Machine Learning and Artificial Intelligence
J.-L. Kim (Ed.)

doi:10.3233/FAIA220428

Numerical Investigation on Three Local-Adaptive *k*-Point Multiquadric Neural Networks

Pichapop PAEWPOLSONG [a], Krittidej CHANTHAWARA [b], Narongdech DUNGKRATOKE [a] and Sayan KAENNAKHAM [a,1]
[a] *School of Mathematics, Institute of Science, Suranaree University of Technology, Nakhon Ratchasima, 30000, Thailand.*
[b] *Program of Mathematics, Faculty of Science, Ubon Ratchathani Rajabhat University, Ubon Ratchathani 34000, Thailand.*

Abstract. Under the architecture of a neural network, this work proposes and applies three multiquadric radial basis function (MQ-RBF) interpolation schemes; The Common Local Radial Basis Function Scheme (CLRBF), The Iterative Local Radial Basis Function Scheme (ILRBF), and The Radius Local Radial Basis Function Scheme (RLRBF). The schemes are designed to perform locally to overcome drawbacks normally encountered when using a global one. The famous Franke function in two dimensions is numerically tackled. It is revealed in this work that all three local methods outperform the traditional MQ interpolation in terms of both CPU-time and condition number, while the accuracy is overall acceptable, particularly when the number of nodes increases. This finding indicates their potential for dealing with bigger datasets and more complex problems.

Keywords. local interpolation, radial basis function, multiquadric neural networks

1. Introduction

Radial Basis Functions (RBFs), ϕ , are commonly found as multivariate functions whose values are dependent only on the distance from the origin. This means that $\phi(\mathbf{x}) = \phi(r) \in \mathbb{R}$ with $\mathbf{x} \in \mathbb{R}^n$ and $r \in \mathbb{R}$, in other words, on the distance from a point of a given set $\{\mathbf{x}_j\}$, and $\phi(\mathbf{x} - \mathbf{x}_j) = \phi(r_j) \in \mathbb{R}$. Here, r_j is the Euclidean distance. One of the most popular types of RBFs is the Multiquadric (MQ) whose general format is expressed below.

$$\phi(r,\varepsilon) = \sqrt{\varepsilon^2 + r^2} \tag{1}$$

With ε being the so-called 'shape parameter' to be determined by the user. Over decades, MQ-RBFs and other forms have been receiving a great amount of attention from scientists and engineers. Some successful applications are those for function

[1] Corresponding Author, Sayan KAENNAKHAM, School of Mathematics, Institute of Science, Suranaree University of Technology, Nakhon Ratchasima, 30000, Thailand. E-mail: sayan_kk@g.sut.ac.th.

approximation [1], for solving regulator equations [2], for classifying weblog dataset [3], for support vector machine classifiers [4], and for numerically solving partial differential equations [5, 6].

Despite the wide range of applications nicely documented in the literature, it is to be noted that most of them are based on a global manner of RBF interpolation. Proceeding it this way leads to several undesirable aspects such as the highly-dense nature of the interpolation matrix which could lead to ill-condition, high computational costs, and stability issues [7]. Inspired by our previous work [8], in this work, on the other hand, we focus on performing the MQ function interpolation in three local architectures of neural networks. For the sake of comparison, a global format is to be included in the experiments and all results obtained are recorded and compared against one another.

2. Mathematical methodology

The whole study can be seen as an extended version of that proposed by [7]. It starts with letting $\{x_i\}_{i=1}^{N}$ be a given large set of scattered data points (Interpolation points), and $\{\mathbf{z}_j\}_{j=1}^{N_t}$ be a set of evaluation points, as shown in Figure 1. Suppose the function values $\{f(\mathbf{x}_i)\}_{i=1}^{N}$ at the interpolation points are given; we try to use these known values to obtain the approximate values of $\{f(\mathbf{z}_j)\}_{j=1}^{N_t}$.

