Current Applied Science and Technology Vol. 22 No. 6 (November-December 2022)

Research article

A Modified Local Distance-weighted (MLD) Method of Interpolation and Its Numerical Performances for Large Scattered Datasets

Dusita Ritthison, Sunisa Tavaen and Sayan Kaennakham*

School of Mathematics, Institute of Science, Suranaree University of Technology, Nakhon Ratchasima, Thailand

Received: 30 June 2021, Revised: 26 January 2022, Accepted: 15 February 2022

DOI: 10.55003/cast.2022.06.22.006

Keywords

interpolation;
scattered datasets;
distance-weighted;
triangle-based blending;
radial basis function

Abstract

The purpose of this study was to propose a new interpolation scheme that was designed to remedy the shortcomings encountered in two popular interpolation methods; the triangle-based blending (TBB) method and the inverse distance weighted (IDW) method. At the same time, the proposed method combines their desirable aspects, which are the local nature and non-use of quadratic surface construction, making it comparatively less time-consuming and more independent of the global effect. Because of these properties, the new scheme was named the 'Modified Local Distance-Weighted (MLD)' method, and it was tested in detail with different sizes of datasets. The datasets involved in this investigation were of two types; uniformly and non-uniformly distributed. The performances were carefully monitored and assessed via several criteria; accuracy, sensitivity to parameters, CPU-time, storage requirement and ease of implementation. For comparative purposes, three alternative interpolation methods were simultaneously carried out, i.e. TBB, IDW and Radial Basis Function-Based (RBF) method. The investigation clearly revealed promising aspects of the proposed scheme where a good quality of results was anticipated, and CPU-time and storage were seen to have been significantly reduced. The research strongly indicates the benefits of the proposed method for larger sized datasets for real practical scientific and engineering uses in the future.

*Corresponding author: (+66) 44224747 Fax: (+66)44224649
E-mail: sayan_kk@g.sut.ac.th

1. Introduction

With great and obvious impacts on science and engineering applications in the fields of industry, economy, education, healthcare, environment and many more areas, interpolation and function approximations are known as one of the most crucial ingredients used in certain branches of applied mathematics. The interpolation problem starts with a set of discrete data $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}, \mathbf{x}_i \in \mathbb{R}^d$ where for each $\mathbf{x}_i$ there is its corresponding real value $y_i \in \mathbb{R}$. Then, the task is to construct a continuous function $\Phi(\mathrm{x}): \mathbb{R}^d \to \mathbb{R}$ such that;

$$\Phi(\mathbf{x}_i) = y_i . \tag{1}$$

When $d = 1$, this problem is traditionally approached by different schemes, yet they all obey the theory of interpolation. If $x_0, ..., x_d \in \mathbb{R}$ are distinct, then for any $y_i = f(x_i) \in \mathbb{R}$, there exists a unique polynomial $\Phi(\mathrm{x}): \mathbb{R}^d \to \mathbb{R}$ of degree $\leq N$ such that the above interpolation conditions are satisfied. Both development and implementation of interpolation methods have been receiving a great amount of attention from both scientists and engineers. Some recent studies include interpolation of small datasets in the sandstone hydrocarbon reservoirs [1], determination of the spatial distribution of nitrogen compounds in groundwater [2], investigation on interpolation with large datasets using radial basis functions [3] and application of the Modified Shepard's Method (MSM) [4] (see also references therein).

When it comes to real practical applications, several criteria must be taken into consideration for the method's validation. Included are:

Accuracy: This can be carried out using some appropriate error norms and measurements. Ideally, the higher accuracy, the better the method is claimed to be.
CPU-time: With less amount of time required for the computational process, a method would be more desirable.
Storage requirement: Each computation step involved in the algorithm should ideally take the least amount of storage space as possible.
User's Interference: A good interpolation method should be able to process entirely by itself, no human interruption or judgments should be involved.
Sensitivity to parameters: A small change in parameters embedded in the process should have as little effect as possible on the overall performance. Methods containing no parameters would be best in practice.
Ease of implementation: Once the desirable method has been proven with small test models, it should then be implementable for larger problems with no difficulties (in terms of both mathematical structures and programming/coding).

With all this in mind, the proposed interpolation method is constructed accordingly with the hope to optimize and compromise all the factors related. Section 2 provides the fundamental ideas of three of the other most popular choices of interpolation, and two of them remain the inspiration for the one being proposed in this work. Section 3 illustrates the methods' effectiveness through a series of numerical experiments in which the results are carefully validated using criteria previously mentioned. Finally, the main findings of the work are concluded and listed in Section 4.

