A modified moving least-squares suitable for scattered data fitting with outliers

Sanpeng Zheng[a], Renzhong Feng[a,*], Aitong Huang[a]

[a]*School of Mathematics and Systematic Science & Key Laboratory of Mathematics, Informatics and Behavioral Semantics, Ministry of Education, Beihang University, Beijing, China*

Abstract

In the fitting of scattered data, there may be a few sample values that contain high noise, which are called outliers. In dealing with such scattered data, the approximation effect of the classical moving least squares (abbr. MLS) is greatly reduced due to the existence of outliers. In this paper, a modified moving least squares (abbr. MMLS) is proposed, which can recognize outliers automatically from scattered data and weaken the influence of the outliers in fitting by an added weight function in MLS. It is theoretically proven that if the only noise existing in scattered data is outliers, the solution of MMLS is close to that of MLS in the absence of outliers. Because the computational process of the proposed MMLS is consistent with the classical MLS, the computational efficiency of MMLS is higher than that of Levin's moving least-hardy method (abbr. MLH) which is proposed to also deal with the fitting of scattered data with outliers by iterative solution. The numerical experiments not only confirm the property of MMLS but also show that the approximation effect of MMLS is almost identical with that of MLH.

Keywords: scattered data, data fitting, outlier, moving least squares, weight function, modified moving least squares
2000 MSC: 65D15, 33F05, 65D10, 65D07

1. Introduction

Moving least squares is one of the most effective methods for the approximation of scattered data. It was proposed by McLain[1] in 1974, then Lancaster

*Corresponding author: fengrz@buaa.edu.cn
Email address: fengrz@buaa.edu.cn (Renzhong Feng)

Preprint submitted to Elsevier

and Salkauskas[2] carried out theoretical analysis on MLS method. For an object function $f:\Omega\to\mathbb{R}$ defined on a given region $\Omega\subseteq\mathbb{R}^d$, we assume that $\mathbf{X}=\{\mathbf{x}_i\}_{i=1}^N$ is a group of scattered points in $\Omega\subseteq\mathbb{R}^d$ and $\mathbf{F}=\{f_i\}_{i=1}^N$ are the data values corresponding to set $\mathbf{X}$. At point $\mathbf{x}\in\Omega$ the m-order moving least squares approximation $p^*(\mathbf{x})$ is the one which minimizes the following weighted l_2 norm in polynomial space Π_m

$$E_{2,\theta,\mathbf{x}}(p)=\sum_{i=1}^{N}(f_i-p(\mathbf{x}_i))^2\theta(\|\mathbf{x}-\mathbf{x_i}\|), \tag{1.1}$$

where θ is a nonnegative weight function, $\|\cdot\|$ is Euclidean distance in $\mathbb{R}^d$ and Π_m is the polynomial space which contains all polynomials whose degrees are less than or equal to m.

If θ decreases rapidly as $r\to\infty$, then this approximation is local. If $\theta(0)=\infty$, then this approximation is interpolation. For every point $\mathbf{x}$, the calculation of MLS approximation involves solving a minimization problem, and the choice of l_2 norm simplifies this minimization problem. If $f_i=f(\mathbf{x}_i), i=1,\dots,N$, $f\in C^{m+1}$ and θ is a compactly supported function, then the approximation order of $m+1$ can be achieved [3][4][5]. Another important property of MLS is that if $\theta\in C^\infty$, then the MLS approximation is also C^∞.

The studies and applications on MLS are emerging endlessly. Li et al [6] studied the complex moving least squares approximation and the associated element-free Galerkin method. Mirzaei et al [7] proposed a generalized moving least squares to gain the diffuse derivatives of an unknown function directly and applied it to solving PDEs on spheres in [8]. Dehghan et al [9] presented an improved MLS for 2D elliptic interface problems. Amirfakhrian et al [10] used the moving least squares method to approximate the parametric curves.

Under normal circumstances, the approximation effect, continuity and computational efficiency of MLS method are excellent. However, if scattered data contain high noise, the approximation of MLS would have a large deviation. We call the high noise values in scattered data outliers.

If the l_2 norm in MLS is changed into l_1 norm, then MLS method becomes the moving least l_1 method (abbr. MLl_1). MLl_1 is insensitive to outliers [11], but can not give a smooth, or even a continuous approximation [12]. In order to solve the problems in MLS, Levin [12] introduced Hardy's multiquadric function and proposed the moving least-Hardy method (abbr. MLH) by combining the thoughts of MLS approximation and MLl_1 approximation. MLH moves the error of each sampling point into Hardy's multiquadric function, then multiplies the weight function

$\theta(\|\mathbf{x}-\mathbf{x}_i\|)$, and finally sums all. Through minimizing the following expression

$$E_{H_d,\theta,\mathbf{x}}(p)=\sum_{i=1}^{N}\sqrt{(f_i-p(\mathbf{x}_i))^2+d^2}\,\theta(\|\mathbf{x}-\mathbf{x_i}\|), \tag{1.2}$$

the approximation function $p^*(\mathbf{x})$ is obtained. It has been proven in [12] that MLH method is insensitive to outliers and can give a solution with first-order approximation accuracy for the scattered data sampled from a smooth function with a small number of outliers. However, it requires an iterative process, which makes its computational efficiency lower than that of the classical MLS method.

This paper proposes a modified moving least squares (abbr. MMLS). MMLS would let each term on the right side of formula (1.1) multiply a new weight function. The new weight function is used to identify the outlier points and weaken the influence of those points on approximation. We prove that if the sampling point set $\mathbf{X}$ is relatively dense, the outlier points are distributed according to our assumption in Section 3 and there is no other noise, the approximation of MMLS would approach the approximation of classical MLS without outliers. In addition, since the MMLS method only needs to obtain the new weight function and solve a system of linear equations at every point $\mathbf{x}\in\Omega$, it avoids the iterative process in MLH. Thus, the computational efficiency of MMLS is higher than that of MLH.

It is a general idea to select outlier points from $\mathbf{X}$. There are some studies on how to find the outliers. For example, Amir and Levin recently proposed a method to find outlier points from $\mathbf{X}$ and remove them in [13]. They proved that this method can find outliers accurately in some specific cases. However, the time complexity of Amir and Levin's method is slightly high. Our method only needs to find out the quasi-outlier points by the weight function, hence it efficiency is high. It is not excluded that there may be individual key points in the quasi-outlier points selected by our method. If these key points do not participate in the least squares approximation, the accuracy of approximation will be affected definitely. For the above reason, we do not remove all the found quasi-outliers, but retained them, and used the new weight function to reduce their impact on the approximation.

In the paper, the construction and theoretical analysis of MMLS method are limited to $\Omega\subseteq\mathbb{R}^2$. Although the idea in the paper can be extended to higher dimension, it requires more complex construction and theoretical analysis. This paper is arranged as follows: Section 2 introduces the basic knowledge required for the paper; Section 3 introduces the modified moving least-squares method; Section 4 is the numerical experiment; Section 5 is the conclusion.

2. Basic knowledge

This section introduces some basic knowledge required later: natural neighbors and an approximate gradient estimation method for smooth bivariate function.

2.1. Natural Neighbors

Assume that Conv(X) is the convex hull of the point set X and h is the fill distance of X in Conv(X). We call $C_i = \{\mathbf{x} \in \mathbb{R}^d | \|\mathbf{x} - \mathbf{x}_i\| \leq \|\mathbf{x} - \mathbf{x}_j\|, \forall j \neq i\}$ the Voronoi element of $\mathbf{x}_i \in \mathbb{R}^d$. The natural neighbors of a node $\mathbf{x}_i$ in $\mathbb{R}^d$ are the nodes in X whose Voronoi element has common edge with the Voronoi element of $\mathbf{x}_i$, in other words, nodes connected to $\mathbf{x}_i$ by a edge of the Delaunay triangle with vertex $\mathbf{x}_i$. Fig.1 shows node A with 5 natural neighbors, and Fig.2 shows the corresponding Delaunay triangulation.

Although natural neighbors usually refer to the nodes, one can equally well define

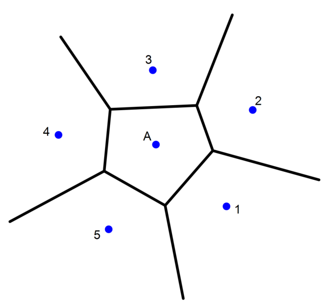

Fig. 1: Node A and its natural neighbors.

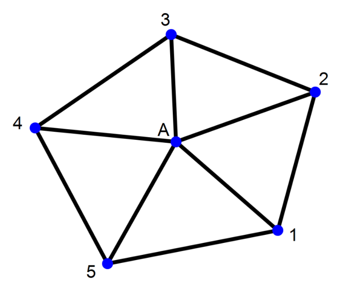

Fig. 2: Delaunay triangulation.

a set of natural neighbors to any point $\mathbf{x} \in \mathbb{R}^d$, as the set of nodes which would be connected to the point if it were added to the Delaunay triangulation. Natural neighbors redefine a set of 'closest surrounding nodes' whose number and position are determined merely by the local nodal distribution. One can regard the natural neighbors about any point as a unique set of nodes that defines the neighbourhood of the point. If the distance between nodes is relatively large in some place, or the distribution is highly anisotropic, then the set of natural neighbors will reflect these features, nevertheless it still denote the best set of nearby surrounding nodes. The method that finds the natural neighbors of any arbitrary point $\mathbf{x} \in \mathbb{R}^d$ in a given point set is referred to [14][15].

2.2. *An approximate gradient estimation method for smooth bivariate function based on scattered data*

Goodman et al [16] proposed a method for calculating the approximate gradient of a smooth bivariate function by the triangle mesh of the scattered node set $\mathbf{X} \subseteq \mathbb{R}^2$. In order to make this method be suitable for the needs of this paper, we modify the method slightly.

We calculate the approximate gradient at node $\mathbf{p}$ in two cases: the first case is that the node $\mathbf{p}$ is inside the triangular mesh, as shown in Fig. 3. For each $i = 1, \ldots, M$, π_i denotes triangle $\triangle_{\mathbf{p}\mathbf{x_i}\mathbf{x_{i+1}}}(\mathbf{x}_{M+1} = \mathbf{x}_1)$ and $\mathbf{g}_i$ denotes the gradient of the linear interpolation on triangle π_i.

We use the convex combination $\mathbf{g}$ of those approximate gradient $\mathbf{g}_1, \ldots, \mathbf{g}_M$ to

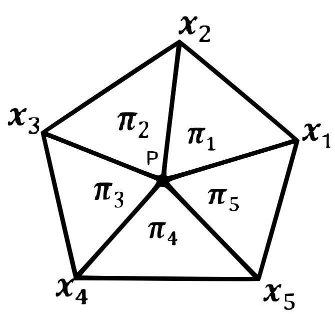

Fig. 3: The inner node $\mathbf{p}$ and the triangles surrounding it.