Let ϕ be a radial basis function. Choose all interpolation points in the local domain Ω_j as the centers of the basis functions. Using RBF interpolation on Ω_j , we have the following linear system.

$$\begin{bmatrix} \hat{f}(\mathbf{x}_1^{[i]}) \\ \hat{f}(\mathbf{x}_2^{[i]}) \\ \vdots \\ \hat{f}(\mathbf{x}_k^{[i]}) \end{bmatrix} = \begin{bmatrix} \phi(\|\mathbf{x}_1^{[i]}-\mathbf{x}_1^{[i]}\|) & \phi(\|\mathbf{x}_1^{[i]}-\mathbf{x}_2^{[i]}\|) & \cdots & \phi(\|\mathbf{x}_1^{[i]}-\mathbf{x}_k^{[i]}\|) \\ \phi(\|\mathbf{x}_2^{[i]}-\mathbf{x}_1^{[i]}\|) & \phi(\|\mathbf{x}_2^{[i]}-\mathbf{x}_2^{[i]}\|) & \cdots & \phi(\|\mathbf{x}_2^{[i]}-\mathbf{x}_k^{[i]}\|) \\ \vdots & \vdots & \vdots & \vdots \\ \phi(\|\mathbf{x}_k^{[i]}-\mathbf{x}_1^{[i]}\|) & \phi(\|\mathbf{x}_k^{[i]}-\mathbf{x}_2^{[i]}\|) & \cdots & \phi(\|\mathbf{x}_k^{[i]}-\mathbf{x}_k^{[i]}\|) \end{bmatrix} \begin{bmatrix} \alpha_1^{[i]} \\ \alpha_2^{[i]} \\ \vdots \\ \alpha_k^{[i]} \end{bmatrix}. \tag{2}$$

Which can be written in the matrix form as follows.

$$\hat{\mathbf{f}}_k = \mathbf{\Phi}_{k\times k}\boldsymbol{\alpha}^{[i]} \tag{3}$$

Thus,

$$\boldsymbol{\alpha}^{[i]} = \mathbf{\Phi}_{k\times k}^{-1}\hat{\mathbf{f}}_k \tag{4}$$

Therefore,

$$f(\mathbf{z}_i)=\sum_{j=1}^{k}\alpha_j^{[i]}\phi(\|\mathbf{z}_i-\mathbf{x}_j^{[i]}\|)=\left[\phi(\|\mathbf{z}_i-\mathbf{x}_1^{[i]}\|,\phi(\|\mathbf{z}_i-\mathbf{x}_2^{[i]}\|,\ldots,\phi(\|\mathbf{z}_i-\mathbf{x}_k^{[i]}\|\right]\boldsymbol{\alpha}^{[i]}$$
$$=\left[\phi(\|\mathbf{z}_i-\mathbf{x}_1^{[i]}\|,\phi(\|\mathbf{z}_i-\mathbf{x}_2^{[i]}\|,\ldots,\phi(\|\mathbf{z}_i-\mathbf{x}_k^{[i]}\|\right]\boldsymbol{\Phi}_{k\times k}^{-1}\hat{\mathbf{f}}_k. \quad (5)$$

The values of $\left\{f(\mathbf{z}_j)\right\}_{j=1}^{N_t}$ are now roughly determined.

From this point, the main idea is then combined with the sliding style of a local influent domain previously performed by [9]. It then leads to the formation of three local interpolation schemes investigated in this work.

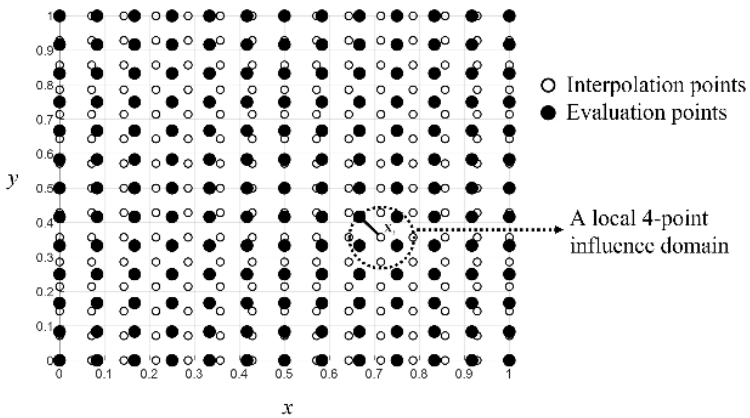

Figure 1. An example of a local domain with the interpolation and evaluation points.

