Research Article

On Multiscale RBF Collocation Methods for Solving the Monge–Ampère Equation

Zhiyong Liu and Qiuyan Xu

School of Mathematics and Statistics, Ningxia University, Yinchuan 750021, China

Correspondence should be addressed to Zhiyong Liu; zhiyongliu1983@163.com

Received 11 September 2019; Revised 20 January 2020; Accepted 14 February 2020; Published 19 March 2020

Academic Editor: Mohammad D. Aliyu

Copyright © 2020 Zhiyong Liu and Qiuyan Xu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper considers some multiscale radial basis function collocation methods for solving the two-dimensional Monge–Ampère equation with Dirichlet boundary. We discuss and study the performance of the three kinds of multiscale methods. The first method is the cascadic meshfree method, which was proposed by Liu and He (2013). The second method is the stationary multilevel method, which was proposed by Floater and Iske (1996), and is used to solve the fully nonlinear partial differential equation in the paper for the first time. The third is the hierarchical radial basis function method, which is constructed by employing successive refinement scattered data sets and scaled compactly supported radial basis functions with varying support radii. Compared with the first two methods, the hierarchical radial basis function method can not only solve the present problem on a single level with higher accuracy and lower computational cost but also produce highly sparse nonlinear discrete system. These observations are obtained by taking the direct approach of numerical experimentation.

1. Introduction

The Monge–Ampère equation with Dirichlet boundary is a fully nonlinear partial differential equation, which is given by

$$\det(D^2u(z)) = f(z), \quad \text{in } \Omega, \quad (1)$$

$$u(z) = g(z), \quad \text{in } \partial \Omega, \quad (2)$$

where \( z \) has \( n \) independent variables in a bounded domain \( \Omega \subset \mathbb{R}^d \) and \( D^2u \) is the Hessian of the function \( u \). When restricting it to \( \Omega \subset \mathbb{R}^2 \), we can rewrite the equation as

$$Fu = f, \quad \text{in } \Omega, \quad (3)$$

$$u = g, \quad \text{in } \partial \Omega, \quad (4)$$

with a nonlinear map \( F: \omega \rightarrow u_{xx}u_{yy} - u_{xy}^2 \).

The Monge–Ampère equation is originated in geometric surface theory and has been widely applied in dynamic meteorology, elasticity, geometric optics, image processing, conformal geometry, optimal transport, and others. And the numerical solution of the Monge–Ampère equation has been a subject of increasing interest recently. Refer Brenner et al. [1] and Feng and Neilan [2] study for some finite element methods and Oberman’s study [3] for finite difference methods. In order to design a certain type of numerical method which can avoid the tedious mesh generation and the domain integration and can be capable of dealing with any irregular distribution of nodes, some meshfree collocation methods have been researched by Liu and He [4, 5], Böhmer and Schaback [6], and Rashidinia and collaborators [7–9]. However, we are obtaining highly accurate solutions from severely ill-conditioned algebraic systems and high computational cost when using meshfree approximation. Compactly supported radial basis functions can lead to a very well-conditioned sparse system (see [10–12]) but at the cost of a poor approximation accuracy. This is a trade-off principle. That is to say, small support leads to a well-conditioned system but also poor accuracy, while large support yields excellent accuracy at the price of ill-conditioned system. Consequently, to speed up the meshfree
collocation for the Monge–Ampère equation, we need to design some efficient multiscale radial basis function collocation methods. Undoubtedly, this is the motivation of the paper.

The remainder of this paper is organized as follows. In next section, we introduce several multiscale RBF collocation methods. Section 3 presents some numerical results, where we will compare these multiscale algorithms through direct numerical observation. And Section 4 closes the paper with a concise summary.

2. Multiscale RBF Collocation Methods

Let \( X = \{x_1, \ldots, x_N\} \) be a finite point set in \( \Omega \subseteq \mathbb{R}^2 \). We define a commonly used trial discretization parameter:

\[
h_{X, \Omega} := \sup_{x \in \Omega} \min_{y \in X} \|x - y\|_2^2,
\]

which can be regarded as the radius of the largest empty ball that can be placed among the data sites \( x_j \). Generate nested point sets

\[
V_j = \operatorname{span} \left\{ \Phi_j(x, x_1), \ldots, \Phi_j(x, x_{M_j}), \ldots, \Phi_j(x, x_{M_j}), \ldots, \Phi_j(x, x_M_j) \right\},
\]

\[
U_j = \operatorname{span} \left\{ \Phi_j(x, x_1), \ldots, \Phi_j(x, x_{M_j}), \ldots, \Phi_j(x, x_{M_j}), \ldots, \Phi_j(x, x_M_j) \right\}.
\]