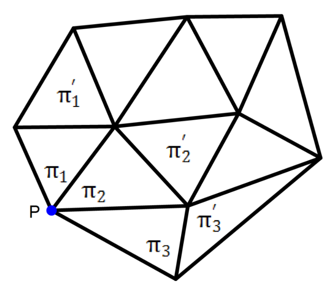

Fig. 4: The node $\mathbf{p}$ is at theboundary of the triangulation.

approximate the gradient of function $f(x,y)$ at point $\mathbf{p}$, $(\frac{\partial f}{\partial x}(\mathbf{p}), \frac{\partial f}{\partial y}(\mathbf{p}))$, i.e.

$$\mathbf{g} = \sum_{i=1}^{M} (\mathbf{g}_i \cdot \eta_i / \sum_{j=1}^{M} \eta_j). \tag{2.1}$$

In (2.1), $\eta_i = S_i, i = 1, \ldots, M$, and S_i is the area of the triangle π_i. The $\mathbf{g}_i$ and S_i are calculated as follows: suppose that π_i has vertex (x_j, y_j) and the corresponding data $z_j, j = 1,2,3$. The linear interpolation of these data is a plane following

$$\alpha x + \beta y + \gamma z + \delta = 0.$$

Here α, β, γ are the components of this plane normal vector, i.e.

$$\begin{aligned} \alpha &= (y_2 - y_1)(z_3 - z_1) - (y_3 - y_1)(z_2 - z_1), \\ \beta &= (x_3 - x_1)(z_2 - z_1) - (x_2 - x_1)(z_3 - z_1), \\ \gamma &= (x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1). \end{aligned} \tag{2.2}$$

Therefore the gradient of this plane is

$$\mathbf{g}_i = (\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}) = (-\frac{\alpha}{\gamma}, -\frac{\beta}{\gamma}). \tag{2.3}$$

The area of the triangle π_i is

$$S_i = \frac{1}{2}|\gamma|. \tag{2.4}$$

The second case is that $\mathbf{p}$ is at the boundary of the triangulation, as shown in Fig. 4. In this case, we need more information around the point. Therefore, we add the triangle π_i' which has common edge with the triangle π_i. $\mathbf{g}_i'$ and S_i' corresponding to π_i' are calculated by (2.3) and (2.4). However, the convex combination (2.1) has been changed into the following form:

$$\mathbf{g} = \left(\sum_{i=1}^{M} S_i(((2S_i + S_i')\mathbf{g}_i - S_i\mathbf{g}_i')/(S_i + S_i')) \right) \Bigg/ \left(\sum_{i=1}^{M} S_i \right). \tag{2.5}$$

3. A modified moving least-squares method

In the case of data containing small noise, the approximation capability of MLS is guaranteed. However, if the scattered data contains outliers, the approximation effect of MLS is very unsatisfactory, referring to [12]. To solve this problem, Levin proposed the MLH method in [12]. Although MLH method has great performance in dealing with scattered date with outliers, its computational efficiency still has room for improvement. Hence, this paper proposes a modified MLS method. MMLS lets each term on the right side of formula (1.1) multiply a new weight function. The new weight function is used to identify the outlier points automatically and weaken the influence of the outliers on approximation. Next, we introduce the construction of the weight function and analyze its properties theoretically. Finally, we propose the modified moving least-squares method.

3.1. The construction and property of the weight function

The construction of this weight function consists of two stages. First of all, we select the points that may be outlier points from $\mathbf{X}$ to constitute a new point set which is called quasi-outlier point set and written as $\mathscr{O}'$. Then, a weight function is constructed, and we apply it to giving each point in $\mathbf{X}$ a weight. The weight is expected to weaken the influence of the outliers in MLS. After accomplishing these two stages, we will discuss the property of the weight function.

3.1.1. The selection of the quasi-outlier points

Suppose the sample value at $\mathbf{x}_i \in \Omega \subseteq \mathbb{R}^2, i = 1, \ldots, N$ is $\widetilde{f}_i = f(\mathbf{x}_i) + e_i$. e_i denotes the noise which may be zero. We use $N(\mathbf{x}_i)$ to denote the natural neighbors of point $\mathbf{x}_i$ in node set $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$, and define $\overline{N(\mathbf{x}_i)} := N(\mathbf{x}_i) \cup \{x_i\}$.
Let $\widetilde{\mathbf{p}}_i$ be the approximate gradient at point $\mathbf{x}_i$, which is calculated by the method in Section 2.2 with the sampled data $\{(\mathbf{x}_j, \widetilde{f}_j) | \mathbf{x}_j \in \overline{N(\mathbf{x}_i)}\}$. Then we construct a function $core(\mathbf{x}_i)$ to select the quasi-outlier points from $\mathbf{X}$. Suppose that $|N(\mathbf{x}_i)| = M_i$, $|N(\mathbf{x}_i)|$ denotes the number of the natural neighbors of $\mathbf{x}_i$, and write $N(\mathbf{x}_i) = \{\mathbf{x}_{i_j}\}_{j=1}^{M_i}$. The function $core(\mathbf{x}_i)$ is defined as

$$core(\mathbf{x}_i) := \sum_{j=1}^{M_i} \left| \mathbf{u}_{i_j} \cdot \widetilde{\mathbf{p}}_i - \frac{\widetilde{f}_{i_j} - \widetilde{f}_i}{\|\overrightarrow{\mathbf{x}_i \mathbf{x}_{i_j}}\|} \right|, \tag{3.1}$$

here, $\mathbf{u}_{i_j}$ stands for the unit direction vector corresponding to $\overrightarrow{\mathbf{x}_i \mathbf{x}_{i_j}}$.
According to (3.1), the function $core(\mathbf{x}_i)$ is the sum of all the absolute values of the differences between two kinds of approximate directional derivatives along M_i directions. It is obvious that the larger $core(\mathbf{x}_i)$ is, the more likely the sample value at $\mathbf{x}_i$ is an outlier. The detailed analysis is in Section 3.1.3. Then the method to select the quasi-outlier point set $\mathscr{O}'$ from $\mathbf{X}$ can be stated as follows:

Algorithm 1 The selection of the quasi-outlier point set $\mathscr{O}'$

Input: input node set $\mathbf{X}$, sample value $\{\widetilde{f}_i\}_{i=1}^N$, and control parameter δ.
Output: output the quasi-outlier point set $\mathscr{O}'$

Compute $core(\mathbf{x}_i)$ for each point in node set $\mathbf{X}$;
Set $\{c_j\}_{j=1}^N = sort(\{core(\mathbf{x}_i)\}_{i=1}^N)$;
Set $k = 1$;
while $k <= N-1$ and $c_{k+1} - c_k <= \delta$ **do**
  $k = k+1$
end while
if $k == N$ **then**
  $c = c_N$;
  $\mathscr{O}' = \emptyset$;
else
  $c = c_k$;
  $C' = \{\mathbf{x}_i | core(\mathbf{x}_i) > c\}$;
  $\mathscr{O}' = \{\mathbf{x}_i | \overline{N(\mathbf{x}_i)} \subseteq C'\}$;
end if
return $\mathscr{O}'$

Remark 3.1. *The members of set C' are the points whose $core(\mathbf{x}_i)$ is larger than c in $\mathbf{X}$. By analyzing formula (3.1), we can find that C' contains not only outlier points but also the natural neighbors of outlier points. Therefore the quasi-outlier point set we need is $\mathscr{O}' = \{\mathbf{x}_i | \overline{N(\mathbf{x}_i)} \subseteq C'\}$, which is an approximation of the unknown outlier point set $\mathscr{O}$.*

3.1.2. The construction of the weight function

After gaining the quasi-outlier point set $\mathscr{O}'$, we regard all points in $X \backslash \mathscr{O}'$ as normal. Thus when we construct the weight function, we only modify the weights of the points in $\mathscr{O}'$, and set the weights of points in $\mathbf{X} \backslash \mathscr{O}'$ to 1. Then the weight function ω is defined as follows

$$\omega(\mathbf{x}_i, \delta) := \begin{cases} \frac{1}{1+core(\mathbf{x}_i)}, & \mathbf{x}_i \in \mathscr{O}', \\ 1, & \mathbf{x}_i \in X \backslash \mathscr{O}'. \end{cases} \tag{3.2}$$

The δ in (3.2) is the parameter which is used to get the quasi-outlier point set $\mathscr{O}'$. Its size is determined by the smoothness of the sampling function and the singularity of the outliers. The method to select this parameter will be given after the analysis in Section 3.1.3.

3.1.3. The asymptotic property of the weight function

Because of the existence of $\widetilde{\mathbf{p}}_i$ and the difference in (3.1), we suppose that the sampling function $f(\mathbf{x})$ is at least $C^2(\Omega)$. Therefore, we define

$$\mathscr{M}(f) := \max_{\substack{\mathbf{x}\in\Omega \\ \|\mathbf{u}\|=1}} \left| \frac{\partial^2 f}{\partial \mathbf{u}^2}(\mathbf{x}) \right|.$$

The h in the paper is the fill distance defined as

$$h = h_{X,\Omega} := \sup_{\mathbf{x}\in\Omega} \min_{1\le i\le N} \|\mathbf{x}-\mathbf{x_i}\|.$$

We define two additional distances

$$h_M := \max_{x_i\in\mathbf{X}} \max_{x_j\in N(x_i)} \|\mathbf{x_j}-\mathbf{x_i}\|,$$

$$h_m := \min_{x_i\in\mathbf{X}} \min_{x_j\in N(x_i)} \|\mathbf{x_j}-\mathbf{x_i}\|.$$

In order to analyze the property of the weight function conveniently, we define two operations on the node set $\mathbf{X}$:

(i) $\overline{N(Y)} := \bigcup_{\mathbf{x}_i\in Y} \overline{N(\mathbf{x}_i)}, \forall\, Y \subseteq X,$

(ii) $\overline{iN(Y)} := \{\mathbf{x}_i \in Y | N(\mathbf{x}_i) \subseteq Y\}, \forall\, Y \subseteq X.$

For the needs of later theoretical analysis, we make five important assumptions.

(1) Suppose that there are few outlier points in $\mathbf{X}$ and they are not adjacent to each other.

(2) Suppose that only the error e_i corresponding to the outlier point is nonzero , namely $\mathscr{O} = \{\mathbf{x}_i \in \mathbf{X} \big| |e_i| > 0\}$.

(3) There exists a constant $1 \le K < 2$, such that the three distances satisfy that $h \ge \frac{1}{2}h_m$ and $h_M \le K h_m$.

(4) There exists a constant $\mathscr{E} > 0$, such that for any point $\mathbf{x}_l \in \mathscr{O}$, there is $|e_l| \ge \mathscr{E}$.

(5) $\Omega \subseteq \mathbb{R}^2$ is compact and satisfies the interior cone condition in [5].

Remark 3.2. *The sentence "they are not adjacent to each other" in assumption (1) means that there is no intersection between the natural neighbor sets of two outlier points. If the scattered data meets the assumption (1), then we can find $\overline{iN(\overline{N(\mathcal{O})})} = \mathcal{O}$. It provides a guarantee for the reasonability of getting the quasi-outlier point set $\mathcal{O}'$ in the Algorithm 1. If the scattered data does not meet the assumption (1), there may be $\mathcal{O} \subseteq \overline{iN(\overline{N(\mathcal{O})})}$. In this case, our MMLS method is still valid, but its approximation effect may decrease. The existence of the assumption (2) is only convenient to analyze the property of the core function. Because of $h \leq h_M$, the assumption (3) makes that $\mathbf{X}$ is quasi-uniform[5] in Ω. What is more, as $h \to 0$, then $h_M \to 0$ and $h_m \to 0$. The sampled data meeting assumption (3) means that the constant K on different h is the same. The meaning of the sampled data meeting assumption (4) is similar.*

The weight function at the points on the boundary of the triangular mesh contains too many possibilities, so we only analyse the situation that the points are inside the triangular mesh. Hence we assume that the points in the natural neighbor set $N(\mathbf{x}_i) = \{\mathbf{x}_{i_j}\}_{j=1}^{M_i}$ are counterclockwise according to their index $i_1, i_2, \ldots, i_{M_i}$, as shown in Fig.5. In particular, $\mathbf{x}_{i_0} = \mathbf{x}_{i_{M_i}}$ and $\mathbf{x}_{i_{M_i}+1} = \mathbf{x}_{i_1}$.

Let the area of the triangle $\triangle_{(\mathbf{x}_i, \mathbf{x}_{i_j}, \mathbf{x}_{i_{j+1}})}$ be S_{i_j}, and the angle corresponding to

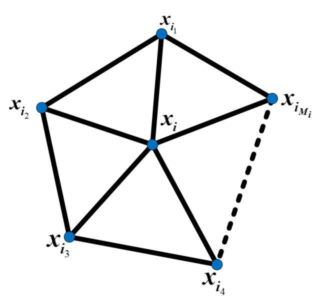

Fig. 5: Point $\mathbf{x}_i$ and its natural neighbors

$\mathbf{x}_i$ in this triangle be θ_{i_j}. According to the assumption (3), all the triangles are confined. For all i and j, there are three inequalities

$$\min\{\sqrt{1-\frac{K^2}{4}}, \frac{1}{K}\sqrt{1-\frac{1}{4K^2}}\} \leq \sin(\theta_{i_j}) \leq 1, \tag{3.3}$$

$$\frac{1}{2}{h_m}^2 \cdot min\{\frac{1}{K}\sqrt{1-\frac{1}{4K^2}}, \sqrt{1-\frac{K^2}{4}}\} \leq S_{i_j} < \frac{1}{2}{h_M}^2, \tag{3.4}$$

$$3 \le M_i \le \frac{2\pi}{\arccos(\max\{\frac{K}{2}, 1-\frac{1}{2K^2}\})}. \tag{3.5}$$

We use $\overline{M}$ to denote the upper bound of M_i, namely $\overline{M} = (2\pi)/\arccos(\max\{\frac{K}{2}, 1-\frac{1}{2K^2}\})$. s_m is the lower bound of $\sin(\theta_{i_j})$, namely $s_m = \min\{\sqrt{1-\frac{K^2}{4}}, \frac{1}{K}\sqrt{1-\frac{1}{4K^2}}\}$. It is important to note that both of these bounds depend only on K in the assumption (3).

