf,sol); % sparse evaluation
scan3Dplot(xe,ye,ze,Pf,neval);           %visualiztion
```

Table 5 Information of several test problems

Prob	N	GMRES		ILU(0)-GMRES			
		time	iters	ilu	gmres	total	iters
B4	1,351	0.056699	26	0.000474	0.032375	0.032849	8
B3	5,643	0.137854	29	0.001891	0.043237	0.045128	8
D3	15,563	0.205449	24	0.005438	0.078885	0.084323	8
B2	24,425	0.468109	30	0.008712	0.097339	0.106051	8
D2	68,830	1.431795	30	0.031673	0.235703	0.267376	8
B1	105,615	2.096634	30	0.041696	0.297057	0.338753	7
D1	300,298	6.521828	32	0.160924	1.149587	1.310511	9

6.3 Conditioning

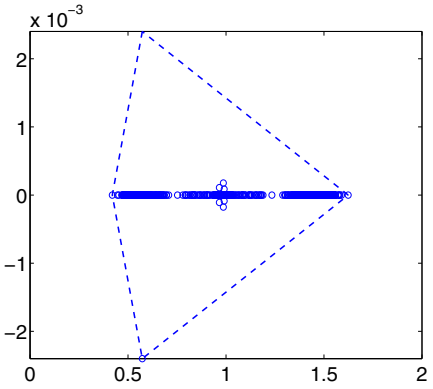
Three ways are used to verify that the interpolation matrix is very well conditioned. First, the lower bounds of the real part of the eigenvalue are calculated when checking the diagonal dominance: $\mathtt{Re}(\lambda) \geq \min_j\{1 - \sum_i a_{ij}\}$. Second, we observe the convergence history of GMRES methods without preconditioning. Third, Matlab `condest` is used for the largest sparse matrices in this paper; which returns a number between 5 to 13. It should be pointed out that the `condest(A,2)` can estimate the condition number to within a factor of 2, whilst `condest(A,4)` which gives a more accurate estimate returns a number around 12.

ILU(0) preconditioning for GMRES is also considered. The performance for the Stanford bunny and Stanford dragon problems are listed in Table 5. The restarted GMRES(50) are used, while all the problem are terminated in 32 iterations without any preconditioning.

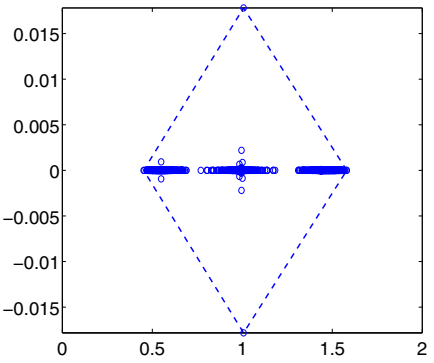
The eigenvalues for two relatively small models are calculated. It turns out that the eigenvalue distribution of the two problem share similar features, all the eigenvalues are clustered in a very small disc with center $(1, 0)$ (see Fig. 8), and the convergence history for GMRES without preconditioning are similar. For large problems, it is not easy to calculate all of the eigenvalues of the interpolation matrix, but the convergence histories for GMRES are similar; see Fig. 8e for the largest problem in this paper. A GMRES solution approach for globally supported radial basis function is discussed in [5].