Here, we have written \( X^*_j = \{x_1^j, \ldots, x_{M_j}^j\} \) for \( k = 1, 2, \ldots, j \). \( V_j \) and \( U_j \) have multiscale expressions, which differ from the traditional radial basis trial spaces

\[
Q_j = \operatorname{span} \left\{ \Phi(x, x_1^j), \ldots, \Phi(x, x_{M_j}^j), \ldots, \Phi(x, x_1^j), \ldots, \Phi(x, x_{M_j}^j) \right\}.
\]

Let \( M = \sum_{k=1}^N M_k \) be the total number of the centers and \( \{B(x, x_k)\}_{k=1}^M \) be basis functions from one of the trial spaces above. Then, with a trial function \( v(x) = \sum_{k=1}^M c_k B(x, x_k) \), the nonsymmetric kernel-based collocation method for equations (3) and (4) will generate the following nonlinear system

\[
Fv(y) = f(y), \quad y \in Y^I,
\]

\[
v(y) = g(y), \quad y \in Y^B,
\]

for some given interior collocation sites \( Y^I \) and boundary collocation sites \( Y^B \) and some unknown coefficients \( c_k \). We need to compute the derivatives of \( B(x, x_k) \) because the nonlinear operator \( F \) acts on the basis functions now. The derivatives can be obtained by the chain rule from Table 1.

For example, the Wendland’s \( C^6 \) function \( \phi(r) = (1 - er)^6 (32(6er)^2 + 25(6er)^2 + 8er + 1) \) with scaling parameter \( \varepsilon \) has second-order partial derivatives:

\[
\frac{\partial^2}{\partial x^2} \phi(r) = 22\varepsilon^2 (1 - er)^6 (24\varepsilon^2 x^2 (6er + 1)
\]

\[
+ \left( 16(6er)^3 - 9(6er)^2 - 6er - 1 \right),
\]

\[
\frac{\partial^2}{\partial y^2} \phi(r) = 22\varepsilon^2 (1 - er)^6 (24\varepsilon^2 y^2 (6er + 1)
\]

\[
+ \left( 16(6er)^3 - 9(6er)^2 - 6er - 1 \right),
\]

\[
\frac{\partial^2}{\partial x \partial y} \phi(r) = 528\varepsilon^4 xy (1 - er)^6 (6er + 1).
\]

2.1. Cascadic MeshFree Method. Once a convex solution be chosen, equations (3) and (4) can be rewritten in the equivalent form:

\[
X_1 \subset X_2 \subset \cdots \subset X_j \subset \cdots,
\]

which have trial discretization parameters \( h_j = h_{X,j} \). Let \( X^*_j \) be newly added point set in \( X_{j-1} \) then we have \( X_1 = X^*_1 \) and \( X_j = X_{j-1} \cup X^*_j \) for \( j = 2, \ldots, \).

Given a kind of the compactly supported radial functions \( \Phi: \mathbb{R}^2 \rightarrow \mathbb{R} \), we can rescale it (in the translation-invariant kernel-based case) as

\[
\Phi_j(\cdot, y) = \varepsilon_j^d \Phi(\varepsilon_j (\cdot - y)), \quad y \in X^*_j \quad \text{or} \quad y \in X_j.
\]

In order to make the support radii of \( \Phi_j \) become smaller and smaller with the addition of new points, we will select

\[
\varepsilon_j = b \frac{1}{h_j}, \quad \text{with a given constant } 0 < b \leq 1.
\]

Let \( V_j = V_{j-1} + W_j \) with \( V_j = W_j \) and \( W_j = \operatorname{span} \{ \Phi_j(\cdot, y) : y \in X^*_j \} \) for \( j = 1, \ldots \). Then, we can build two types of trial spaces with the form

\[
\Phi_j(x, x_1^j), \ldots, \Phi_j(x, x_{M_j}^j), \ldots, \Phi_j(x, x_1^j), \ldots, \Phi_j(x, x_{M_j}^j) \].
\]

11

12
very difficult to construct the approximation of
Benamou et al. [13] have considered an iterative finite
even though the nonlinear terms are moved to the right side.
find that Algorithm 1 avoids tedious interpolation and makes
in [14]. The cascadic meshfree method is as follows.
based on kernel-based meshfree discretization was proposed
in [14]. The cascadic meshfree method was proposed in [14].
the cascadic meshfree method requires neither
compute. Compared with the traditional cascadic multigrid
method [15], the cascadic meshfree method requires neither
domain nor boundary discretization; i.e., it is truly meshfree.
It is flexible to deal with physical domain with curved
boundary.