The derivation of these three inequalities requires only some simple geometric knowledge in addition to the assumption (4). Refer to Appendix A for the detailed derivation.

From the definition of the weight function (3.2), we can find that the core function plays an important role in it. Therefore, before analyzing the property of the weight function ω, we first analyze the property of the $core(\mathbf{x})$.

At point $\mathbf{x}_i$,

$$core(\mathbf{x}_i) = \sum_{j=1}^{M_i} \left| \mathbf{u}_{i_j} \cdot \widetilde{\mathbf{p}}_i - \frac{\widetilde{f}_{i_j} - \widetilde{f}_i}{\|\overrightarrow{\mathbf{x}_i\mathbf{x}_{i_j}}\|} \right|.$$

In the calculation of $\widetilde{\mathbf{p}}_i$ by the method in Section 2.2, the approximate gradient obtained by the linear interpolation on the triangle $\triangle_{(\mathbf{x}_i, \mathbf{x}_{i_j}, \mathbf{x}_{i_{j+1}})}$ is written as $\mathbf{g}_{i_j}$. Let $\mathbf{x}_i = (x_i, y_i)$. Then on the triangle $\triangle_{(\mathbf{x}_i, \mathbf{x}_{i_j}, \mathbf{x}_{i_{j+1}})}$, there are

$$\begin{aligned}
\alpha_{i_j} &= (y_{i_j} - y_i)(\widetilde{f}_{i_{j+1}} - \widetilde{f}_i) - (y_{i_{j+1}} - y_i)(\widetilde{f}_{i_j} - \widetilde{f}_i), \\
\beta_{i_j} &= (x_{i_{j+1}} - x_i)(\widetilde{f}_{i_j} - \widetilde{f}_i) - (x_{i_j} - x_i)(\widetilde{f}_{i_{j+1}} - \widetilde{f}_i), \\
\gamma_{i_j} &= (x_{i_j} - x_i)(y_{i_{j+1}} - y_i) - (x_{i_{j+1}} - x_i)(y_{i_j} - y_i).
\end{aligned} \tag{3.6}$$

Therefore, we can obtain

$$\mathbf{g}_{i_j} = (-\frac{\alpha_{i_j}}{\gamma_{i_j}}, -\frac{\beta_{i_j}}{\gamma_{i_j}})^T, \tag{3.7}$$

$$S_{i_j} = \frac{1}{2}\gamma_{i_j}. \tag{3.8}$$

Let $\mathbf{u}_{i_j}$ denote the unit direction row vector from $\mathbf{x}_i$ to $\mathbf{x}_{i_j}$. Define the matrix $\mathbf{U}_j = \begin{bmatrix} \mathbf{u}_{i_j} \\ \mathbf{u}_{i_{j+1}} \end{bmatrix}$. We use df_{i_j} to denote the difference $\frac{\widetilde{f}_{i_j} - \widetilde{f}_i}{\|\overrightarrow{\mathbf{x}_i\mathbf{x}_{i_j}}\|}$, and d_{i_j} to denote $\|\overrightarrow{\mathbf{x}_i\mathbf{x}_{i_j}}\|$. Since $\mathbf{u}_{i_j} = \frac{1}{d_{i_j}}(x_{i_j} - x_i, y_{i_j} - y_i)$, after an easy manipulation there is

$$\mathbf{U}_j \cdot \mathbf{g}_{i_j} = \begin{bmatrix} df_{i_j} \\ df_{i_{j+1}} \end{bmatrix}. \tag{3.9}$$

Use $\mathbf{p}_i$ to denote the exact gradient for $f(\mathbf{x})$ at $\mathbf{x}_i$. Then there is

$$\mathbf{U}_j(\mathbf{g}_{i_j}-\mathbf{p}_i)=\begin{bmatrix} df_{i_j}-\frac{\partial f}{\partial \mathbf{u}_{i_j}}(\mathbf{x}_i) \\ df_{i_{j+1}}-\frac{\partial f}{\partial \mathbf{u}_{i_{j+1}}}(\mathbf{x}_i) \end{bmatrix}. \tag{3.10}$$

Use $\widetilde{R}_{i_j}$ to denote $df_{i_j}-\frac{\partial f}{\partial \mathbf{u}_{i_j}}(\mathbf{x}_i)$. Since $f(\mathbf{x})$ is $C^2(\Omega)$, we use Lagrange mean value theorem twice to get

$$\begin{aligned}\widetilde{R}_{i_j} &= \frac{\partial f}{\partial \mathbf{u}_{i_j}}(\xi_{i_j})-\frac{\partial f}{\partial \mathbf{u}_{i_j}}(\mathbf{x}_i)+\frac{e_{i_j}-e_i}{d_{i_j}} \\ &= \frac{\partial^2 f}{\partial \mathbf{u}_{i_j}{}^2}(\xi'_{i_j})\|\mathbf{x}_i-\xi_{i_j}\|+\frac{e_{i_j}-e_i}{d_{i_j}},\end{aligned} \tag{3.11}$$

where the ξ_{i_j} and ξ'_{i_j} are the points on the line segment $\overrightarrow{\mathbf{x}_i\mathbf{x}_{i_j}}$.
If there is no noise, then $R_{i_j}=\frac{\partial^2 f}{\partial \mathbf{u}_{i_j}{}^2}(\xi'_{i_j})\|\mathbf{x}_i-\xi_{i_j}\|$.
We rewrite formula (3.10) as

$$\mathbf{g}_{i_j}-\mathbf{p}_i=\mathbf{U}_j^{-1}\cdot\begin{bmatrix}\widetilde{R}_{i_j} \\ \widetilde{R}_{i_{j+1}}\end{bmatrix}, \tag{3.12}$$

here

$$\mathbf{U}_j^{-1}=\begin{bmatrix}(y_{i_{j+1}}-y_i)\frac{d_{i_j}}{\gamma_{i_j}} & -(y_{i_j}-y_i)\frac{d_{i_{j+1}}}{\gamma_{i_j}} \\ -(x_{i_{j+1}}-x_i)\frac{d_{i_j}}{\gamma_{i_j}} & (x_{i_j}-x_i)\frac{d_{i_{j+1}}}{\gamma_{i_j}}\end{bmatrix}. \tag{3.13}$$

Let $S=\sum_{j=1}^{M_i}S_{i_j}$. According to (3.4) and (3.5), a rough bound of S can be obtained,

$$s_m {h_m}^2 \le S \le \frac{1}{2}\overline{M}{h_M}^2. \tag{3.14}$$

Back to the approximate gradient $\widetilde{\mathbf{p}}_i$, there is

$$\widetilde{\mathbf{p}}_i=\sum_{k=1}^{M_i}\frac{S_{i_k}}{S}\cdot\mathbf{g}_{i_k}. \tag{3.15}$$

Because of $\sum_{k=1}^{M_i}(S_{i_k}/S)=1$ and formula (3.12), there is

$$\mathbf{u}_{i_j}(\widetilde{\mathbf{p}}_i-\mathbf{p}_i)=\mathbf{u}_{i_j}\cdot\sum_{k=1}^{M_i}\frac{S_{i_k}}{S}(\mathbf{g}_{i_k}-\mathbf{p}_i)=\sum_{k=1}^{M_i}\mathbf{u}_{i_j}\cdot\frac{S_{i_k}}{S}\cdot\mathbf{U}_k^{-1}\begin{bmatrix}\widetilde{R}_{i_k}\\ \widetilde{R}_{i_{k+1}}\end{bmatrix}. \tag{3.16}$$

By substituting (3.13) and (3.8) into (3.16), a tedious calculation gives

$$\mathbf{u}_{i_j}(\widetilde{\mathbf{p}}_i-\mathbf{p}_i)=\sum_{k=1}^{M_i}Q_j^k(\mathbf{x}_i,\mathbf{X})\cdot\widetilde{R}_{i_k}, \tag{3.17}$$

here

$$Q_j^k(\mathbf{x}_i,\mathbf{X})=\frac{1}{2S}\frac{d_{i_k}}{d_{i_j}}\left(\left(x_{i_j}-x_i\right)\left(y_{i_{k+1}}-y_{i_{k-1}}\right)-\left(y_{i_j}-y_i\right)\left(x_{i_{k+1}}-x_{i_{k-1}}\right)\right). \tag{3.18}$$

Refer to Appendix B for the detailed derivation of (3.17) and (3.18).
In formula (3.18), we find that

$$\left(x_{i_j}-x_i\right)\left(y_{i_{k+1}}-y_{i_{k-1}}\right)-\left(y_{i_j}-y_i\right)\left(x_{i_{k+1}}-x_{i_{k-1}}\right)=\det\left(\begin{bmatrix}\overrightarrow{\mathbf{x}_i\mathbf{x}_{i_j}}\\ \overrightarrow{\mathbf{x}_i\mathbf{x}_{i_{k+1}}}\end{bmatrix}\right)-\det\left(\begin{bmatrix}\overrightarrow{\mathbf{x}_i\mathbf{x}_{i_j}}\\ \overrightarrow{\mathbf{x}_i\mathbf{x}_{i_{k-1}}}\end{bmatrix}\right), \tag{3.19}$$

here, $\overrightarrow{\mathbf{x}_i\mathbf{x}_{i_j}}$, $\overrightarrow{\mathbf{x}_i\mathbf{x}_{i_{k+1}}}$ and $\overrightarrow{\mathbf{x}_i\mathbf{x}_{i_{k-1}}}$ are row vector.
According to (3.4) and (3.14), for every $Q_j^k(\mathbf{x}_i,\mathbf{X})$, there is

$$|Q_j^k(\mathbf{x}_i,\mathbf{X})|\leq\frac{K^3}{s_m}. \tag{3.20}$$

In particular, according to (3.6) and (3.8), if $j=k$, then there is

$$Q_k^k(\mathbf{x}_i,\mathbf{X})=\frac{S_{i_k}+S_{i_{k-1}}}{S}. \tag{3.21}$$

Because the number of triangles is greater than or equal to 3, we can get that

$$0<Q_k^k(\mathbf{x}_i,\mathbf{X})\leq\frac{2K^2}{2K^2+s_m}<1. \tag{3.22}$$

Refer to Appendix C for the detailed derivation of (3.20), (3.21)and (3.22).
Based on the above analysis, we have

$$\begin{aligned} core(\mathbf{x}_i) &= \sum_{j=1}^{M_i} \left|\mathbf{u}_{i_j} \cdot \widetilde{\mathbf{p}}_i - df_{i_j}\right| \\ &= \sum_{j=1}^{M_i} \left|\mathbf{u}_{i_j} \cdot \widetilde{\mathbf{p}}_i - \mathbf{u}_{i_j} \cdot \mathbf{p}_i + \mathbf{u}_{i_j} \cdot \mathbf{p}_i - df_{i_j}\right| \\ &= \sum_{j=1}^{M_i} \left| \left(\sum_{k=1}^{M} Q_j^k(\mathbf{x}_i, \mathbf{X}) \widetilde{R}_{i_k} \right) - \widetilde{R}_{i_j} \right|. \end{aligned} \tag{3.23}$$