2. Methodology

Before providing the formula's construction of the proposed interpolation, it is important to briefly revisit two popular choices of interpolation; the Inverse Distance Weighted (IDW) method and the Triangle Based Blending (TBB) method, based on which our proposed method was constructed.

2.1 Inverse Distance Weighted method (IDW)

This method is originally modified from the very first version invented by Shepard in 1968 [5] or known as 'Shepard's method'. For each known point $(x_i, y_i) \in \mathbb{R}^2$, we denote $f_i = z(x_i, y_i) \in \mathbb{R}$, for $i = 1, 2, ...N$, as its corresponding known value. The formula of this method, for a new location (x, y), takes the following form;

$$F(x,y) = \sum_{i=1}^{N} w_i(x,y) f_i \Big/ \sum_{i=1}^{N} w_i(x,y) \tag{2}$$

where w_i is now defined in a global manner, i.e. $w_i = w_i^{(Global)}$, as expressed below.

$$w_i^{(Global)} = d_i^{\mu} = \left[\left((x - x_i)^2 + (y - y_i)^2 \right)^{1/2} \right]^{\mu}. \tag{3}$$

Here, μ may be replaced by μ_i and could be different for each i.

The method is known to be global meaning that it takes on all interpolating nodes all over the domain for a calculation of a point. This normally results in far-away nodes having too much influence, where ideally only closer-ones should have a greater effect. Amongst several attempts designed to localize the method is one adopted by Barnhill [6] where the following 'local manner' form is adopted.

$$w_i^{(Local)}(x,y) = \left[\frac{(R - d_i)_+}{R d_i} \right]^2. \tag{4}$$

The idea is done by piecing together a parabolic segment with d_i^{-2} in such a way as to obtain a w_i which is zero outside some disk, say of given radius R, centered at (x, y) and by which $F(x, y)$ is still C^1 [6].

Another choice was to make use of the information about derivatives, either given or generated from the data, which was suggested by Shepard and resulted in an approximation of the form.

$$F(x,y) = \sum_{i=1}^{N} w_i(x,y) \left[f_i + \left(\frac{\partial f}{\partial x} \right)_i (x - x_i) + \left(\frac{\partial f}{\partial y} \right)_i (y - y_i) \right] \Big/ \sum_{i=1}^{N} w_i(x,y). \tag{5}$$

Where the more general form can be written as follows;

$$F(x,y)=\sum_{i=1}^{N} w_i(x,y)L_i f(x,y)\Big/\sum_{i=1}^{N} w_i(x,y), \tag{6}$$

where $L_i f$ approximates f such that $L_i f(x_i,y_i)=f_i$, the $L_i f$ are called 'nodal functions'. More localizing ideas can be found in Franke [7]. In summary, the performance of methods in this group is very dependent on the appropriate weight function $w_i(x,y)$ and for the sake of comparison, this work takes on $\mu=2$ and 3 only.

2.2 The Triangle Based Blending (TBB) method

The method was first proposed in 1976 by McLain [8]. The process begins with a construction of corresponding quadratic polynomial interpolant, $\hat{f}_i$, for each known point $(x_i,y_i)\in\square^2$, expressed as follows;

$$\hat{f}_i(x,y)=A_1x^2+A_2xy+A_3y^2+A_4x+A_5y+A_6. \tag{7}$$

This passes through the point (x_i,y_i) itself and its five nearest neighbors where $\hat{f}_i(x_k,y_k)=z(x_k,y_k)$ for $k=1,2,...,6$. It then yields a system of linear equations for each (x_i,y_i) expressed below.

$$\begin{bmatrix}\hat{f}_1\\ \hat{f}_2\\ \hat{f}_3\\ \hat{f}_4\\ \hat{f}_5\\ \hat{f}_6\end{bmatrix}=\begin{bmatrix}x_1^2 & x_1y_1 & y_1^2 & x_1 & y_1 & 1\\ x_2^2 & x_2y_2 & y_2^2 & x_2 & y_2 & 1\\ x_3^2 & x_3y_3 & y_3^2 & x_3 & y_3 & 1\\ x_4^2 & x_4y_4 & y_4^2 & x_4 & y_4 & 1\\ x_5^2 & x_5y_5 & y_5^2 & x_5 & y_5 & 1\\ x_6^2 & x_6y_6 & y_6^2 & x_6 & y_6 & 1\end{bmatrix}\begin{bmatrix}A_1\\ A_2\\ A_3\\ A_4\\ A_5\\ A_6\end{bmatrix}. \tag{8}$$

From here the coefficients A_k are expected to be available, provided that the system is well-posed.