2.2. Stationary Multilevel Method. With a stationary mul-
tiscale algorithm [16], the condition number of the discrete matrix can be relatively small, and the computation can be performed in \(O(N)\) operations.

In this algorithm, the present problems (3) and (4) are solved first on the coarsest level by one of the compactly supported radial basis functions with a larger support (usually scaling the size of the support with the fill distance). Then, the residual can be formed and be computed on the next finer level by the same compactly supported radial basis function but with a smaller support. This process can be repeated and be stopped on the finest level. And the final numerical solution is the sum of all of approximation. For interpolation problems, the linear convergence order has been proved in Sobolev spaces on the sphere in [17] and on bounded domains in [18]. And the applications of this algorithm in solving linear partial differential equations on spheres were proposed in [19] and on bounded domains were proposed in [20–22]. A detailed discussion on the stationary multilevel algorithm is reviewed in the latest papers [23, 24]. Algorithm 2 is the specific form for non-
linear situation.

2.3. Hierarchical Radial Basis Function Method. In this subsection, we consider a very simple algorithm for solving Monge–Amphère equations (3) and (4). If we choose to use the trial spaces \(V_j\) to construct kernel-based collocation, then we will obtain a single-level Algorithm 3.

Algorithm 3 is called the hierarchical radial basis function method because it is similar to multilevel splitting of finite element spaces [25]. In fact, \(V_j\) are obtained by splitting the reproducing Hilbert space (or native space of the corresponding strictly positive definite radial basis function). We consider the Matlab implementation of Algorithm 3. Let the radial basis function matrix be \(B_j\) on the \(j\) level, which has the form

\[
\begin{bmatrix}
\Phi_1(y_1, x_1^j) & \Phi_1(y_1, x_1^j) & \ldots & \Phi_j(y_1, x_1^M_j) \\
\Phi_1(y_1, x_1^j) & \Phi_1(y_1, x_1^j) & \ldots & \Phi_j(y_2, x_1^M_j) \\
\vdots & \vdots & \ddots & \vdots \\
\Phi_1(y_1, x_1^j) & \Phi_1(y_1, x_1^j) & \ldots & \Phi_j(y_N, x_1^M_j) \\
\vdots & \vdots & \ddots & \vdots \\
\Phi_1(y_1, x_1^j) & \Phi_1(y_1, x_1^j) & \ldots & \Phi_j(y_{N+j}, x_1^M_j)
\end{bmatrix}
\]

(14)

Here, \(y_1, \ldots, y_{N_j}\) are interior collocation data and \(y_{N_j+1}, \ldots, y_N\) are boundary collocation data in \(\tilde{\Omega} \subset \mathbb{R}^2\). Let \((B_j)_{xx}, (B_j)_{yy}\), and \((B_j)_{xy}\) be some matrices formed by the derivatives of the corresponding functions in \(B_j\), respectively. Then, the approximate values of \(u, u_{xx}, u_{yy}\), and \(u_{xy}\) on the data \(y_1, \ldots, y_N\) can be represented by the column vectors as

\[
\begin{align*}
\bar{u} & = B_j \zeta, \\
\bar{u}_{xx} & = (B_j)_{xx} \zeta, \\
\bar{u}_{yy} & = (B_j)_{yy} \zeta, \\
\bar{u}_{xy} & = (B_j)_{xy} \zeta,
\end{align*}
\]

(15)

with \(\zeta = [\zeta_1, \ldots, \zeta_M]^T\), respectively. And the nonlinear discrete system in Algorithm 3 can be written as

\[
\begin{align*}
\bar{u}_{xx} & - \bar{u}_{yy} f(y_k) = 0, \quad \text{for } k = 1, \ldots, N_j, \\
\bar{u}_{xy} & - g(y_k) = 0, \quad \text{for } k = N_j + 1, \ldots, N.
\end{align*}
\]

(16)