By the assumption (2), if $\mathbf{x}_i \notin \overline{N(\mathscr{O})}$, then for every j there is $\widetilde{R}_{i_j} = R_{i_j}$, so that $|\widetilde{R}_{i_j}| = |R_{i_j}| \leq \mathscr{M}(f)h_M$. In such a case, according to formula (3.5) and (3.20), there is

$$\begin{aligned} core(\mathbf{x}_i) &\leq \sum_{j=1}^{M_i} \left(\left(\sum_{k=1}^{M_i} \left| Q_j^k(\mathbf{x}_i, \mathbf{X}) \right| |R_{i_k}| \right) + \left| R_{i_j} \right| \right) \\ &\leq \mathscr{M}(f)h_M \sum_{j=1}^{M_i} \left(\left(\sum_{k=1}^{M_i} \left| Q_j^k(\mathbf{x}_i, \mathbf{X}) \right| \right) + 1 \right) \\ &\leq \left(\frac{\overline{M}^2 K^3}{s_m} + \overline{M} \right) \mathscr{M}(f)h_M. \end{aligned} \tag{3.24}$$

We define a group of constants

$$E_j^k(\mathbf{x}_i, \mathbf{X}) := \begin{cases} Q_j^k(\mathbf{x}_i, \mathbf{X}), & j \neq k, \\ Q_j^k(\mathbf{x}_i, \mathbf{X}) - 1, & j = k. \end{cases}$$

By the group of constants, we can rewrite formula (3.23) as

$$core(\mathbf{x}_i) = \sum_{j=1}^{M_i} \left| \sum_{k=1}^{M_i} E_j^k(\mathbf{x}_i, \mathbf{X}) \widetilde{R}_{i_j} \right|.$$

According to (3.22), if $j = k$, then there is $E_k^k(\mathbf{x}_i, \mathbf{X}) = Q_k^k(\mathbf{x}_i, \mathbf{X}) - 1$. Hence there is

$$-1 < E_k^k(\mathbf{x}_i, \mathbf{X}) \leq \frac{-s_m}{2K^2 + s_m} < 0. \tag{3.25}$$

Furthermore, for any k there is

$$\frac{s_m}{2K^2+s_m} \leq \left|E_k^k(\mathbf{x}_i,\mathbf{X})\right| < 1. \tag{3.26}$$

If $N(\mathbf{x}_i)\cap\mathscr{O}\neq\emptyset$, then there exists $1\leq p\leq M_i$, such that $\widetilde{R}_{i_p}=R_{i_p}+e_{i_p}/d_{i_p}$ and $\widetilde{R}_{i_j}=R_{i_j}$ if $j\neq p$. In such a case, according to formula (3.26), (3.24) and assumption (4) there is

$$\begin{aligned}
core(\mathbf{x}_i) &\geq \sum_{j=1}^{M_i}\left|E_j^p(\mathbf{x}_i,\mathbf{X})\frac{e_{i_p}}{d_{i_p}}\right| - \sum_{j=1}^{M_i}\left|\sum_{k=1}^{M_i}E_j^k(\mathbf{x}_i,\mathbf{X})R_{i_j}\right| \\
&\geq \left|E_p^p(\mathbf{x}_i,\mathbf{X})\frac{e_{i_p}}{d_{i_p}}\right| - \left(\frac{\overline{M}^2K^3}{s_m}+\overline{M}\right)\mathscr{M}(f)h_M \\
&\geq \frac{\mathscr{E}}{h_M}\cdot\frac{s_m}{2K^2+s_m} - \left(\frac{\overline{M}^2K^3}{s_m}+\overline{M}\right)\mathscr{M}(f)h_M.
\end{aligned} \tag{3.27}$$

If $\mathbf{x}_i\in\mathscr{O}$, then there is $\widetilde{R}_{i_j}=R_{i_j}-e_i/d_{i_j}$, $\forall j$. In such a case, there is

$$core(\mathbf{x}_i) \geq \sum_{j=1}^{M_i}\left|\sum_{k=1}^{M_i}E_j^k(\mathbf{x}_i,\mathbf{X})\frac{e_i}{d_{i_k}}\right| - \sum_{j=1}^{M_i}\left|\sum_{k=1}^{M_i}E_j^k(\mathbf{x}_i,\mathbf{X})R_{i_j}\right|. \tag{3.28}$$

According to (3.18) and (3.19), an easy computation gives

$$\begin{aligned}
\sum_{k=1}^{M_i}E_j^k(\mathbf{x}_i,\mathbf{X})\frac{e_i}{d_{i_k}} &= \left(\sum_{k=1}^{M_i}Q_j^k(\mathbf{x}_i,\mathbf{X})\frac{e_i}{d_{i_k}}\right) - \frac{e_i}{d_{i_j}} \\
&= \left(\frac{1}{2S}\frac{e_i}{d_{i_j}}\sum_{k=1}^{M_i}\left(\det\left(\begin{bmatrix}\overrightarrow{\mathbf{x}_i\mathbf{x}_{i_j}}\\ \overrightarrow{\mathbf{x}_i\mathbf{x}_{i_{k+1}}}\end{bmatrix}\right)-\det\left(\begin{bmatrix}\overrightarrow{\mathbf{x}_i\mathbf{x}_{i_j}}\\ \overrightarrow{\mathbf{x}_i\mathbf{x}_{i_{k-1}}}\end{bmatrix}\right)\right)\right) - \frac{e_i}{d_{i_j}} \\
&= \frac{1}{2S}\frac{e_i}{d_{i_j}}\det\left(\begin{bmatrix}\overrightarrow{\mathbf{x}_i\mathbf{x}_{i_j}}\\ \sum_{k=1}^{M_i}\overrightarrow{\mathbf{x}_i\mathbf{x}_{i_{k+1}}}-\sum_{k=1}^{M_i}\overrightarrow{\mathbf{x}_i\mathbf{x}_{i_{k-1}}}\end{bmatrix}\right) - \frac{e_i}{d_{i_j}} \\
&= -\frac{e_i}{d_{i_j}}.
\end{aligned} \tag{3.29}$$

If substituting (3.29) into (3.28), we obtain

$$core(\mathbf{x}_i) \geq \frac{\mathscr{E}}{h_M} - \left(\frac{\overline{M}^2K^3}{s_m}+\overline{M}\right)\mathscr{M}(f)h_M. \tag{3.30}$$

For brevity, we define two constants that only depend on the assumptions (1)-(3)

$$C_1 := \min\{\frac{s_m}{2K^2+s_m}, 1\} = \frac{s_m}{2K^2+s_m} < \frac{1}{3},$$
$$C_2 := \frac{\overline{M}^2 K^3}{s_m} + \overline{M} > 0.$$

It is obvious that C_1 and C_2 depend on the constants s_m and $\overline{M}$. We note that $\overline{M} = (2\pi)/\arccos(\max\{\frac{K}{2}, 1-\frac{1}{2K^2}\})$ and $s_m = \min\{\frac{1}{K}\sqrt{1-\frac{1}{4K^2}}, \sqrt{1-\frac{K^2}{4}}\}$, so that C_1 and C_2 only depend on the constant K in assumption (3).
According to the formula (3.24), (3.27), and (3.30), if for any h the sampled data satisfy assumptions (1)-(4), then we can infer two conclusions:

c.1 If $\mathbf{x}_i \in \overline{N(\mathscr{O})}$, then as $h \to 0$, $core(\mathbf{x}_i) \geq C_1 \frac{\mathscr{E}}{h_M} - C_2 \mathscr{M}(f) h_M \to \infty$.

c.2 If $\mathbf{x}_i \notin \overline{N(\mathscr{O})}$, then as $h \to 0$, $core(\mathbf{x}_i) \leq C_2 \mathscr{M}(f) h_M \to 0$.

Remark 3.3. *The two conclusions indicate that for a certain node set* $\mathbf{X}$ *and sampling function, the larger* $core(\mathbf{x}_i)$ *is, the more likely* $\mathbf{x}_i \in \overline{N(\mathscr{O})}$. *What is more, there exists* h_0*, such that if* $h_M \leq h_0$*, then there is* $core(\mathbf{x}_i) \leq core(\mathbf{x}_l)$ *for any* $\mathbf{x}_l \in \overline{N(\mathscr{O})}$ *and any* $\mathbf{x}_i \notin \overline{N(\mathscr{O})}$.

After the asymptotic property of the core function is clear, we can begin to analyze the quasi-outlier point set $\mathscr{O}'$ obtained by Algorithm 1.
Assume that there is another point $\mathbf{x}_l \in \mathbf{X}$ which has M_l natural neighbors, i.e. $N(\mathbf{x}_l) = \{\mathbf{x}_{l_j}\}_{j=1}^{M_l}$. According to the conclusion c.2, if $\mathbf{x}_i, \mathbf{x}_l \in \mathbf{X} \backslash \overline{N(\mathscr{O})}$, there is

$$|core(\mathbf{x}_i) - core(\mathbf{x}_l)| \leq 2C_2 \mathscr{M}(f) h_M. \tag{3.31}$$

If $\mathbf{x}_i \in \overline{N(\mathscr{O})}$ and $\mathbf{x}_l \in \mathbf{X} \backslash \overline{N(\mathscr{O})}$, there is

$$|core(\mathbf{x}_i) - core(\mathbf{x}_l)| \geq C_1 \frac{\mathscr{E}}{h_M} - 2C_2 \mathscr{M}(f) h_M. \tag{3.32}$$

From (3.31) and (3.32), we can explain the effectiveness of Algorithm 1. By means of the sorting operation in Algorithm 1, it is easy to show that

Proposition 3.1. *Suppose that the sampled data satisfy assumptions (1)-(4) and the sampling function* $f(\mathbf{x}) \in C^2(\Omega)$. *For the Algorithm 1 in Section 3.1.1, if the control parameter* $\delta > 0$ *satisfies the following inequality*

$$\delta \geq 2C_2 \mathscr{M}(f) \cdot h_M,$$

then there is $C' \subseteq N(\mathscr{O})$, namely $\mathscr{O}' \subseteq \mathscr{O}$.
If the control parameter $\delta > 0$ satisfies the following inequality

$$\delta \leq C_1 \frac{\mathscr{E}}{h_M} - 2C_2 \mathscr{M}(f) h_M,$$

then there is $N(\mathscr{O}) \subseteq C'$, namely $\mathscr{O} \subseteq \mathscr{O}'$.

Furthermore, we can obtain the following proposition based on the conditions in Proposition 3.1.

Proposition 3.2. *Suppose the sampled data satisfy assumptions (1)-(4) and the sampling function $f(\mathbf{x}) \in C^2(\Omega)$. $\mathscr{O}$ is the unknown outlier point set and $\mathscr{O}'$ is obtained by Algorithm 1. For any control parameter $\delta > 0$, $\exists h_0 > 0$, s.t. if $h_M \leq h_0$, then $\mathscr{O}' = \mathscr{O}$.*

Due to the two conclusions c.1, c.2 about core function and Proposition 3.2, there are two important conclusions about the weight function

$$\omega(\mathbf{x}_i, \delta) = \begin{cases} \frac{1}{1+core(\mathbf{x}_i)}, & \mathbf{x}_i \in \mathscr{O}', \\ 1, & \mathbf{x}_i \in X \backslash \mathscr{O}'. \end{cases}$$

*1 If $\mathbf{x}_i$ is an outlier point in node set $\mathbf{X}$, no matter what the control parameter $\delta > 0$ is, then as $h \to 0$, $\omega(\mathbf{x}_i, \delta) \to 0$.

*2 If $\mathbf{x}_i$ is not the outlier point in node set $\mathbf{X}$, no matter what the control parameter $\delta > 0$ is , then as $h \to 0$, $\omega(\mathbf{x}_i, \delta) \to 1$.

So far, we have obtained the asymptotic property of the weight function, then we would explain that the method using the weight function to weaken the influence of outliers is theoretically feasible.