Now, to find the interpolated value z at an arbitrary point (x,y), it starts with finding which triangle of the triangulation this point is located within. Then, the value z is determined at the weighted average of the values at the point of the three interpolating polynomials that correspond to the three triangle vertices [9], expressed as;

$$z=F(x,y)=w_1\hat{f}_1(x,y)+w_2\hat{f}_2(x,y)+w_3\hat{f}_3(x,y). \tag{9}$$

The weight w_i can be obtained in such a way that a continuous transition from one triangle to the next is ensured. For this purpose, the following simple expression of the weight is defined;

$$w_i=\frac{d_i^k}{\left(d_1^k+d_2^k+d_3^k\right)}. \tag{10}$$

Other variants of this method can also guarantee C^1 continuity across triangle boundaries, see more in McLain [8]. (A note of notation shall be taken here; in this work, we note k in the above equation as μ in all numerical test cases in Section 4).

2.3 The proposed Modified Local Distance-Weighted (MLD) method

It can be observed from the previous section that TBB preserves the local nature allowing only neighboring nodes to have an effect. The biggest disadvantage, nevertheless, is the need to construct a quadratic polynomial interpolant surface for every data point in the training datasets. This process takes a great amount of time. To remedy this unpleasant aspect, our method considers preserving the local nature by calculating only the nearest three points, say $\{(x_j, y_j)\}_{j=1}^{3}$ of data using their exact nodal values. The formula is as follows;

$$F(x,y) = \bar{w}_1 f_1(x_1, y_1) + \bar{w}_2 f_2(x_2, y_2) + \bar{w}_3 f_3(x_3, y_3) \tag{11}$$

And

$$\bar{w}_i = \left(\frac{(R - d_i)_+}{R d_i} \right)^2 \tag{12}$$

where R is the radius of influence about the node (x, y), and d_i is the distance between the node (x, y) and its i^{th} – nearest one.

At this point, the weight function $\bar{w}_i$ can be seen as that found in the IDW method. However, while a lot of complicated choices have been proposed and tested as mentioned in the previous section, in this work the following simple but robust form is proposed;

$$(R - d_i)_+ = \begin{cases} R - d_i & , \text{if } d_i < R \\ 0 & , \text{if } d_i \geq R. \end{cases} \tag{13}$$

Where d_i is the Euclidean norm: $d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}$. The radius being used here can represent a line segment in 1D, a circle in 2D, a sphere in 3D, or a hypersphere in the n-dimensional space. With this form being in the interpolation process, the local aspect is inherited and there is no need for building any sophisticated surfaces.

2.4 Radial Basis Function (RBF) method

For the sake of comparison and method validation, another popular methodology of interpolation called Radial Basis Functions (RBF) was also studied, applied and taken into consideration. As investigated and adopted by many researchers [10, 11], in RBFs, φ , are commonly found as multivariate functions whose values are dependent only on the distance from the origin. This means that $\varphi(\mathbf{x}) = \varphi(r) \in \mathbb{R}$ with $\mathbf{x} \in \mathbb{R}^n$ and $r \in \mathbb{R}$; or, in other words, on the distance between each pair of vectors in $\{\mathbf{x}_k\}_{k=1}^{N}$, $\varphi(\mathbf{x}_i - \mathbf{x}_j) = \varphi(r_{ij}) \in \mathbb{R}$, that can normally be defined as follows;

$$r_{ij} = \left\| \mathbf{x}_i - \mathbf{x}_j \right\|_2 = \sqrt{(x_i^{(1)} - x_j^{(1)})^2 + (x_i^{(2)} - x_j^{(2)})^2 + ... + (x_i^{(n)} - x_j^{(n)})^2} \ . \tag{14}$$

When the interpolation is tackled using a radial basis function, the interpolating function is simply written in the form of a linear combination of certain basic functions, expressed as follows;

$$z_i = F(\mathbf{x}_i) = \sum_{j=1}^{N} \alpha_j \varphi\left(\left\| \mathbf{x}_i - \mathbf{x}_j \right\|_2 \right). \tag{15}$$

for $i = 1, \ldots, N$. Once this condition is satisfied, it leads to the following matrix system.