In the next numerical experiments, the linear discrete system will be solved by the least square method, which is a “\(\backslash\)” operation in Matlab. And the Matlab function lsqnonlin

\[
\begin{array}{c}
\begin{array}{c}
\text{Table 1: Derivatives of } \phi(r) = \phi(\|x\|) \text{ with variables } x = (x, y) \in \mathbb{R}^2. \\
\end{array}
\end{array}
\]

\[\begin{array}{c|c}
\text{Partial derivatives} & \frac{\partial}{\partial x} \phi(r) = ((x/r)(d/dr)\phi(r) \\
\frac{\partial}{\partial y} \phi(r) = ((y/r)(d/dr)\phi(r) \\
\frac{\partial}{\partial n} \phi(r) = (((x, n)/r)(d/dr)\phi(r)
\end{array}\]

\[\begin{array}{c|c}
\text{Second-order} & \frac{\partial^2}{\partial x^2} \phi(r) = ((x^2/r^2)((d^2/dr^2)\phi(r) \\
\frac{\partial^2}{\partial y^2} \phi(r) = ((y^2/r^2)((d^2/dr^2)\phi(r) \\
\frac{\partial^2}{\partial x \partial y} \phi(r) = ((xy/r^2)((d^2/dr^2)\phi(r) \\
\frac{\partial^2}{\partial y \partial x} \phi(r) = ((yx/r^2)((d^2/dr^2)\phi(r)
\end{array}\]

\[
\Delta u = \sqrt{2} f(x, y) + u_{xx}^2 + u_{yy}^2 + 2u_{xy}^2, \quad \text{in } \Omega,
\]

\[
u = g(x, y), \quad \text{in } \partial \Omega.
\]

(13)
will be used for solving nonlinear discrete system. Computations were performed on a laptop with 2.4GHz Intel Core i7 processor, using Matlab running in the Windows 7 operating system.

### 3. Numerical Examples

In this section, we discuss the implementation of Algorithms 1–3. We consider the multiscale collocation methods for Monge–Ampère equations (3) and (4) with the true solution:

$$u = \exp \left( \frac{x^2 + y^2}{2} \right).$$

Before the nonlinear solver starts working in Algorithms 2 and 3, a reasonable choice for the initial data is solving the Laplace equation $\Delta u = \sqrt{2} f$. This method of selecting initial values is natural, which has been used in [13, 14].
3.1. Square Domain. First, we generate a nested Halton points 9, 25, 81, 289, and 1089 in the interior of the regular domain $\Omega = [0, 1]^2$ and create additional $4(\sqrt{N} - 1)$ equally spaced collocation points for the boundary conditions on each level. So, the boundary data are $8, 16, 32, 64,$ and $128$ successively. In our experiments, Wendland’s $C^6$ functions carried the parameters $\epsilon_j$ used to construct trial spaces $V_j$ and $U_j$. $M = 40 \times 40$ equally spaced evaluation points are used to compute

$$\text{RMS} - \text{error} = \sqrt{\frac{1}{M} \sum_{k=1}^{M} \left[ u(\xi_k) - u_j(\xi_k) \right]^2} = \frac{1}{\sqrt{M}} \| u - u_j \|_2^2.$$  

(18)

Table 2 shows three kinds of strategies for choosing trial data and testing data. A detailed explanation is as follows:

TT1: trial data and testing data are equal
TT2: trial data and testing data are equal only on boundary, while additional collocation points are created in the internal domain
TT3: trial data only include interior points, while collocation is implemented in internal domain and the boundary

Let $\epsilon_j = 2^{j-1} \epsilon$ be scaled parameter defined by (8) on the $j$-level. Then, for different values of $\epsilon$, the numerical results (including RMS errors and convergence rates) are displayed in Tables 3 to 5. It should be noted that Algorithms 1 and 2 are terrible and nonconvergent if the centers just being placed in the interior of the domain. This is the reason for the lack of corresponding TT3 experiments in Tables 3 to 4. But in Algorithm 3, we let the centers only include Halton data while the collocation sites contain the centers and additional data on the boundary. This is because the boundary data are not nested.

By TT1 strategy and with an initial value $\epsilon = 0.25$, the cascadic meshfree method is convergent although it seems that the convergence ceases at a later stage. But when implementing the algorithm by TT2 strategy with the same initial value, the cascadic meshfree method is non-convergent. This is not contradictory to the linear and nonlinear theory of Schaback [26] and Böhmmer and Schaback [6, 27]. The theory holds that a testing with many more degrees of freedom is necessary because the small trial space must be tested on a fine-grained space discretization. But TT2 strategy does not bring a good approximation on the beginning level. This makes $\sqrt{2f + (u_1)^2_{xx} + (u_1)^2_{yy} + 2(u_1)^2_{xy}}$ be no longer an ideal right side on the next level.