Remark 3.4. *The above analysis also indicates that for a certain node set $\mathbf{X}$ and sampling function, if δ satisfies the two inequalities in Proposition 3.1, then $\mathscr{O}' = \mathscr{O}$. If δ satisfies the second inequality, then $\mathscr{O} \subseteq \mathscr{O}'$,namely all the outlier points are found. Therefore the weight function $\omega(\mathbf{x}_i, \delta)$ with a small δ is still guaranteed to weaken the influence of outlier points. The existence of the weight function allows us to use $\mathscr{O}'$ to obtain an ideal approximation effect and avoid expensive computation. We suggest to take $\delta = 1/(3h_M)$, if the effect is not ideal, reduce δ slightly.*

3.2. The theoretical basis for the construction of the weight in MMLS

Assume that the approximation function space in problem (1.1) is the polynomial space $\Pi_m = span\{p_1(\mathbf{x}), \ldots, p_n(\mathbf{x})\}$ of total degree $\leq m$ and the normal equation solving the MLS problem (1.1) is

$$\mathbf{Ac} = \mathbf{b}, \tag{3.33}$$

where $\mathbf{c} = (c_1, \ldots, c_n)^T$ is the coefficient vector of the approximation function $p^*(\mathbf{x}) = \sum_{i=1}^{n} c_i \cdot p_i(\mathbf{x})$. The expressions for matrix $\mathbf{A}$ and vector $\mathbf{b}$ are as follows

$$\mathbf{A} = \begin{bmatrix} \sum_{i=1}^{N} p_1(\mathbf{x}_i)p_1(\mathbf{x}_i)\theta(\|\mathbf{x}-\mathbf{x}_i\|) & \cdots & \sum_{i=1}^{N} p_1(\mathbf{x}_i)p_n(\mathbf{x}_i)\theta(\|\mathbf{x}-\mathbf{x}_i\|) \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^{N} p_n(\mathbf{x}_i)p_1(\mathbf{x}_i)\theta(\|\mathbf{x}-\mathbf{x}_i\|) & \cdots & \sum_{i=1}^{N} p_n(\mathbf{x}_i)p_n(\mathbf{x}_i)\theta(\|\mathbf{x}-\mathbf{x}_i\|) \end{bmatrix},$$

$$\mathbf{b} = \begin{bmatrix} \sum_{i=1}^{N} p_1(\mathbf{x}_i)f_i\theta(\|\mathbf{x}-\mathbf{x}_i\|) \\ \vdots \\ \sum_{i=1}^{N} p_n(\mathbf{x}_i)f_i\theta(\|\mathbf{x}-\mathbf{x}_i\|) \end{bmatrix}.$$

Now, remove a point at random from node set $\mathbf{X}$, it may be set as $\mathbf{x}_1$. In this case, the system (3.33) to solve the new MLS problem becomes the following form

$$\mathbf{A}'\mathbf{c}' = \mathbf{b}'. \tag{3.34}$$

We set

$$\mathbf{A}_1 = \begin{bmatrix} p_1(\mathbf{x}_1)p_1(\mathbf{x}_1)\theta(\|\mathbf{x}-\mathbf{x}_1\|) & \cdots & p_1(\mathbf{x}_1)p_n(\mathbf{x}_1)\theta(\|\mathbf{x}-\mathbf{x}_1\|) \\ \vdots & \ddots & \vdots \\ p_n(\mathbf{x}_1)p_1(\mathbf{x}_1)\theta(\|\mathbf{x}-\mathbf{x}_1\|) & \cdots & p_n(\mathbf{x}_1)p_n(\mathbf{x}_1)\theta(\|\mathbf{x}-\mathbf{x}_1\|) \end{bmatrix},$$

$$\mathbf{b}_1 = \begin{bmatrix} p_1(\mathbf{x}_1)f_1\theta(\|\mathbf{x}-\mathbf{x}_1\|) \\ \vdots \\ p_n(\mathbf{x}_1)f_1\theta(\|\mathbf{x}-\mathbf{x}_1\|) \end{bmatrix}.$$

Then, there are $\mathbf{A}' = \mathbf{A} - \mathbf{A}_1$ and $\mathbf{b}' = \mathbf{b} - \mathbf{b}_1$.
Typically, we assume that matrix $\mathbf{A}$ and matrix $\mathbf{A}'$ are symmetric positive definite. Hence,

$$\mathbf{A}'(\mathbf{c}-\mathbf{c}') = \mathbf{b}_1 - \mathbf{A}_1\mathbf{c} = \begin{bmatrix} p_1(\mathbf{x}_1)(f_1 - p^*(\mathbf{x}_1)\theta(\|\mathbf{x}-\mathbf{x}_1\|)) \\ \vdots \\ p_n(\mathbf{x}_1)(f_1 - p^*(\mathbf{x}_1)\theta(\|\mathbf{x}-\mathbf{x}_1\|)) \end{bmatrix}. \tag{3.35}$$

Under the assumptions (3) and (5) at the begin in Section 3.1.3, Corollary 4.8 in [5] indicates that if the sampling function $f(\mathbf{x}) \in C^n(\Omega)$, then there is

$$|p^*(\mathbf{x}) - f(\mathbf{x})| < O(h^n), \tag{3.36}$$

where h is the filling distance of point set $\mathbf{X}$. Thus there is

$$\|\mathbf{A}'(\mathbf{c}-\mathbf{c}')\| \le M_p \theta(\|\mathbf{x}-\mathbf{x}_1\|) O(h^n), \tag{3.37}$$

here $M_p = \|(p_1(x_1), \ldots, p_n(x_1))^T\|$.
Because $\mathbf{A}'$ is symmetric positive definite, we can obtain the following proposition.

Proposition 3.3. *Let* $\mathbf{c}$ *be the solution of system (3.33) based on* $\mathbf{X}$ *and* $\mathbf{c}'$ *be the solution of system (3.34) based on* $\mathbf{X}\backslash\{\mathbf{x}_1\}$. *Then there is*

$$\lim_{h\to 0} \mathbf{c}' = \mathbf{c}.$$

Proposition 3.3 shows that if the sampling point is sufficiently dense, then the influence of the removal of a point on the approximation is small enough.
Assume that there is an outlier point in $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ which may be be set as $\mathbf{x}_1$, and the corresponding sample value is written as $\widetilde{f}_1 = f_1 + e_1$ with large $|e_1|$. In this case, the system of MLS problem is

$$\mathbf{A}\widetilde{\mathbf{c}} = \widetilde{\mathbf{b}}, \tag{3.38}$$

where

$$\widetilde{\mathbf{b}} = \begin{bmatrix} p_1(\mathbf{x}_1)\widetilde{f}_1\theta(\|\mathbf{x}-\mathbf{x}_1\|) + \sum_{i=2}^N p_1(\mathbf{x}_i) f_i \theta(\|\mathbf{x}-\mathbf{x}_i\|) \\ \vdots \\ p_n(\mathbf{x}_1)\widetilde{f}_1\theta(\|\mathbf{x}-\mathbf{x}_1\|) + \sum_{i=2}^N p_n(\mathbf{x}_i) f_i \theta(\|\mathbf{x}-\mathbf{x}_i\|) \end{bmatrix}.$$

From (3.38) we can find that the existence of outlier only transform $\mathbf{b}$ in (3.33) into $\widetilde{\mathbf{b}}$, then the solution $\mathbf{c}$ of equation (3.33) becomes $\widetilde{\mathbf{c}}$. Set

$$\widetilde{\mathbf{b}_1} = \begin{bmatrix} p_1(\mathbf{x}_1)\widetilde{f}_1\theta(\|\mathbf{x}-\mathbf{x}_1\|) \\ \vdots \\ p_n(\mathbf{x}_1)\widetilde{f}_1\theta(\|\mathbf{x}-\mathbf{x}_1\|) \end{bmatrix}.$$

With the preceding notations, there is

$$(\mathbf{A}' + \mathbf{A}_1)\widetilde{\mathbf{c}} = \widetilde{\mathbf{b}_1} + \mathbf{b}'. \tag{3.39}$$

After equivalent transformation, there is

$$\mathbf{A}'(\widetilde{\mathbf{c}}-\mathbf{c}') = \widetilde{\mathbf{b}_1} - \mathbf{A}_1\widetilde{\mathbf{c}}. \tag{3.40}$$

According to the compatibility of the matrix norm and vector norm, there is

$$\|\widetilde{\mathbf{c}}-\mathbf{c}'\| \leq \|\mathbf{A}'^{-1}\|\|(\widetilde{\mathbf{b}_1} - \mathbf{A}_1\widetilde{\mathbf{c}})\|. \tag{3.41}$$

From the inequality (3.41) we can obtain that: $\forall \varepsilon > 0$, if $\|\widetilde{\mathbf{b}_1} - \mathbf{A}_1\widetilde{\mathbf{c}}\| < \frac{\varepsilon}{\|\mathbf{A}'^{-1}\|}$, then $\|\widetilde{\mathbf{c}}-\mathbf{c}'\| < \varepsilon$.
Hence if we put a weight ω into the system (3.39) in the location which is corresponding to the outlier, then it will make the system transformed into

$$(\mathbf{A}' + \omega\mathbf{A}_1)\widetilde{\mathbf{c}} = \mathbf{b}' + \omega\widetilde{\mathbf{b}_1}. \tag{3.42}$$

From the above analysis, we can find that : $\forall \varepsilon > 0$, if ω s.t. $|\omega|\|\widetilde{\mathbf{b}_1} - \mathbf{A}_1\widetilde{\mathbf{c}}\| < \frac{\varepsilon}{\|\mathbf{A}'^{-1}\|}$, then $\|\widetilde{\mathbf{c}}-\mathbf{c}'\| < \varepsilon$. We can obtain the following proposition.

Proposition 3.4. *Assume that $\widetilde{\mathbf{c}}$ is the solution of the weighted MLS problem whose system is (3.42), and $\mathbf{c}'$ is the solution of the MLS problem only removed the outlier point. Then there is*

$$\lim_{\omega\to 0}\widetilde{\mathbf{c}} = \mathbf{c}'.$$

Combining the Proposition 3.3 with Proposition 3.4, we get the following desired result.

Proposition 3.5. *On the above assumptions ,in the weighted MLS whose system is (3.42), as $h \to 0$, if the weight $\omega \to 0$, then there is*

$$\lim_{h\to 0}\widetilde{\mathbf{c}} = \mathbf{c},$$

where $\widetilde{\mathbf{c}}$ is the solution of the weighted MLS problem, and $\mathbf{c}$ is the solution of the MLS without outliers.

Remark 3.5. *Although the discussion in Proposition 3.3 - 3.5 only considers the removal of one point or the existence of one outlier point, it is easy to be generalized to several points. And due to Proposition 3.5, we know that if we can find a weight function ω satisfying the condition, then we can obtain an approximate solution to $\mathbf{c}$.*

3.3. *The modified moving least squares method*

According to the asymptotic property analysis of the weight function $\omega(\mathbf{x}_i, \delta)$ in Section 3.1, we can find the weight function satisfies the conditions proposed in Proposition 3.5. Therefore, we give a modified moving least squares: the approximation function $p^*(\mathbf{x}) \in \Pi_m$ is obtained by minimizing the following weighted l_2 norm

$$\bar{E}_{2,\theta,x}(p) = \sum_{i=1}^{N} \omega(\mathbf{x}_i, \delta)(f_i - p(\mathbf{x}_i))^2 \theta(\|\mathbf{x} - \mathbf{x_i}\|), \tag{3.43}$$

where θ is the weight function in MLS, and the weight function $\omega(\mathbf{x}_i, \delta)$ is defined as (3.2).

Remark 3.6. *If the scattered data satisfy the five assumptions (1)-(5) in Section 3.1.3 and the sampling function $f(\mathbf{x}) \in C^2(\Omega)$, then the approximation effect of MMLS is optimal. If they do not satisfy the five assumptions, the approximation effect will decrease, but the experiments indicate that our method is still effective.*

4. Numerical experiments

In this section, we first introduce the running environment of the experiments. Then four experiments are implemented. The first experiment shows the property of the *core* function. The second experiment compares the computational efficiency of three methods: MMLS, MLS and MLH. The third experiment compares the approximation effect of the three methods. The last experiment exhibits the anti-noise capability of MMLS and MLH.