$$\begin{bmatrix} \varphi\left(\left\|\mathbf{x}_1 - \mathbf{x}_1\right\|_2\right) & \varphi\left(\left\|\mathbf{x}_1 - \mathbf{x}_2\right\|_2\right) & \cdots & \varphi\left(\left\|\mathbf{x}_1 - \mathbf{x}_N\right\|_2\right) \\ \varphi\left(\left\|\mathbf{x}_2 - \mathbf{x}_1\right\|_2\right) & \varphi\left(\left\|\mathbf{x}_2 - \mathbf{x}_2\right\|_2\right) & \cdots & \varphi\left(\left\|\mathbf{x}_2 - \mathbf{x}_N\right\|_2\right) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi\left(\left\|\mathbf{x}_N - \mathbf{x}_1\right\|_2\right) & \varphi\left(\left\|\mathbf{x}_N - \mathbf{x}_2\right\|_2\right) & \cdots & \varphi\left(\left\|\mathbf{x}_N - \mathbf{x}_N\right\|_2\right) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix}. \tag{16}$$

It can then be rewritten as:

$$\mathbf{A}\boldsymbol{\alpha} = \mathbf{F} \tag{17}$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_N)^T$ is the corresponding coefficient vector, and $\mathbf{A} = \left[\mathbf{A}_{ij}\right] = \left[\varphi\left(\left\|\mathbf{x}_i - \mathbf{x}_j\right\|\right)\right]$, $i, j = 1, \ldots, N$ is the interpolation matrix with φ being the radial basis function under investigation.

In this work, one of the mostly common used forms, known as 'Gaussian-RBF' [12], was focused on, and the formula is expressed as follows;

$$\varphi(r) = \exp(-r^2 / 2\varepsilon^2). \tag{18}$$

where the parameter ε is the shape parameter, to be determined by the user. Choosing the value for this shape parameter remains a big challenge and in this work, the one proposed by Carlson and Foley [13] was considered. It starts by computing the least squares bivariate quadratic polynomial that fits the data $(\overline{x}_i, \overline{y}_i, \overline{z}_i)$ and denotes the quadratic by $q(\overline{x}_i, \overline{y}_i)$, as follows;

$$V = \sum_{i=1}^{N} \frac{\left(\overline{z}_i - q(\overline{x}_i, \overline{y}_i)\right)^2}{N} \tag{19}$$

by setting $\overline{x}_i = \dfrac{(x_i - x_{\min})}{(x_{\max} - x_{\min})}$, $\overline{y}_i = \dfrac{(y_i - y_{\min})}{(y_{\max} - y_{\min})}$ and $\overline{z}_i = \dfrac{(z_i - z_{\min})}{(z_{\max} - z_{\min})}$. The proposed form of shape parameter is then as follows;

$$\varepsilon = \frac{1}{1 + 120V}. \tag{20}$$

With some numerical support from our previous experiment on pattern recognition problems [14], this choice is once again utilized in this work.

3. Results and Discussion

To demonstrate the effectiveness of the proposed interpolation method, this section provides results obtained from several numerical experiments. For comparison purposes, three other well-known methods (TBB, IDW and RBF) were also conducted using the same datasets and for the sake of comparison, this work takes on $\mu = 2$ and 3 only. Datasets used in this study were split into 2 groups; training and testing sets. The training ones came in seven sizes (529, 1024, 2025, 5014, 10000, 20164 and 30276 nodes) with two distribution manners; uniform and non-uniform. Figure 1 shows node distribution manners under consideration of this investigation. Following the work nicely documented by Lazzaro and Montefusco [12], the testing dataset contains 50×50 uniformly distributed nodes.