The same set of experiments as for the cascadic meshfree method is displayed in Table 4 for the stationary multilevel RBF collocation method. Different from the numerical results of the stationary multilevel method for solving linear elliptic problems (see Table 41.4 in Fasshauer’s book [28]), Algorithm 2 with an initial $\epsilon = 0.25$ is nonconvergent. It seems that the TT2 strategy is more numerically stable than the TT1 strategy, but with limited accuracy. Maybe a more reasonable parameter scheme should not be $\epsilon_j = 2^{j-1} \epsilon$, when using the stationary multilevel RBF collocation method to

| Table 3: Cascadic meshfree method (square domain). |
|-----------------|-----------------|-----------------|
| $\xi = 0.25$    | Rate            | $\epsilon = 0.25$ | Rate |
| 6.093110e-02   | 2.022638        | 3.075884e-01     |
| 1.499562e-02   | 1.203669        | 6.131756e-01     |
| 6.510642e-03   | -0.472618       | 1.470053e+00     |
| 9.034329e-03   | 1.876969e+00    | -0.995299        |

| Table 4: Stationary multilevel RBF collocation (square domain). |
|-----------------|-----------------|-----------------|
| $\xi = 0.25$    | Rate            | $\epsilon = 0.25$ | Rate |
| 1.768032e-02   | -1.558144       | 8.699399e-01     |
| 5.155207e-02   | -1.881363       | 2.448284e-01     |
| 1.910740e-01   | -0.634312       | 1.454338e-01     |
| 2.965791e-01   | 1.536339e-01    | 0.751409         |

| Table 5: Hierarchical radial basis function method (square domain). |
|-----------------|-----------------|-----------------|
| $\xi = 0.5$     | Rate            | $\epsilon = 0.25$ | Rate |
| 2.187937e+00   | 1.571578        | 1.967965e+00     |
| 7.361101e-01   | 2.181718        | 8.919338e-02     |
| 1.622483e-01   | 0.851573        | 8.846755e-03     |
| 8.991480e-02   | 2.202111e-04    | 5.328189         |
solve fully nonlinear partial differential equation. On lack of the theoretical guidance, we have not obtained some satisfactory numerical results.

In Table 5, we list RMS errors and convergence rate for the hierarchical radial basis function collocation method. We observe that this method has an ideal convergence behavior. Even to a relatively large initial parameter $\varepsilon = 0.5$, the convergence rate of this method is close to 1.5. With an initial parameter $\varepsilon = 0.25$, the RMS error on 4 level is remarkably reduced to $10^{-4}$. Algorithm 3 is a single-level collocation method but has a multiscale trial space. So there is no relationship between the former numerical solution and the latter one. We do not obtain an ideal numerical result on 1 level where only 9 centers are arranged in the interior of the domain. In addition, these 9 centers are far from the boundary. But this situation has been significantly improved after the addition of the added 16 centers. Refer Figure 1 for the initial scattered centers and its refinement. Clearly, compared with the first two algorithms, Algorithm 3 can solve the model problem with a higher accuracy and lower computational cost.

In Figure 2, we show plots of the RMS errors of the above three algorithms. It shows the exponential decay of the error by the hierarchical radial basis function collocation.
3.2. Sector Domain. In this subsection, we consider numerically solving the problems (3) and (4) in a sector domain (see Figure 3) but with the same true solution as in previous example. We first generate a set of successive Halton points: \( N = 9, 25, 81, 289, 1089 \). We place some equidistant points on the rectangular boundary and the curved boundary, respectively, and make the total number of boundary points best equal to 4\( \sqrt{N} - 1 \). The first \( M = 100 \) interior points are used to compute the RMS error, and the same \( \varepsilon_j^* = 2^{-j} \varepsilon \) is used for \( j \)-level experiments. As in example 1, Wendland’s \( C^6 \) functions carried the parameters \( \varepsilon_j \) used to construct trial spaces \( V_j \) and \( U_j \). The numerical results are displayed in Tables 6–8 and Figure 4.

From numerical results, we observe that TT2 testing is not an ideal strategy for the cascadic meshfree method. And given a relatively small initial \( \varepsilon \) stationary multilevel method will be convergent although it becomes very slow at a later stage. But the performance of hierarchical radial basis function method is different from first two kinds of algorithms. With an initial parameter \( \varepsilon = 0.25 \), the RMS error on the 4-level is reduced to \( 10^{-5} \). Figure 4 clearly shows the superiority of the hierarchical radial basis function method.