4.1. *Preparation and detail of numerical experiments.*

The node set $\mathbf{X}$ in the following experiments is taken as the Halton point set ([17],[18]) in $(0,1) \times (0,1) \in \mathbb{R}^2$. The Halton point set in any dimension is a quasi-scattered point set that is generated by the van der Corput sequence. Our approximation function space is a two-variables cubic polynomial space $\Pi_3 = span\{1, x, y, x^2, xy, y^2, x^3, x^2y, xy^2, y^3\}$. In order to show the approximation effect clearly, we use the Franke test function proposed in [19] as our sampling function

$$\begin{aligned} f(x,y) =& \frac{3}{4}\exp(\frac{(9x-2)^2+(9y-2)^2}{4}) + \frac{3}{4}\exp(-\frac{(9x-2)^2}{49} - \frac{(9y+1)^2}{10}) \\ &- \frac{1}{5}\exp(-(9x-4)^2-(9y-7)^2) + \frac{1}{2}\exp(\frac{(9x-7)^2+(9y-3)^2}{4}). \end{aligned}$$

The weight $\theta(\|\mathbf{x}-\mathbf{x_i}\|)$ shared in three methods is

$$\theta(\|\mathbf{x}-\mathbf{x_i}\|)=\begin{cases}\frac{2}{3}-4(\frac{\|\mathbf{x}-\mathbf{x_i}\|}{r_{max}})^2+4(\frac{\|\mathbf{x}-\mathbf{x_i}\|}{r_{max}})^3, & \frac{\|\mathbf{x}-\mathbf{x_i}\|}{r_{max}}\le\frac{1}{2},\\ \frac{4}{3}-4\frac{\|\mathbf{x}-\mathbf{x_i}\|}{r_{max}}+4(\frac{\|\mathbf{x}-\mathbf{x_i}\|}{r_{max}})^2-\frac{4}{3}(\frac{\|\mathbf{x}-\mathbf{x_i}\|}{r_{max}})^3, & \frac{1}{2}<\frac{\|\mathbf{x}-\mathbf{x_i}\|}{r_{max}}\le 1,\\ 0, & \frac{\|\mathbf{x}-\mathbf{x_i}\|_2}{r_{max}}>1,\end{cases}$$

where the r_{max} is the maximum impact radius which is usually a constant. We set $r_{max}=0.1$ in those experiments. In MLH method, we take the small parameter in (1.2) $d=0.01$. In our modified moving least squares method, the control parameter remarked in Section 3.1.3 $\delta=1/(3h_M)$.

In $\mathbf{X}$, we pick the 5% of the points and add the disturbances to these corresponding sample values as the outliers. K stands for the number of the outliers points, namely $K=5\%\cdot N$. The disturbances would be picked whose absolute values are in the region $[2,5]$. In following experiments, the scattered data with outliers mean that only those K points' sample values are perturbed in the Halton point set $\mathbf{X}$, and the others are accurate.

The points in $\mathbf{X}$ from the region $(0.1,0.9)\times(0.1,0.9)$ are denoted as set $\{\mathbf{x}_m\}_{m\in I}$, and I is the index set. In all experiments, we calculate the approximate values $p^*(\mathbf{x}_m), m\in I$ by the three methods: MMLS, MLS and MLH. Then we use the error

$$MaxError_f=max\{|f_m-p^*(\mathbf{x}_m)|, m\in I\}$$

to measure the approximation effect of the three methods.

Finally, it is explained that all the experiments are implemented on CPU with Intel Core i7-7700hq, and main frequency is 2.8 GHz. The programming language which we use is Matlab R2014a.

4.2. The experiment of the asymptotic property of core function

In the weight function $\omega(\mathbf{x}_i,\delta)$, it can be found that the function $core(\mathbf{x}_i)$ is very important. The property " If $\mathbf{x}_i\in\overline{N(\mathscr{O})}$, then as $h\to 0$, $core(\mathbf{x}_i)\to\infty$." is particularly important. It is related not only to the selection of quasi-outlier point set, but also to weaken the effect of outliers .

In this experiment, we keep the number of outlier points $K=50$. Since the outlier point set $\mathscr{O}$ in our experiments is given in advance, we can obtain $c_{inf}=inf\{core(\mathbf{x}_i)|\mathbf{x}_i\in\overline{N(\mathscr{O})}\}$. Owing to the nestedness of Halton points, the "$h\to 0$" can be achieved by increasing the number of Halton points. The test function

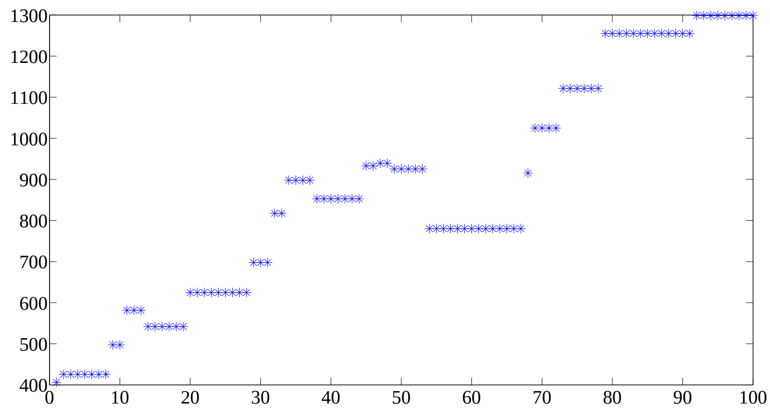

Fig. 6: The variation of c_{inf} as the number of sample points increases

$f(x,y) \in C^2([0,1] \times [0,1])$.
In order to verify the property, we only need to observe the change of the c_{inf} on different scattered data. Fig.6 shows the change of c_{inf} values observed at the different Halton point sets with the size of $N = 900 + 100 \cdot i, i = 1, \ldots, 100$, where the horizontal coordinate denotes the value of i. It can be seen that the values of c_{inf} increase as sampling points becoming denser. The result conforms the conclusion : If $\mathbf{x}_i \in \overline{N(\mathscr{O})}$, then as $h \to 0$, $core(\mathbf{x}_i) \to \infty$.

4.3. The comparison among the computational efficiency of MMLS, MLS, MLH

We will show their computational efficiency on 10 different point sets, whose size is $N = 1000 + i \cdot 200, i = 0, 1, \ldots, 9$, respectively. In order to amplify the differences among the three methods, we use each method's total time calculating $p^*(\mathbf{x}_m), m \in I$ to stand for the computational efficiency of every method.

Table 1: The computational time (unit:s) of MMLS, MLS and MLH

Method \ Number	1000	1200	1400	1600	1800
MMLS	5.7217	8.0771	11.6674	14.1885	17.4008
MLS	5.1136	7.3895	10.2114	13.4199	16.5842
MLH	12.4672	18.4805	23.7984	31.2693	38.6682
Method \ Number	2000	2200	2400	2600	2800
MMLS	22.0151	25.3133	30.8835	35.7785	44.9394
MLS	20.3999	24.1810	30.0594	35.7034	42.6873
MLH	48.8693	58.2602	69.9138	83.6967	99.1721

Table 1 shows that the computational times of MLS and MMLS are roughly equivalent. The result is actually predictable, because the difference between the two methods is only the weight function ω, and the function can be calculated at one time. It is obvious that the computational time of MMLS is much lower than that of MLH. The solution of MLH needs an iterative process, and the iterative initial value at each point is the solution of MLS. Therefore, It is not surprising that the computational efficiency of MLH is lower than that of MMLS.
However, Table 1 does not display the difference between MMLS and MLS clearly. In fact,the MMLS do more calculations to obtain the weight function. Therefore, we increase the values of N and show the time to obtain the weight function. The results are shown in Table 2.

Table 2: The computational time (unit:s) of MMLS and MLS on more sampling points

Method \ Number	10000	15000	20000	25000	30000
MMLS	401.1797	908.0058	1645.7756	2566.0294	3691.2025
t_w	4.0137	7.6506	10.1065	16.5409	18.1927
MLS	396.7398	900.3402	1646.6563	2570.8855	3690.8648

In Table 2, t_w denotes the time to calculate the weight function by MMLS. Table 2 shows that the increase of t_w is much lower than the increase of the time to solve the optimization problem. The reason is that for a set of scattered data, no matter how many times it needs to be calculated, the weight function needs to be calculated only once.

4.4. *The comparison among the approximation effect of MMLS, MLS, MLH*

In order to illustrate the effectiveness of Algorithm 1, we show the number of the quasi-outlier points in $\mathscr{O}'$ and use γ to denote the percentages of the found outlier points. K is the number of outlier points. In this experiment we use two point sets with size of N=2500 and N=5000. The results without noise and outliers are shown in Table 3.

Table 3: The $MaxError_f$ of the three methods without noise and outliers

Method	Error	N	K	$\lvert\mathscr{O}'\rvert$	γ
MMLS	1.521E-03	2500	0	0	–
MLS	1.521E-03	2500	0	–	–
MLH	1.460E-03	2500	0	–	–
MMLS	1.626E-03	5000	0	0	–
MLS	1.626E-03	5000	0	–	–
MLH	1.534E-03	5000	0	–	–

Table 3 demonstrates that Algorithm 1 dose not pick out any points when there are no outliers and noise in the data. At this point, the approximation effect of MLS and MMLS is the same. In addition, we find that the approximation effect of MLH is slightly ahead of that of MMLS on the two different N, but their error orders are the same. Therefore the approximation effect of the three methods is approximately equal in the absence of outliers and noise.
Then, we add the outliers made in Section 4.1 into the scattered data. The computational results are shown in Table 4.

Table 4: The $MaxError_f$ of the three methods without noise but with outliers

Method	Error	N	K	$\lvert\mathscr{O}'\rvert$	γ
MMLS	2.327E-03	2500	125	125+5	100%
MLS	1.4641	2500	125	–	–
MLH	5.295E-03	2500	125	–	–
MMLS	2.811E-03	5000	250	250+55	100%
MLS	8.488E-01	5000	250	–	–
MLH	3.726E-03	5000	250	–	–

Table 4 shows that MLS method is a failure when the scattered data contain outliers. However, MMLS and MLH are still effective, and the errors of MMLS are smaller than the errors of MLH on the two different N. Observing Table 3 and Table 4, we can see that the effect of MMLS with outliers is slightly less than the effect of MMLS without outliers, but their error orders are the same.
What is more, Table 3 and Table 4 show that if there are outliers, Algorithm 1 could find all the outliers points; if there are no outliers, Algorithm 1 could get an empty set, which means that the Algorithm 1 is valid. Furthermore, we increase the number of scattered data and keep $K = 5\% \cdot N$ to show the relationship between the number of falsely detected outlier points in $\mathscr{O}'$ and N in Table 5.

Table 5: The $|\mathscr{O}'|$ of MMLS method on different N

Number	10000	15000	20000	25000	30000
$\lvert\mathscr{O}'\rvert$	500+120	750+56	1000+156	1250+265	1500+327
p	24%	7.47%	15.6%	21.2%	21.8%

p in Table 5 denotes the ratio of falsely detected outlier points to true outlier points. The first three results in Table 5 indicate that if $\delta = \frac{1}{3h_M}$, there may be a large fluctuation of the ratio p on different N. At the same time, Table 5 indicates our strategy to set δ would keep p around 20%.

4.5. The anti-noise capability

This experiment aims to test the anti-noise capability of MMLS method. Let scattered data contain not only the outliers, but also noisy at every points. The noise ratio α indicates that at every point $\mathbf{x}_i$, there exists a random number $0 < \alpha_i \le \alpha$, s.t. $\tilde{f}_i = (1 \pm \alpha_i) \cdot f(\mathbf{x}_i)$. When the noise ratios α are 2% and 5%, the results are shown in Table 6 and Table 7 respectively.

Table 6: In the case of noise ratio of 2% and outliers, the $MaxError_f$ of MMLS and MLS

Method	Error	N	K	$\lvert\mathscr{O}'\rvert$	γ
MMLS	1.259E-02	2500	125	125+5	100%
MLH	1.639E-02	2500	125	–	–

Table 7: In the case of noise ratio of 5% and outliers, the $MaxError_f$ of MMLS and MLS

Method	Error	N	K	$\lvert\mathscr{O}'\rvert$	γ
MMLS	2.643E-02	2500	125	125+5	100%
MLH	4.190E-02	2500	125	–	–

Table 6 shows that when the noise ratio is 2%, the approximation effect of MMLS and MLH is almost the same. However, according to the Table 7 if the noise ratio is increased to 5%, we can find that the approximation effect of MMLS becomes better than that of MLH. The reason is that the structure of MMLS is similar to that of MLS. Therefore, we consider that MMLS inherits the anti-noise capability from MLS. What is more, Table 6 and Table 7 indicate that small noise dose not affect MLS to find the outlier points.