6.4 Efficiency

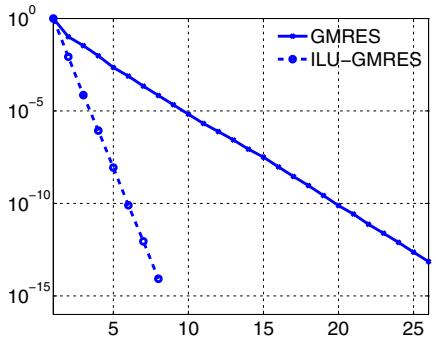
The Matlab implementation in Algorithms 1 and 2 is inefficient for large scale problems due to the long loop in Algorithm 2. Indeed the algorithms here do not guarantee at least 3 elements per column as described in Theorem 3. Procedures to guarantee Theorem 3 would



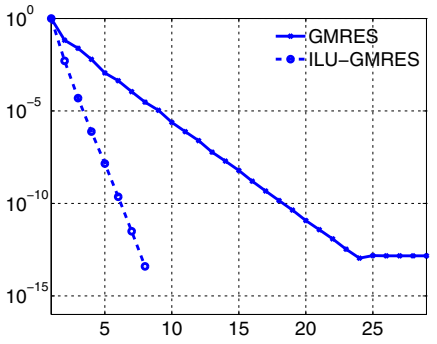
(a) Eigenvalue for Bunny453



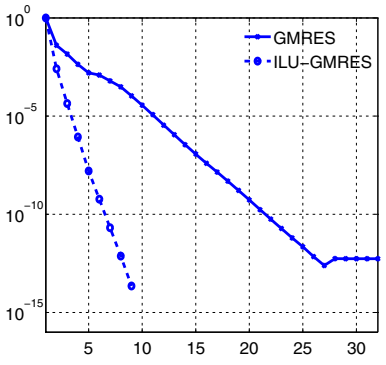
(b) Eigenvalues for Bunny1889



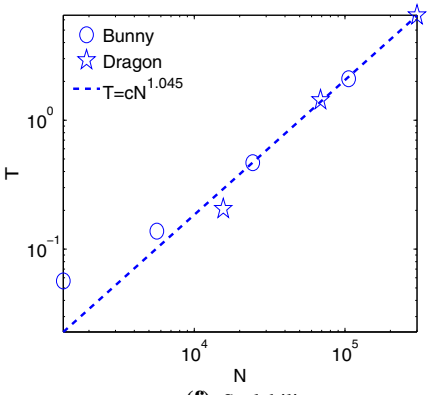
(c) GMRES method for Bunny453



(d) GMRES method for Bunny1889



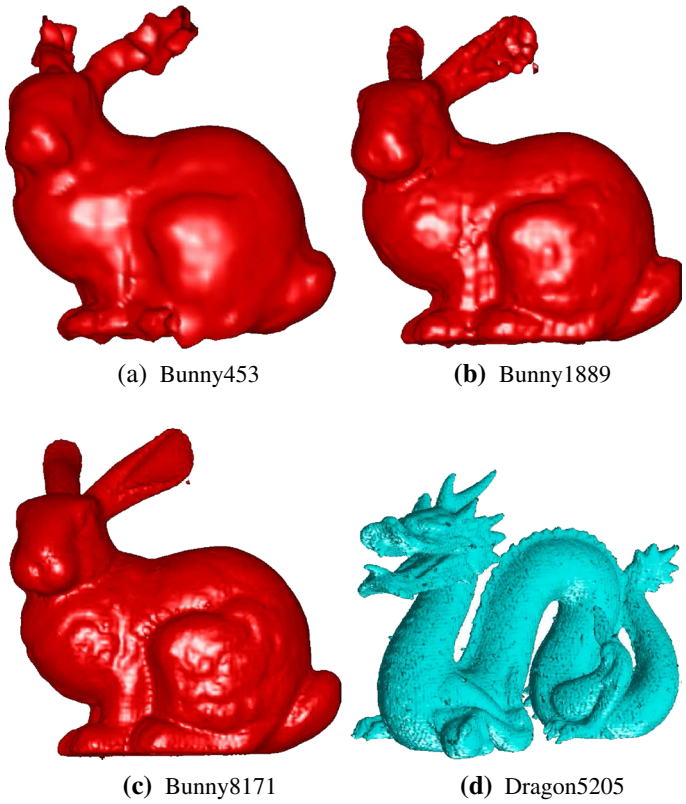
(e) GMRES method for Dragon100250



(f) Scalability

Fig. 8 **a**, **b** The distribution of eigenvalues, the *blue circles* are eigenvalues and the blue polygon are the convex hull of the eigenvalues. **c**–**e** The convergence history of GMRES and preconditioned GMRES methods. **f** The time of solving the interpolation linear systems resulted by Algorithm 2, `mkA` with GMRES solver without any preconditioning is linearly dependent on the dimensional of the linear system. where $c = 1.2168\text{e}{-}05$

require several `if` or `while` branches, which are inefficient in Matlab. If C/C++ is used, an algorithm based on Theorem 3 works efficiently. To illustrate the efficiency, timing results are



(a) Bunny453 **(b)** Bunny1889

(c) Bunny8171 **(d)** Dragon5205

Fig. 9 Surface reconstruction from 3D Scanning Repository

presented in Table 3, where we report the time for `knnsearch` in Matlab (based on kd-tree) to search for the 20 nearest neighbourhood for each point, the time for the `check` procedure, and the command `sparse(row,col,val)` in Matlab to generate a sparse matrix, and the time for `ilu` and `gmres`. The `check` procedure is a Matlab wrapper function with C/C++ kernel function which checks the diagonal dominance condition in Theorem 3 and prepares `row, col` and `val` for the `sparse` function. If experiments are carried out outside of the Matlab environment, the `sparse` function is not necessary. The results are presented to demonstrate that checking the theorem and generating the interpolation matrix takes a relatively small fraction of the overall time. Details of the C/C++ implementation will be published in an ongoing software package.