![Figure 3: Scattered centers and its refinement in sector domain.](image)

![Figure 4: Errors of Algorithms 1 to 3 for different testing strategies and initial \( \varepsilon \) for sector example.](image)

| Table 6: Cascadic meshfree method (sector domain). |
|---|---|---|---|
| TT1 \( \varepsilon = 0.25 \) | Rate | TT2 \( \varepsilon = 0.25 \) | Rate |
| 1.221624e−02 | 3.3232465e−02 | 1.853373e−03 | 3.791312e−02 | −1.258228 |
| 1.380500e−03 | 0.424910 | 3.077875e−01 | −1.949046 |
| 2.619970e−03 | −0.924307 | 4.410556e−01 | −0.519026 |

| Table 7: Stationary multilevel RBF collocation (sector domain). |
|---|---|---|---|
| TT1 \( \varepsilon = 0.25 \) | Rate | TT2 \( \varepsilon = 0.25 \) | Rate |
| 1.192816e−01 | 4.395848e−01 | 3.150129e−01 | 1.453136 |
| 3.150129e−01 | −1.401039 | 1.605481e−01 | 0.715994 |
| 3.808680e−01 | −0.273880 | 9.773937e−02 | 0.168315 |
| 4.758944e−01 | −0.321350 | 8.697644e−02 | 0.168315 |

| Table 8: Hierarchical radial basis function method (sector domain). |
|---|---|---|---|
| TT3 \( \varepsilon = 0.5 \) | Rate | TT3 \( \varepsilon = 0.25 \) | Rate |
| 1.980441e+00 | 4.540010e−01 | 4.8999895e−01 | 1.935033e−02 | 4.552266 |
| 4.8999895e−01 | 2.014999 | 1.935033e−02 | 4.552266 |
| 9.722933e−02 | 2.333287 | 5.096518e−02 | 0.781260 |
| 3.016228e−02 | 1.688646 | 5.384921e−02 | 0.168315 |

Figure 3: Scattered centers and its refinement in sector domain.

Figure 4: Errors of Algorithms 1 to 3 for different testing strategies and initial \( \varepsilon \) for sector example.
4. Conclusions

The paper considered three kinds of multiscale RBF collocation methods for solving the fully nonlinear Monge–Ampère equation. By a specific example, we found that the hierarchical radial basis function collocation method has higher accuracy and lower computational complexity. However, a convergence proof for the hierarchical radial basis function collocation method is missing. This will depend on some Poincaré-type inequality and Bernstein-type inequality. We will prove the boundedness and coerciveness of nonlinear testing operator. This will depend on some Poincaré-type inequality and Bernstein-type inequality. The detailed discussion will be given in [29].

Appendix

Following are the Matlab codes of the hierarchical radial basis function method for Monge–Ampère equation. The DistanceMatrixCSRBF function and 2D Halton data can be cited from [28]:

(1) % Main routine
(2) rbf = @(e,r) r .^ 8 .*(66 .* spones(r)-...)
(3) 154 .* r + 121 .* r .^ 2-32 .* r .^ 3;
(4) Lrbf = @(e,r) 44 .* e .^ 2 .* r .^ 6 *(84 .* spones(r)-...)
(5) 264 .* r + 267 .* r .^ 2-88 .* r .^ 3;
(6) % exact solution and its Laplacian
(7) u = @(x,y) exp((x .^ 2+y .^ 2)/2);
(8) Fu = @(x,y) (1+x .^ 2+y .^ 2).* exp((x .^ 2+y .^ 2));
(9) K = 4; neval = 40; gridtype = 'h';
(10) N = (2^K+1).^2;
(11) name = sprintf('Data2D_%d%s' ,N,gridtype);
(12) load(name);
(13) xdata = dsites(:,1); ydata = dsites(:,2);
(14) ep = 0.5 .^ 2 .* [ 0:K-1];
(15) % create neval-by-neval equally spaced points
(16) % in the unit square
(17) grid = linspace(0,1,neval); [xe,ye] = meshgrid(grid);
(18) epoints = [xe(:) ye(:)];
(19) % compute exact solution
(20) exact = u(epoints(:,1),epoints(:,2));
(21) N1 = [0 9 25 81 289 1089 4225];
(22) for k = 1:K
(23) N2 = (2 .^ [k]+1).^2;
(24) intdata = [xdata(1:N2) ydata(1:N2)];
(25) % add boundary points
(26) sn = sqrt(N2); bdylin = linspace(0,1,sn);
(27) bdy0 = zeros(sn-1,1); bdy1 = ones(sn-1,1);
(28) bdydata = [bdylin(1:end–1) bdy0;...]
(29) bdy1 bdylin(1:end–1);... 
(30) flipud(bdylin(2:end)) bdy1;...
(31) bdy0 flipud(bdylin(2:end)));
(32) rhs1 = Fu(intdata(:,1),intdata(:,2));
(33) rhs2 = u(bdydata(:,1),bdydata(:,2));
(34) % solve linear system on k–level
(35) c[k] = linh(k,N1,ep,xdata, ydata,...
(36) intdata, bdydata,rhs1,rhs2);
(37) % solve nonlinear system on k–level
(38) coef[k] = lsqnlin(@c)nonfh(c,k,N1,ep, xdata,...
(39) ydata, intdata,bdydata,rhs1,rhs2),c[k]);
(40) % assemble evaluation matrix
(41) for j = 1:k
(42) xctrs[j] = [xdata(N1(j) + 1:N1(j + 1))]....
(43) ydata(N1(j) + 1:N1(j + 1))];
(44) DM_eval = DistanceMatrixCSRBF(epoints,...
(45) xctrs[j],ep(j));
(46) EMK[j] = rbf(ep(j),DM_eval);
(47) end
(48) EM = EMK[1];
(49) if(k > 1)
(50) for j = 2:k
(51) EM = [EM EMK[j]];
(52) end
(53) end
(54) % evaluate RBF approximation
(55) Pf = EM * coef[k];
(56) % compute maximum error on evaluation grid
(57) rms_err = norm(Pf–exact)/neval;
(58) if (k > 1)
(59) rms_rate = log(rms_err_old/rms_err)/log(2);
(60) end
(61) rms_err_old = rms_err;
(62) end
Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper used some Halton datasets and drew lessons from partial codes from Fasshauer's book [28]. The authors are grateful to [28] for its free CD. The research of the first author was partially supported by the Natural Science Foundations of China (no. 11501313), the Natural Science Foundations of Ningxia Province (no. 2019AAC02001), and the Third Batch of Ningxia Youth Talents Supporting Program (no. TJGC2018037). Research of the second author was partially supported by the Natural Science Foundations of Ningxia Province (no. NZ2018AAC003026) and the Fourth Batch of Ningxia Youth Talents Supporting Program (no. TJGC2019012).

References

[1] S. C. Brenner, T. Gudi, M. Neilan, and L.-Y. Sung, "C0 penalty methods for the fully nonlinear Monge-Ampère equation," *Mathematics of Computation*, vol. 80, no. 276, p. 1979, 2011.
[2] X. Feng and M. Neilan, "Mixed finite element methods for the fully nonlinear monge-ampe equation based on the vanishing moment method," *SIAM Journal on Numerical Analysis*, vol. 47, no. 2, pp. 1226–1250, 2009.
[3] A. M. Oberman, "Wide stencil finite difference schemes for the elliptic monge-ampe equations and functions of the eigenvalues of the Hessian," *Discrete and Continuous Dynamical Systems—Series B*, vol. 10, no. 1, pp. 221–238, 2008.
[4] Z. Liu and Y. He, "Solving the elliptic monge-ampere equation by Kansa’s method," *Engineering Analysis with Boundary Elements*, vol. 37, no. 1, pp. 84–88, 2013.
[5] Z. Liu and Y. He, "An iterative meshfree method for the elliptic monge-ampere equation in 2D," *Numerical Methods for Partial Differential Equations*, vol. 30, no. 5, pp. 1507–1517, 2014.
[6] K. Böehmer and R. Schaback, "A meshfree method for solving the Monge-Ampère equation," *Numerical Algorithms*, vol. 82, no. 2, pp. 539–551, 2019.
[7] J. Rashidinia, G. E. Fasshauer, and M. Khasi, "A stable method for the evaluation of Gaussian radial basis function solutions of interpolation and collocation problems," *Computers & Mathematics with Applications*, vol. 72, no. 1, pp. 178–193, 2016.
[8] J. Rashidinia, M. Khasi, and G. E. Fasshauer, "A stable Gaussian radial basis function method for solving nonlinear unsteady convection-diffusion-reaction equations," *Computers & Mathematics with Applications*, vol. 75, no. 5, pp. 1831–1850, 2018.
[9] J. Rashidinia and M. Khasi, "Stable Gaussian radial basis function method for solving Helmholtz equations," *Computational Methods for Differential Equations*, vol. 7, pp. 138–151, 2019.
[10] H. Wendland, "Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree," *Advances in Computational Mathematics*, vol. 4, no. 1, pp. 389–396, 1995.
[11] Z. Wu, “Compactly supported positive definite radial functions,” *Advances in Computational Mathematics*, vol. 4, no. 1, pp. 283–292, 1995.