5. Conclusion

In this paper, we propose a modified moving least-squares method for the fitting of scattered data with outliers. The computational efficiency of the method is similar to that of classical MLS. Its approximation effect closes to that of MLH. Furthermore, the experiments indicate that the anti-noise capability of our method is better than that of MLH.

However, MMLS method is based on moving least-squares method, so its approximation effect will not be higher than that of MLS method when dealing with the scattered data without outliers, no matter how we optimize the weight function. Furthermore, our weight function only works on $\mathbf{X} \subseteq \mathbb{R}^2$ and needs mesh generation to obtain the approximate gradient. If one can combine our idea with a meshless method to obtain the approximate gradient and solve the theoretical analysis barrier, then the method can be generalized to $\mathbf{X} \subseteq \mathbb{R}^d$ for $d > 2$.

This method has many applications. Except for the direct application to the fitting of scattered data with outliers, the weight function ω in the method can be extracted into other methods that need to weaken the influence of the outliers. In addition, the weight function proposed in this paper can also be applied separately to selecting the outlier points from the scattered data.

Acknowledgement

The work is supported by National Natural Science Foundation no.91630203.

References

[1] D. H. Mclain, Drawing contours from arbitrary data points, Computer Journal 17 (17) (1974) 318–324.

[2] P. Lancaster, K. Salkauskas, Surfaces generated by moving least squares methods, Mathematics of Computation 37 (155) (1981) 141–158.

[3] D. Levin, The approximation power of moving least-squares, Mathematics of Computation 67 (224) (1998) 1517–1531.

[4] Y. Lipman, Stable moving least-squares, Journal of Approximation Theory 161 (1) (2009) 371–384.

[5] W. H., Scattered Data Approximation, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, 2004. doi:10.1017/CBO9780511617539.

[6] X. Li, S. Li, Analysis of the complex moving least squares approximation and the associated element-free galerkin method, Applied Mathematical Modelling 47.

[7] D. Mirzaei, R. Schaback, M. Dehghan, On generalized moving least squares and diffuse derivatives, Ima Journal of Numerical Analysis 32 (3) (2012) 983–1000(18).

[8] D. Mirzaei, Direct approximation on spheres using generalized moving least squares, Bit Numerical Mathematics (2017) 1–23.

[9] M. Dehghan, M. Abbaszadeh, Interpolating stabilized moving least squares (mls) approximation for 2d elliptic interface problems, Computer Methods in Applied Mechanics & Engineering (2017) 328.

[10] M. Amirfakhrian, H. Mafikandi, Approximation of parametric curves by Moving Least Squares method, Elsevier Science Inc., 2016.

[11] T. E. Dielman, Least absolute value regression: recent contributions, Journal of Statistical Computation & Simulation 75 (4) (2005) 263–286.

[12] D. Levin, Between moving least-squares and moving least-l_1, BIT Numerical Mathematics 55 (3) (2015) 781–796. doi:10.1007/s10543-014-0522-0. URL `https://doi.org/10.1007/s10543-014-0522-0`

[13] A. Amir, D. Levin, Quasi-interpolation and outliers removal, Numerical Algorithms 78 (3) (2018) 1–21.

[14] C. L. Lawson, Software for c^1 surface interpolation, Mathematical Software (1977) 161–194.

[15] D. F. Watson, Natural neighbour sorting, Australian Computer Journal 17 (4) (1985) 189–193.

[16] T. Goodman, H. Said, L. Chang, Local derivative estimation for scattered data interpolation, Applied Mathematics and Computation 68 (1) (1995) 41–50. doi:https://doi.org/10.1016/0096-3003(94)00086-J.
URL https://www.sciencedirect.com/science/article/pii/009630039400086J

[17] G. E. Fasshauer, Meshfree Approximation Methods with MATLAB, World Scientific,, 2007.

[18] J. H. Halton, On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals, Numerische Mathematik 2 (1) (1960) 84–90.

[19] R. J. Renka, R. Brown, Algorithm 792: Accuracy tests of ACM algorithms for interpolation of scattered data in the plane, ACM, 1999.

Appendix A.

From the Law of Cosines, there is

$$\cos(\theta_{i_j}) = \frac{{d_{i_j}}^2 + {d_{i_{j+1}}}^2 - \|\overrightarrow{\mathbf{x}_{i_{j+1}}\mathbf{x}_{i_j}}\|^2}{2d_{i_j}d_{i_{j+1}}}. \tag{A.1}$$

By the monotonicity of function $f(x,y,z) = (x^2+y^2-z^2)/2xy$ with respect to z on $[h_m,h_M]\times[h_m,h_M]\times[h_m,h_M]$, the following two equations are valid

$$\min_{h_m\le x\le h_M, h_m\le y\le h_M, h_m\le z\le h_M} f(x,y,z) = \min_{h_m\le x\le h_M, h_m\le y\le h_M} f(x,y,h_M), \tag{A.2}$$

$$\max_{h_m\le x\le h_M, h_m\le y\le h_M, h_m\le z\le h_M} f(x,y,z) = \max_{h_m\le x\le h_M, h_m\le y\le h_M} f(x,y,h_m). \tag{A.3}$$

Let $g_1(x,y) = f(x,y,h_M)$ and $g_2(x,y) = f(x,y,h_m)$. Then there is

$$\frac{\partial g_1}{\partial x} = \frac{x^2-(y^2-{h_M}^2)}{2x^2y} > 0.(h_m \le y \le h_M). \tag{A.4}$$

Similarly,

$$\frac{\partial g_1}{\partial y} = \frac{y^2-(x^2-{h_M}^2)}{2xy^2} > 0.(h_m \le x \le h_M). \tag{A.5}$$

According to (A.4) and (A.5), we get

$$\min_{h_m\le x\le h_M, h_m\le y\le h_M, h_m\le z\le h_M} f(x,y,z) = \frac{2{h_m}^2-{h_M}^2}{2{h_m}^2} \ge 1-\frac{K^2}{2}. \tag{A.6}$$

Using the same method, we know that there is only one stable point of the function $g_2(x,y)$ in $[h_m,h_M]\times[h_m,h_M]$ and the point is a minimum point, so that the maximum point must be on the boundary of $[h_m,h_M]\times[h_m,h_M]$.
Namely,

$$\max_{h_m\le x\le h_M, h_m\le y\le h_M} f(x,y,h_m) = \max\{g_2(h_m,h_M), g_2(h_M,h_m), g_2(h_M,h_M), g_2(h_m,h_m)\}. \tag{A.7}$$

According to assumption (3), there are

$$\frac{1}{2} \le g_2(h_m,h_M) = g_2(h_M,h_m) = \frac{h_M}{2h_m} \le \frac{K}{2} < 1,$$
$$g_2(h_m,h_m) = \frac{1}{2},$$
$$\frac{1}{2} \le g_2(h_M,h_M) = 1-\frac{{h_m}^2}{2{h_M}^2} \le 1-\frac{1}{2K^2} < \frac{7}{8}.$$

If $h_M/h_m = 1.1$, then $g_2(h_m,h_M) < g_2(h_M,h_M)$. However, if $h_M/h_m = 1.8$, then $g_2(h_m,h_M) > g_2(h_M,h_M)$. So that,

$$\max_{h_m \le x \le h_M, h_m \le y \le h_M, h_m \le z \le h_M} f(x,y,z) = \max\{\frac{h_M}{2h_m}, 1 - \frac{{h_m}^2}{2{h_M}^2}\}. \tag{A.8}$$

According to (A.1), (A.6) and (A.8), we get

$$1 - \frac{K^2}{2} \le \cos(\theta_{i_j}) \le \max\{\frac{h_M}{2h_m}, 1 - \frac{{h_m}^2}{2{h_M}^2}\} \le \max\{\frac{K}{2}, 1 - \frac{1}{2K^2}\}. \tag{A.9}$$

Due to $0 \le \theta_{i_j} \le \pi$, there is

$$0 < \arccos(\max\{\frac{K}{2}, 1 - \frac{1}{2K^2}\}) \le \theta_{i_j} \le \arccos(1 - \frac{K^2}{2}) < \pi. \tag{A.10}$$

Now, we consider the following optimization problem

$$\min_{\arccos(\max\{K/2, 1-1/(2K^2)\}) \le \theta \le \arccos(1-K^2/2)} \sin(\theta). \tag{A.11}$$

We know that there is only one stable point of the function $\sin(\theta)$ in $[0,\pi]$ and the point is a maximum point, then the minimum point must be on the boundary of $[\arccos(\max\{K/2, 1-1/(2K^2)\}), \arccos(1-K^2/2)]$. Namely,

$$\begin{aligned}&\min_{\arccos(\max\{K/2, 1-1/(2K^2)\}) \le \theta \le \arccos(1-K^2/2)} \sin(\theta)\\ &= \min\{\sin(\arccos(\max\{\frac{K}{2}, 1 - \frac{1}{2K^2}\})), \sin(\arccos(1 - \frac{K^2}{2}))\}.\end{aligned}$$

By $\sin^2(\theta) + \cos^2(\theta) = 1$, the bound of $\sin(\theta_{i_j})$ can be expressed as

$$\min\{\sqrt{1 - \frac{K^2}{4}}, \frac{1}{K}\sqrt{1 - \frac{1}{4K^2}}, K\sqrt{1 - \frac{K^2}{4}}\} \le \sin(\theta_{i_j}) \le 1. \tag{A.12}$$

Furthermore, the K in assumption (3) belongs to $[1,2]$, so that there is

$$\min\{\sqrt{1 - \frac{K^2}{4}}, \frac{1}{K}\sqrt{1 - \frac{1}{4K^2}}\} \le \sin(\theta_{i_j}) \le 1. \tag{A.13}$$

Because the area S_{i_j} of the triangle $\triangle_{(\mathbf{x}_i, \mathbf{x}_{i_j}, \mathbf{x}_{i_{j+1}})}$ is $\frac{1}{2} d_{i_j} d_{i_{j+1}} \sin(\theta_{i_j})$, the bound of S_{i_j} can be expressed as

$$\frac{1}{2}{h_m}^2 \cdot min\{\frac{1}{K}\sqrt{1 - \frac{1}{4K^2}}, \sqrt{1 - \frac{K^2}{4}}\} \le S_{i_j} < \frac{1}{2}{h_M}^2. \tag{A.14}$$

According to (A.10), the bound of M_i can be expressed as

$$\frac{2\pi}{\arccos(1-\frac{K^2}{2})} \leq M_i \leq \frac{2\pi}{\arccos(\max\{\frac{K}{2}, 1-\frac{1}{2K^2}\})}. \tag{A.15}$$

Further, it is impossible that there are less than two triangles, so that there is

$$3 \leq M_i \leq \frac{2\pi}{\arccos(\max\{\frac{K}{2}, 1-\frac{1}{2K^2}\})}. \tag{A.16}$$

It is important to note that formula (A.13) is formula (3.3), formula (A.14) is formula (3.4) and formula (A.16) is formula (3.5).

Appendix B.