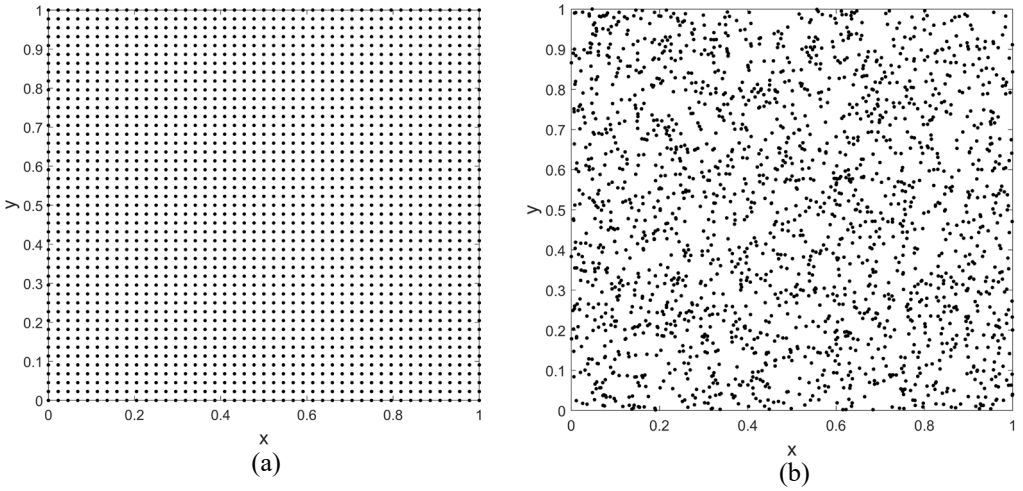

Figure 1. Node distribution manners using 45×45 nodes; (a) uniform and (b) non-uniform (random)

3.1 Error measurement norms

Since one of the criteria used to judge the effectiveness of the proposed interpolation scheme is accuracy, this process of this study was carried out using the following two error norms;

$$L_{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(z^{ext.}(x_i, y_i) - z^{appx.}(x_i, y_i)\right)^2 \tag{21}$$

and

$$L_{\infty} = \max_{1 \le i \le N}\left|z^{ext.}(x_i, y_i) - z^{appx.}(x_i, y_i)\right|. \tag{22}$$

All experiments were carried out on the same computer; Intel(R) Core (TM) i7-8750H CPU @ 2.20GHz 2.20 GHz with RAM 8.00 GB and 64-bit Operating System.

3.2 The Well-known Benchmarking Franke's Function

Known as one of the best test cases for benchmarking any newly- proposed scheme of interpolation, Franke's function given in Franke and Nielson [15], expressed below, was tackled throughout the study. Its surface in three dimensions is illustrated in Figure 2.

$$f(x,y) = 0.75\exp\left[-\frac{(9x-2)^2}{4}-\frac{(9y-2)^2}{4}\right]+0.75\exp\left[-\frac{(9x+1)^2}{49}-\frac{(9y+1)^2}{10}\right] +0.5\exp\left[-\frac{(9x-7)^2}{4}-\frac{(9y-3)^2}{4}\right]-0.2\exp\left[-(9x-4)^2-(9y-7)^2\right]. \quad (23)$$

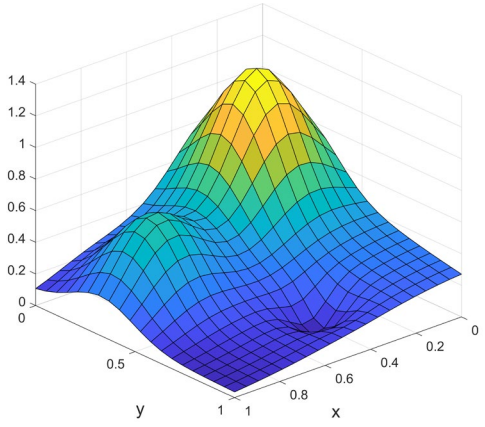

Figure 2. Franke's function's surface in three dimensions

3.3 Numerical results and general discussion

As far as the accuracy is concerned, it can clearly be seen from Tables 1-4 that the proposed MLD is a good candidate. Both L_∞ and L_{MSE} error norms reveal that MLD produces significantly better results than those obtained using RBF and slightly outperforms IDW (for both values of m), whereas the best model for all cases is actually TBB. This is not at all surprising, however, since the construction of quadratic surface as required by TBB is mathematically expected to yield a more accurate representation for each node. It should be noted also that every model tends to perform better with more support fed in from the training datasets, except for the RBF one. This could well be attributed to the fact that this method is well known to highly depend on the shape parameter, node distribution and invisibility of the interpolation matrix, making the use of RBF much more complicated and not easy to implement, particularly in the case of large datasets [12].

Regarding the aspect of CPU-time, Figure 3 illustrates clearly that MLD is comparatively a good choice whereas both TBB and RBF require a tremendous amount of time and storage for

larger sizes of datasets. These two models are seen, from both Figures, to require more than twice the computing time and storage of our proposed method, indicating complications when it comes to implementation for practical use. Another great strength of the proposed MLD method is clearly depicted in Figure 4 where it is seen to require a noticeably and significantly lower amount of storage.