Figure 9 shows the reconstruction results for the Stanford bunny model and the Stanford dragon model.

7 Conclusion

The main results establishing solvability of the equations for interpolation with radial basis functions generally employ Fourier Transforms and Bochner's theorem. As such they can only apply to RBFs with identical shape parameter. In this manuscript, we provide some sufficient conditions for invertibility of the interpolation equations for Wendland functions

with differing shape. The same idea can also be applied to other compactly supported basis functions which have a fast decay property, for instance the Wu functions. The idea might be useful for the Gaussian with some modification. For using globally supported radial basis function with different scales the reader is redirected to [2]

Numerical examples show the scheme is very efficient for large scale problems but with a lower accuracy. It can can be used when the efficiency is of great concern and a high resolution scheme is not necessary. Examples are given which demonstrate the utility. Our motivation and further consideration is how to combine such a scheme with a proper local refinement scheme and multi-step strategies, thus an error estimate for such a scheme is crucial for wider applications and need to be further considered.

Acknowledgments We thank the referees for valuable advice and suggestion on presenting the results in a more illustrative way, in particular, for one referee who pointed out the fact in Lemma 3, which simplifies the proof in Theorem 3.

References

1. Bozzini, M., Lenarduzzi, L., Rossini, M., Schaback, R.: Interpolation by basis functions of different scales and shapes. Calcolo **41**(2), 77–87 (2004). doi:10.1007/s10092-004-0085-6
2. Bozzini, M., Lenarduzzi, L., Rossini, M., Schaback, R.: Interpolation with variably scaled kernels. IMA J. Numer. Anal. 1–21 (2014). doi:10.1093/imanum/drt071
3. Bozzini, M., Lenarduzzi, L., Schaback, R.: Adaptive interpolation by scaled multiquadrics. Adv. Comput. Math. **16**(4), 375–387 (2002). doi:10.1023/A:1014584220418
4. Carr, J.C., Beatson, R.K., Cherrie, J.B., Mitchell, T.J., Fright, W.R., McCallum, B.C., Evans, T.R.: Reconstruction and representation of 3d objects with radial basis functions. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, pp. 67–76. ACM (2001)
5. Castrillón-Candás, J.E., Li, J., Eijkhout, V.: A discrete adapted hierarchical basis solver for radial basis function interpolation. BIT **53**(1), 57–86 (2013). doi:10.1007/s10543-012-0397-x
6. Deng, Q., Driscoll, T.A.: A fast treecode for multiquadric interpolation with varying shape parameters. SIAM J. Sci. Comput. **34**(2), A1126–A1140 (2012). doi:10.1137/110836225
7. Dou, F.F., Hon, Y.C.: Kernel-based approximation for Cauchy problem of the time-fractional diffusion equation. Eng. Anal. Bound. Elem. **36**(9), 1344–1352 (2012). doi:10.1016/j.enganabound.2012.03.003
8. Fasshauer, G.E.: Meshfree Approximation Methods with MATLAB, Interdisciplinary Mathematical Sciences, vol. 6. World Scientific Publishing Co., Pte. Ltd., Hackensack, NJ (2007). With 1 CD-ROM. Windows, Macintosh and UNIX
9. Flyer, N., Lehto, E.: Rotational transport on a sphere: local node refinement with radial basis functions. J. Comput. Phys. **229**(6), 1954–1969 (2010). doi:10.1016/j.jcp.2009.11.016
10. Fornberg, B., Zuev, J.: The Runge phenomenon and spatially variable shape parameters in RBF interpolation. Comput. Math. Appl. **54**(3), 379–398 (2007). doi:10.1016/j.camwa.2007.01.028
11. Fowkes, J.M., Gould, N.I.M., Farmer, C.L.: A branch and bound algorithm for the global optimization of Hessian Lipschitz continuous functions. J. Global Optim. **56**(4), 1791–1815 (2013). doi:10.1007/s10898-012-9937-9
12. Fuselier, E.J., Wright, G.B.: A high-order kernel method for diffusion and reaction-diffusion equations on surfaces. J. Sci. Comput. **56**(3), 535–565 (2013). doi:10.1007/s10915-013-9688-x