According to formula (3.16), we know that

$$\mathbf{u}_{i_j}(\widetilde{\mathbf{p}}_i - \mathbf{p}_i) = \sum_{k=1}^{M_i} \mathbf{u}_{i_j} \cdot \frac{S_{i_k}}{S} \cdot \mathbf{U}_k^{-1} \begin{bmatrix} \widetilde{R}_{i_k} \\ \widetilde{R}_{i_{k+1}} \end{bmatrix}. \tag{B.1}$$

What is more, there are

$$\begin{aligned}
\mathbf{u}_{i_j} &= \frac{1}{d_{i_j}}[x_{i_j} - x_i, y_{i_j} - y_i], \\
S_{i_k} &= \frac{1}{2}\gamma_{i_k}, \\
\mathbf{U}_k^{-1} &= \begin{bmatrix} (y_{i_{k+1}} - y_i)\frac{d_{i_k}}{\gamma_{i_k}} & -(y_{i_k} - y_i)\frac{d_{i_{k+1}}}{\gamma_{i_k}} \\ -(x_{i_{k+1}} - x_i)\frac{d_{i_k}}{\gamma_{i_k}} & (x_{i_k} - x_i)\frac{d_{i_{k+1}}}{\gamma_{i_k}} \end{bmatrix}.
\end{aligned}$$

Further, we have

$$
\begin{aligned}
\mathbf{u}_{i_j}(\widetilde{\mathbf{p}}_i-\mathbf{p}_i) =& \sum_{k=1}^{M_i}\left(\frac{\mathbf{u}_{i_j}}{2S}\cdot\begin{bmatrix}(y_{i_{k+1}}-y_i)d_{i_k} & -(y_{i_k}-y_i)d_{i_{k+1}}\\ -(x_{i_{k+1}}-x_i)d_{i_k} & (x_{i_k}-x_i)d_{i_{k+1}}\end{bmatrix}\cdot\begin{bmatrix}\widetilde{R}_{i_k}\\ \widetilde{R}_{i_{k+1}}\end{bmatrix}\right)\\
=& \sum_{k=1}^{M_i}\left(\frac{1}{2S\cdot d_{i_j}}[x_{i_j}-x_i, y_{i_j}-y_i]\cdot\begin{bmatrix}(y_{i_{k+1}}-y_i)d_{i_k}\widetilde{R}_{i_k}-(y_{i_k}-y_i)d_{i_{k+1}}\widetilde{R}_{i_{k+1}}\\ -(x_{i_{k+1}}-x_i)d_{i_k}\widetilde{R}_{i_k}+(x_{i_k}-x_i)d_{i_{k+1}}\widetilde{R}_{i_{k+1}}\end{bmatrix}\right)\\
=& \frac{1}{2S\cdot d_{i_j}}\sum_{k=1}^{M_i}\left(\left((x_{i_j}-x_i)(y_{i_{k+1}}-y_i)-(x_{i_{k+1}}-x_i)(y_{i_j}-y_i)\right)d_{i_k}\widetilde{R}_{i_k}\right)\\
&+\frac{1}{2S\cdot d_{i_j}}\sum_{k=1}^{M_i}\left(\left(-(x_{i_j}-x_i)(y_{i_k}-y_i)+(x_{i_k}-x_i)(y_{i_j}-y_i)\right)d_{i_{k+1}}\widetilde{R}_{i_{k+1}}\right)\\
=& \frac{1}{2S\cdot d_{i_j}}\sum_{k=1}^{M_i}\left(\left((x_{i_j}-x_i)(y_{i_{k+1}}-y_i)-(x_{i_{k+1}}-x_i)(y_{i_j}-y_i)\right)d_{i_k}\widetilde{R}_{i_k}\right)\\
&+\frac{1}{2S\cdot d_{i_j}}\sum_{k=2}^{M_i+1}\left(\left(-(x_{i_j}-x_i)(y_{i_{k-1}}-y_i)+(x_{i_{k-1}}-x_i)(y_{i_j}-y_i)\right)d_{i_k}\widetilde{R}_{i_k}\right)\\
=& \frac{1}{2S\cdot d_{i_j}}\sum_{k=2}^{M_i}\left(\left((x_{i_j}-x_i)\left(y_{i_{k+1}}-y_{i_{k-1}}\right)-\left(y_{i_j}-y_i\right)\left(x_{i_{k+1}}-x_{i_{k-1}}\right)\right)d_{i_k}\widetilde{R}_{i_k}\right)\\
&+\left((x_{i_j}-x_i)(y_{i_2}-y_i)-(x_{i_2}-x_i)(y_{i_j}-y_i)\right)d_{i_2}\widetilde{R}_{i_1}\\
&+\left(-(x_{i_j}-x_i)(y_{i_{M_i}}-y_i)+(x_{i_{M_i}}-x_i)(y_{i_j}-y_i)\right)d_{i_{M_i+1}}\widetilde{R}_{i_{M_i+1}}
\end{aligned}
$$

As mentioned earlier, there are

$$
\begin{aligned}
x_{i_{M_i}} &= x_{i_0} = x_{i_{1-1}},\\
y_{i_{M_i}} &= y_{i_0} = y_{i_{1-1}},\\
d_{i_{M_i+1}} &= d_{i_{M_1}},\\
\widetilde{R}_{i_{M_i+1}} &= \widetilde{R}_{i_1}.
\end{aligned}
$$

Now we can get

$$
\mathbf{u}_{i_j}(\widetilde{\mathbf{p}}_i-\mathbf{p}_i)=\frac{1}{2S\cdot d_{i_j}}\sum_{k=1}^{M_i}\left(\left((x_{i_j}-x_i)\left(y_{i_{k+1}}-y_{i_{k-1}}\right)-\left(y_{i_j}-y_i\right)\left(x_{i_{k+1}}-x_{i_{k-1}}\right)\right)d_{i_k}\widetilde{R}_{i_k}\right). \tag{B.2}
$$

Let

$$
Q_j^k(\mathbf{x}_i,\mathbf{X})=\frac{1}{2S}\frac{d_{i_k}}{d_{i_j}}\left(\left(x_{i_j}-x_i\right)\left(y_{i_{k+1}}-y_{i_{k-1}}\right)-\left(y_{i_j}-y_i\right)\left(x_{i_{k+1}}-x_{i_{k-1}}\right)\right). \tag{B.3}
$$

Then, there is

$$\mathbf{u}_{i_j}(\widetilde{\mathbf{p}}_i - \mathbf{p}_i) = \sum_{k=1}^{M_i} Q_j^k(\mathbf{x}_i, \mathbf{X}) \cdot \widetilde{R}_{i_k}. \tag{B.4}$$

It is important to note that formula (B.4) is formula (3.17) and formula (B.3) is formula (3.18).

Appendix C.

According to formula (3.18) and (3.19), for any i, j, k, there is

$$Q_j^k(\mathbf{x}_i, \mathbf{X}) = \frac{1}{2S}\frac{d_{i_k}}{d_{i_j}}\left(\det(\begin{bmatrix}\overrightarrow{\mathbf{x}_i\mathbf{x}_{i_j}}\\ \overrightarrow{\mathbf{x}_i\mathbf{x}_{i_{k+1}}}\end{bmatrix}) - \det(\begin{bmatrix}\overrightarrow{\mathbf{x}_i\mathbf{x}_{i_j}}\\ \overrightarrow{\mathbf{x}_i\mathbf{x}_{i_{k-1}}}\end{bmatrix})\right). \tag{C.1}$$

A basic knowledge of analytic geometry gives that

$$\left|\det(\begin{bmatrix}\overrightarrow{\mathbf{x}_i\mathbf{x}_{i_j}}\\ \overrightarrow{\mathbf{x}_i\mathbf{x}_{i_{k+1}}}\end{bmatrix})\right| = 2 \cdot S(\triangle_{(\mathbf{x}_i, \mathbf{x}_{i_j}, \mathbf{x}_{i_{k+1}})}),$$
$$\left|\det(\begin{bmatrix}\overrightarrow{\mathbf{x}_i\mathbf{x}_{i_j}}\\ \overrightarrow{\mathbf{x}_i\mathbf{x}_{i_{k-1}}}\end{bmatrix})\right| = 2 \cdot S(\triangle_{(\mathbf{x}_i, \mathbf{x}_{i_j}, \mathbf{x}_{i_{k-1}})}),$$

here, $S(\triangle_{(\mathbf{x}_i, \mathbf{x}_{i_j}, \mathbf{x}_{i_{k+1}})})$ is the area of the triangle $\triangle_{(\mathbf{x}_i, \mathbf{x}_{i_j}, \mathbf{x}_{i_{k+1}})}$ and $S(\triangle_{(\mathbf{x}_i, \mathbf{x}_{i_j}, \mathbf{x}_{i_{k-1}})})$ is the area of the triangle $\triangle_{(\mathbf{x}_i, \mathbf{x}_{i_j}, \mathbf{x}_{i_{k-1}})}$.
It is easy to get that

$$\frac{1}{2}s_m{h_m}^2 \le S(\triangle_{(\mathbf{x}_i, \mathbf{x}_{i_j}, \mathbf{x}_{i_{k+1}})}) \le \frac{1}{2}{h_M}^2$$
$$\frac{1}{2}s_m{h_m}^2 \le S(\triangle_{(\mathbf{x}_i, \mathbf{x}_{i_j}, \mathbf{x}_{i_{k-1}})}) \le \frac{1}{2}{h_M}^2$$

According to (C.1) and (3.14), we know that $\forall i, j, k$, there is

$$\begin{aligned}|Q_j^k(\mathbf{x}_i, \mathbf{X})| &\le \frac{1}{2s_m{h_m}^2}\frac{h_M}{h_m}\left(2\frac{1}{2}{h_M}^2 + 2\frac{1}{2}{h_M}^2\right)\\ &\le \frac{K^3}{s_m}.\end{aligned} \tag{C.2}$$

In particular, if $j = k$,then there is

$$\begin{aligned}
Q_k^k(\mathbf{x}_i, \mathbf{X}) &= \frac{1}{2S}\frac{d_{i_k}}{d_{i_k}}\left(\det(\begin{bmatrix}\overrightarrow{\mathbf{x}_i\mathbf{x}_{i_k}} \\ \overrightarrow{\mathbf{x}_i\mathbf{x}_{i_{k+1}}}\end{bmatrix}) - \det(\begin{bmatrix}\overrightarrow{\mathbf{x}_i\mathbf{x}_{i_k}} \\ \overrightarrow{\mathbf{x}_i\mathbf{x}_{i_{k-1}}}\end{bmatrix})\right) \\
&= \frac{1}{2S}\left(\det(\begin{bmatrix}\overrightarrow{\mathbf{x}_i\mathbf{x}_{i_k}} \\ \overrightarrow{\mathbf{x}_i\mathbf{x}_{i_{k+1}}}\end{bmatrix}) + \det(\begin{bmatrix}\overrightarrow{\mathbf{x}_i\mathbf{x}_{i_{k-1}}} \\ \overrightarrow{\mathbf{x}_i\mathbf{x}_{i_k}}\end{bmatrix})\right) \\
&= \frac{1}{2S}\left(2S_{i_k} + 2S_{i_{k-1}}\right) \\
&= \frac{S_{i_k} + S_{i_{k-1}}}{S}.
\end{aligned} \tag{C.3}$$

We know that

$$s_m {h_m}^2 \le S_{i_k} + S_{i_{k-1}} \le {h_M}^2. \tag{C.4}$$

In this case, there are more than three triangles, so that there is

$$S - (S_{i_k} + S_{i_{k-1}}) \ge \frac{1}{2} s_m {h_m}^2. \tag{C.5}$$

Clearly,

$$Q_k^k(\mathbf{x}_i, \mathbf{X}) = \frac{S_{i_k} + S_{i_{k-1}}}{S - (S_{i_k} + S_{i_{k-1}}) + S_{i_k} + S_{i_{k-1}}}. \tag{C.6}$$

We consider the maximum of the function $f(x,y) = x/(x+y)$ on $[s_m {h_m}^2, {h_M}^2] \times [\frac{1}{2} s_m {h_m}^2, \infty)$.
There are

$$\begin{aligned}
\frac{\partial f}{\partial x} &= \frac{y}{(x+y)^2} > 0, \\
\frac{\partial f}{\partial y} &= \frac{-x}{(x+y)^2} < 0.
\end{aligned} \tag{C.7}$$

Clearly,

$$\max_{(x,y) \in [s_m {h_m}^2, {h_M}^2] \times [\frac{1}{2} s_m {h_m}^2, \infty)} f(x,y) = \frac{{h_M}^2}{{h_M}^2 + \frac{1}{2} s_m {h_m}^2}. \tag{C.8}$$

According to (C.6) and (C.8), for any i,k there is

$$0 < Q_k^k(\mathbf{x}_i, \mathbf{X}) \le \frac{{h_M}^2}{{h_M}^2 + \frac{1}{2} s_m {h_m}^2} \le \frac{2K^2}{2K^2 + s_m} < 1. \tag{C.9}$$

It is important to note that formula (C.2) is formula (3.20), formula (C.3) is formula (3.21) and formula (C.9) is formula (3.22).