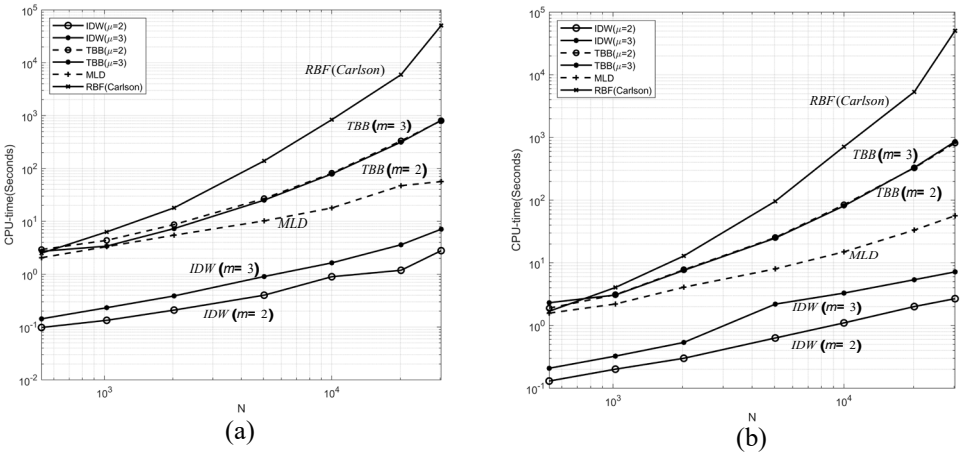

Figure 3. CPU-time used in each method plotted against increase of data sizes; (a) uniformly distributed dataset and (b) randomly distributed dataset

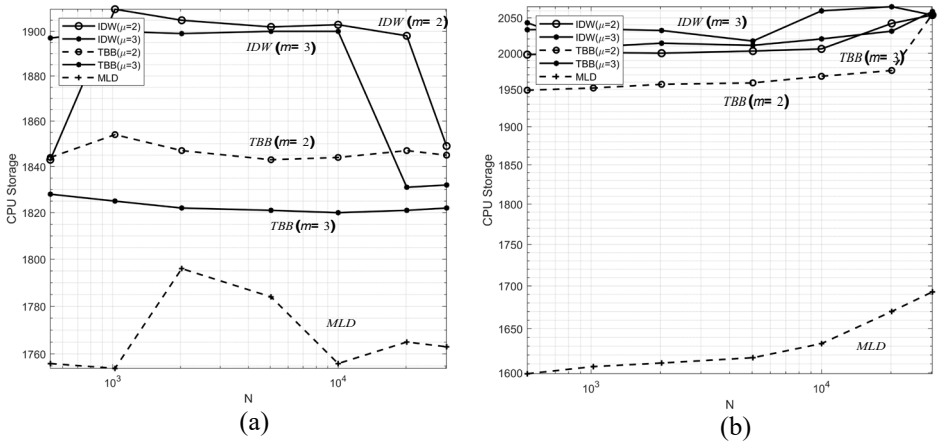

Figure 4. CPU-storage used in each method (except for RBF) plotted against increase of data sizes; (a) uniformly distributed dataset and (b) randomly distributed dataset

As for the sensitivity to parameter aspect, all three alternative models in this study (IDW, TBB and RBF) are clearly seen to highly depend on the particular parameter that each involves. A small change in these values can, evidently shown in all Tables 1-4, result in a noticeable change in the final accuracy. This automatically implies difficulty and uncertainty for application with real datasets in real-world applications. On the other hand, our proposed MLD requires no parameters as all factors can be fixed right from the beginning, meaning no user interference nor judgment is required. This aspect is crucial for future implementation as well.