13. Holdahl, S.R., Hardy, R.L.: Solvability and multiquadric analysis as applied to investigations of vertical crustal movements. Tectonophysics **52**(1–4), 139–155 (1979). doi:10.1016/0040-1951(79)90217-8. URL http://www.sciencedirect.com/science/article/pii/0040195179902178. Recent Crustal Movements
14. Hon, Y.C., Wu, Z.: A quasi-interpolation method for solving stiff ordinary differential equations. Int. J. Numer. Methods Eng. **48**(8), 1187–1197 (2000). doi:10.1002/(SICI)1097-0207(20000720)48:8<1187:AID-NME942>3.0.CO;2-K
15. Jamshidi, A.A., Kirby, M.J.: Skew-radial basis function expansions for empirical modeling. SIAM J. Sci. Comput. **31**(6), 4715–4743 (2009/10). doi:10.1137/08072293X
16. Kansa, E.J.: Multiquadrics–a scattered data approximation scheme with applications to computational fluid-dynamics. II. Solutions to parabolic, hyperbolic and elliptic partial differential equations. Comput. Math. Appl. **19**(8–9), 147–161 (1990). doi:10.1016/0898-1221(90)90271-K

17. Kansa, E.J., Carlson, R.E.: Improved accuracy of multiquadric interpolation using variable shape parameters. Comput. Math. Appl. **24**(12), 99–120 (1992). doi:10.1016/0898-1221(92)90174-G. Advances in the theory and applications of radial basis functions
18. Li, J.: Mixed methods for fourth-order elliptic and parabolic problems using radial basis functions. Adv. Comput. Math. **23**(1–2), 21–30 (2005). doi:10.1007/s10444-004-1807-7
19. Li, M., Wang, Y., Ling, L.: Numerical caputo differentiation by radial basis functions. J. Sci. Comput. 1–16 (2014). doi:10.1007/s10915-014-9857-6
20. Marchandise, E., Piret, C., Remacle, J.F.: CAD and mesh repair with radial basis functions. J. Comput. Phys. **231**(5), 2376–2387 (2012). doi:10.1016/j.jcp.2011.11.033
21. Micchelli, C.A.: Interpolation of scattered data: distance matrices and conditionally positive definite functions. Constr. Approx. **2**(1), 11–22 (1986). doi:10.1007/BF01893414
22. Sarra, S.A., Sturgill, D.: A random variable shape parameter strategy for radial basis function approximation methods. Eng. Anal. Bound. Elem. **33**(11), 1239–1245 (2009). doi:10.1016/j.enganabound.2009.07.003
23. Shankar, V., Wright, G.B., Fogelson, A.L., Kirby, R.M.: A radial basis function finite difference method for the simulation of reaction-diffusion equations on stationary platelets within the augmented forcing method. Int. J. Numer. Meth. Fl. **75**(1), 1–22 (2014). doi:10.1002/fld.3880
24. Strang, G., Fix, G.: A fourier analysis of the finite element variational method. In: Constructive Aspects of Functional Analysis. Springer, New York, pp. 793–840 (1971)
25. Varga, R.S.: Matrix Iterative Analysis, Springer Series in Computational Mathematics, vol. 27, expanded edn. Springer, Berlin (2000). doi:10.1007/978-3-642-05156-2
26. Varga, R.S., Saff, E.B., Mehrmann, V.: Incomplete factorizations of matrices and connections with H-matrices. SIAM J. Numer. Anal. **17**(6), 787–793 (1980). doi:10.1137/0717066
27. Wei, T., Hon, Y.C., Wang, Y.B.: Reconstruction of numerical derivatives from scattered noisy data. Inverse Probl. **21**(2), 657–672 (2005). doi:10.1088/0266-5611/21/2/013
28. Wendland, H.: Scattered Data Approximation, Cambridge Monographs on Applied and Computational Mathematics, vol. 17. Cambridge University Press, Cambridge (2005)
29. Wu, Z.: Compactly supported radial functions and the Strang-Fix condition. Appl. Math. Comput. **84**(2–3), 115–124 (1997). doi:10.1016/S0096-3003(96)00110-5
30. Wu, Z.M.: Compactly supported positive definite radial functions. Adv. Comput. Math. **4**(3), 283–292 (1995). doi:10.1007/BF03177517
31. Wu, Z.M., Liu, J.P.: Generalized Strang-Fix condition for scattered data quasi-interpolation. Adv. Comput. Math. **23**(1–2), 201–214 (2005). doi:10.1007/s10444-004-1832-6
32. Zhou, F., Zhang, J., Sheng, X., Li, G.: Shape variable radial basis function and its application in dual reciprocity boundary face method. Eng. Anal. Bound. Elem. **35**(2), 244–252 (2011). doi:10.1016/j.enganabound.2010.08.009
33. Zhu, S., Wathen, A.J.: Fast Sparse Kernel Summation on Cartesian Grids. Tech. Rep. 1820, The University of Oxford, Oxford (2014). URL http://eprints.maths.ox.ac.uk/1820/