[12] Q. Xu and Z. Liu, “Scattered data interpolation and approximation with truncated exponential radial basis function,” *Mathematics*, vol. 7, no. 11, p. 1101, 2019.

[13] J.-D. Benamou, B. D. Froese, and A. M. Oberman, “Two numerical methods for the elliptic Monge-Ampère equation,” *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 44, no. 4, pp. 737–758, 2010.

[14] Z. Liu and Y. He, “Cascadic meshfree method for the elliptic Monge-Ampère equation,” *Engineering Analysis with Boundary Elements*, vol. 37, no. 7-8, pp. 990–996, 2013.

[15] F. A. Bornemann and P. Deuflhard, “The cascadic multigrid method for elliptic problems,” *Numerische Mathematik*, vol. 75, no. 2, pp. 135–152, 1996.

[16] M. S. Floater and A. Iske, “Multitstep scattered data interpolation using compactly supported radial basis functions,” *Journal of Computational and Applied Mathematics*, vol. 73, no. 1-2, pp. 65–78, 1996.

[17] Q. T. L. Gia, I. H. Sloan, and H. Wendland, “Multiscale analysis in Sobolev spaces on the sphere,” *SIAM Journal on Numerical Analysis*, vol. 48, no. 6, pp. 2065–2090, 2010.

[18] H. Wendland, “Multiscale analysis in Sobolev spaces on bounded domains,” *Numerische Mathematik*, vol. 116, no. 3, pp. 493–517, 2010.

[19] Q. T. Le Gia, I. H. Sloan, and H. Wendland, “Multiscale RBF collocation for solving PDEs on spheres,” *Numerische Mathematik*, vol. 121, no. 1, pp. 99–125, 2012.

[20] A. Chernih and Q. T. Le Gia, “Multiscale methods with compactly supported radial basis functions for Galerkin approximation of elliptic PDEs,” *IMA Journal of Numerical Analysis*, vol. 34, no. 2, pp. 569–591, 2014.

[21] A. Chernih and Q. T. Le Gia, “Multiscale methods with compactly supported radial basis functions for the Stokes problem on bounded domains,” *Advances in Computational Mathematics*, vol. 42, no. 5, pp. 1187–1208, 2016.

[22] P. Farrell and H. Wendland, “RBF multiscale collocation for second order elliptic boundary value problems,” *SIAM Journal on Numerical Analysis*, vol. 51, no. 4, pp. 2403–2425, 2013.

[23] H. Wendland, “Multiscale radial basis functions,” in *Frames and Other Bases in Abstract and Function Spaces–Novel Methods in Harmonic Analysis*, I. Pesenson, Q. T. L. Gia, A. Mayeli, H. Mhaskar, and D. X. Zhou, Eds., vol. 1, pp. 265–299, Springer, Berlin, Germany, 2017.

[24] H. Wendland, “Solving partial differential equations with multiscale radial basis functions,” in *Contemporary Computational Mathematics–A Celebration of the 80th Birthday of Ian Sloan*, J. Dick, F. Y. Kuo, and H. Wozniakowski, Eds., Springer, Berlin, Germany, pp. 1191–1213, 2018.

[25] H. Yserentant, “On the multi-level splitting of finite element spaces,” *Numerische Mathematik*, vol. 49, no. 4, pp. 379–412, 1986.

[26] R. Schaback, “Unsymmetric meshless methods for operator equations,” *Numerische Mathematik*, vol. 114, no. 4, pp. 629–651, 2010.

[27] K. Böhmer and R. Schaback, “A nonlinear discretization theory,” *Journal of Computational and Applied Mathematics*, vol. 254, pp. 204–219, 2013.

[28] G. E. Fasshauer, *Meshfree Approximation Methods with MATLAB*, World Scientific Publishers, Singapore, 2007.

[29] Z. Liu and Q. Xu, “Convergence of hierarchical radial basis function collocation method,” In preparation.