Table 1. L_{MSE} - Comparison measured at dataset sizes with a uniform distribution of nodes

N	IDW (*m*=2)	IDW (*m*=3)	TBB (*m*=2)	TBB (*m*=3)	The Proposed MLD	RBF (Carlson)
529	2.89E-03	1.79E-04	5.17E-07	3.98E-07	1.64E-04	9.23E-04
1024	2.38E-03	9.16E-05	5.50E-08	7.39E-08	8.07E-05	1.50E-03
2025	1.97E-03	4.58E-05	7.17E-09	9.75E-09	3.94E-05	4.80E-03
5041	1.62E-03	1.72E-05	3.74E-10	5.15E-10	1.85E-05	1.34E-02
10000	1.39E-03	5.19E-06	5.87E-11	8.36E-11	5.48E-06	1.87E-02
20164	1.23E-03	4.41E-06	6.26E-12	8.74E-12	3.88E-06	6.02E-01
30276	1.15E-03	2.63E-06	1.59E-12	2.18E-12	2.33E-06	2.06E-01

Table 2. L_{MSE} - Comparison measured at dataset sizes with a random distribution of nodes

N	IDW (*m*=2)	IDW (*m*=3)	TBB (*m*=2)	TBB (*m*=3)	The Proposed MLD	RBF (Carlson)
529	4.32E-03	7.42E-04	4.67E-03	5.34E-03	5.75E-04	2.25E-01
1024	3.38E-03	3.50E-04	6.35E-05	7.83E-05	3.16E-04	6.09E-04
2025	2.38E-03	1.48E-04	1.02E-04	1.08E-04	1.41E-04	1.61E-02
5041	2.06E-03	7.73E-05	5.95E-06	5.91E-06	5.78E-05	2.07E-01
10000	1.66E-03	3.89E-05	5.43E-06	5.36E-06	2.70E-05	1.00E+01
20164	1.44E-03	1.90E-05	1.09E-07	1.35E-07	1.46E-05	8.61E-01
30276	1.43E-03	1.53E-05	1.59E-08	2.12E-08	1.01E-05	7.99E+00

Table 3. L_{∞} - Comparison measured at dataset sizes with a uniform distribution of nodes

N	IDW (*m*=2)	IDW (*m*=3)	TBB (*m*=2)	TBB (*m*=3)	The Proposed MLD	RBF (Carlson)
529	2.39E-01	7.16E-02	4.06E-03	3.63E-03	6.55E-02	1.25E-01
1024	2.15E-01	5.18E-02	1.57E-03	1.88E-03	4.50E-02	1.29E-01
2025	1.76E-01	3.30E-02	5.04E-04	6.06E-04	3.44E-02	2.60E-01
5041	1.67E-01	2.17E-02	1.48E-04	1.74E-04	2.80E-02	3.81E-01
10000	1.40E-01	9.24E-03	4.32E-05	5.26E-05	8.90E-03	6.74E-01
20164	1.51E-01	1.13E-02	1.65E-05	1.99E-05	1.23E-02	2.83E+00
30276	1.42E-01	6.91E-03	7.51E-06	9.06E-06	1.03E-02	1.76E+00

Table 4. L_{∞} - Comparison measured at dataset sizes with a random distribution of nodes

N	IDW (*m*=2)	IDW (*m*=3)	TBB (*m*=2)	TBB (*m*=3)	The Proposed MLD	RBF (Carlson)
529	3.55E-01	2.00E-01	1.65E+00	1.80E+00	1.25E-01	6.80E+00
1024	2.44E-01	1.01E-01	1.89E-01	2.25E-01	1.09E-01	1.57E-01
2025	2.05E-01	6.39E-02	3.43E-01	3.91E-01	6.27E-02	4.19E-01
5041	2.12E-01	6.27E-02	9.59E-02	9.27E-02	4.87E-02	2.20E+00
10000	1.92E-01	4.45E-02	1.12E-01	1.12E-01	3.04E-02	1.35E+01
20164	1.75E-01	3.35E-02	9.86E-03	1.07E-02	2.62E-02	4.43E+00
30276	1.79E-01	3.35E-02	3.63E-03	4.32E-03	2.42E-02	1.16E+01

In addition to all the promising results discussed so far, there are also a few related remarks to be addressed. Firstly, as evidently indicated by the final results, it is believed that the main reason for IDW having less accuracy is that this scheme proceeds in a global manner, that is it takes into

consideration all the points over the domain. This can result in far-away nodes having more effect on the overall interpolation itself. On the other hand, MLD takes it account only those nodes closer to itself, with higher impact on the interpolation process. Secondly, the main reason why only $\mu = 2$ and 3 were used is that this work aims to compare the final results against other investigations done using these same values. Nevertheless, one of the main goals of this work is to further investigate the effectiveness of other values of ' μ '. Lastly, when the number of points are fixed, i.e. 3, it is suspected that the size of the influence domain would still have great effect on the accuracy. This would be the result of the locations of those selected three points which may be farther or closer to the interpolating point. This topic is definitely a subject for future investigation.

4. Conclusions

In this work, a new method of interpolation was proposed. The scheme was designed to remedy the drawbacks of two popular interpolation methods; the triangle-based blending (TBB) method and the inverse distance weighted (IDW) method while simultaneously incorporating their beneficial characteristics. The proposed method was named the 'Modified Local Distance-Weighted (MLD)' method. Datasets with different sizes were numerically tackled and tested through several cases. The method's performances were validated and the main findings of the work are listed as follows.

Accuracy: MLD produces higher accuracy than IDW and RBF, but lower than TBB.
CPU-time and Storage Requirement: A significant reduction in CPU-time and storage was noticeable when the proposed MLD method was in use.
User's Interference and Sensitivity to Parameters: MLD contains no parameters and the algorithm performed completely by itself.
Ease of Implementation: With a much simple formula construction and coding, MLD clearly has advantages for implementation in future applications.

Apart from all the desirable features and advantages mentioned, MLD can also be extended to handle multivariate systems containing more than two independent valuables due to the flexibility of the radius of local support defined. This remains one of our future investigations.

References

[1] Malvić, T., Ivšinović, J., Velić, J. and Rajić, R., 2019. Interpolation of small datasets in the sandstone hydrocarbon reservoirs, case study of the Sava Depression, Croatia. *Geosciences,* 9(5), 201.

[2] Bronowicka-Mielniczuk, U., Mielniczuk, J., Obroślak, R. and Przystupa, W., 2019. A comparison of some interpolation techniques for determining spatial distribution of nitrogen compounds in groundwater. *International Journal of Environmental Research*, 13, 679-687.

[3] Smolik, M. and Skala, V., 2017. Large scattered data interpolation with radial basis functions and space subdivision. *Integrated Computer-Aided Engineering, IOS Press*, 25(1), 49-62, https://doi.org/10.3233/ICA-170556.

[4] Malvić, T., Ivšinović, J., Velić, J., Sremac, J. and Barudžija, U., 2020. Application of the Modified Shepard's Method (MSM): A case study with the Interpolation of Neogene Reservoir Variables in Northern Croatia. *Stats*, 3(1), 68-83.

[5] Shepard, D., 1968. A two-dimensional interpolation function for irregularly spaced data. *ACM '68: Proceedings of the 23rd ACM National Conference*, New York, United States, August 27-29, 1968, pp. 517-524.

[6] Barnhill, R.E., 1977. Representation and approximation of surfaces. In: John R. Rice, ed. *Mathematical Software III*. New York: Academic Press, pp. 69-120.
[7] Franke, R., 1982. Scattered data interpolation: tests of some methods, *Mathematics of Computation*, 38(157), 181-200.
[8] McLain, D.H., 1976. Two dimensional interpolation from random data. *The Computer Journal*, 19(2), 178-181.
[9] Amidror, I., 2002. Scattered data interpolation methods for electronic imaging systems: a survey. *Journal of Electronic Imaging*, 11(2), 157-176.
[10] Franke, C. and Schaback, R., 1998. Convergence order estimates of meshless collocation methods using radial basis functions. *Advances in Computational Mathematics*, 8(4), 381-399.
[11] Tavaen, S., Viriyapong, R. and Kaennakham, S., 2020. Performances of non-parameterised radial basis functions in pattern recognition applications. *Journal of Physics: Conference Series*, 1706, 012165, https://doi.org/10.1088/1742-6596/1706/1/012165.
[12] Lazzaro, D. and Montefusco, L.B., 2002. Radial basis functions for the multivariate interpolationof large scattered data sets. *Journal of Computational and Applied Mathematics*, 140(1-2), 521-536.
[13] Carlson, R.E. and Foley, T.A., 1991. The parameter R^2 in multiquadric interpolation. *Computers and Mathematics with Applications*, 21(9), 29-42.
[14] Tavaen, S. and Kaennakham, S., 2021. A comparison study on shape parameter selection in pattern recognition by Radial Basis Function neural networks. *Journal of Physics: Conference Series,* 1921, 012124, https://doi.org/10.1088/1742-6596/1921/1/012124.
[15] Franke, R. and Nielson, G., 1979. *Smooth Interpolation of Large Sets of Scattered Data*. Technical Report, Naval Postgraduate School, Monterey, California.