2.1. The Common Local Radial Basis Function Scheme (CLRBF)

This method starts with choosing the k interpolation points that are closest to $\mathbf{z}_j$ to generate the local domain Ω_j. It then assigned these interpolation points in the local domain Ω_j as the centers of the basis functions. Then carry out the aforementioned computational procedure.

2.2. The Iterative Local Radial Basis Function Scheme (ILRBF)

Similarly to CLRBF, this approach also produces local domains. The approximate values of $f(\mathbf{z}_j)$ are computed as usual in each local domain Ω_j. However, the main difference is that once $f(\mathbf{z}_j)$ of an preciously-unknown point $\mathbf{z}_j$ has been approximated in Ω_j, this point with its values is then treated as a centre (interpolation point) for the next adjacent local domain.

2.3. The Radius Local Radial Basis Function Scheme (RLRBF)

This approach builds a radius r_j around $\mathbf{z}_j$ in each local domain Ω_j. The k closest interpolation points from $\mathbf{z}_j$ are chosen, and the distance of the point that is farthest from $\mathbf{z}_j$ is used to get the radius for each local domain. All interpolation points within the radius r_j can be used as the basis function centers to approximate the values of $f\left(\mathbf{z}_j\right)$. Then perform the usual approximation of $f\left(\mathbf{z}_j\right)$ values. Likewise to the ILRBF, when approximate values of $f\left(\mathbf{z}_j\right)$ are obtained, $\mathbf{z}_j$ is moved into the set of interpolation points.

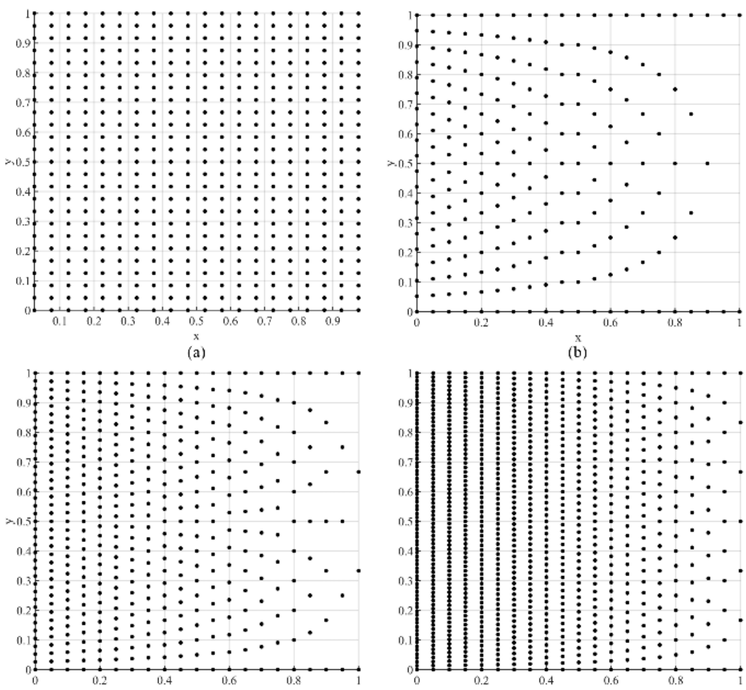

Figure 2. Node distributions involved in the experiment; (a) the fixed 500 evaluation points, (b) Level-1 with 222 nodes, (c) Level-2 with 444 nodes, and (d) Level-3 with 887 nodes.

3. Numerical experiments and results

Three levels of interpolation points were generated (Level-1, 2, 3) together with 500 evaluation points, depicted in Figure 2., for all numerical experiments. For the sake of comparison, apart from the three local schemes; CLRBF, ILRBF, and RLRBF, the traditional global scheme (noted as TGS) was also included in the experiments. To validate the performance of the local schemes under investigation, the famous twodimensional Franke's function is numerically tackled. This function is mathematically expressed as follows, [10] (with its surface profile shown in Figure 3(a).)

$$F(x,y) = 0.75\exp\left[-\frac{(9x-2)^2}{4}-\frac{(9y-2)^2}{4}\right]+0.75\exp\left[-\frac{(9x+1)^2}{49}-\frac{(9y+1)^2}{10}\right]$$
$$0.5\exp\left[-\frac{(9x-7)^2}{4}-\frac{(9y-3)^2}{4}\right]-0.2\exp\left[-(9x-4)^2-(9y-7)^2\right] \quad (6)$$

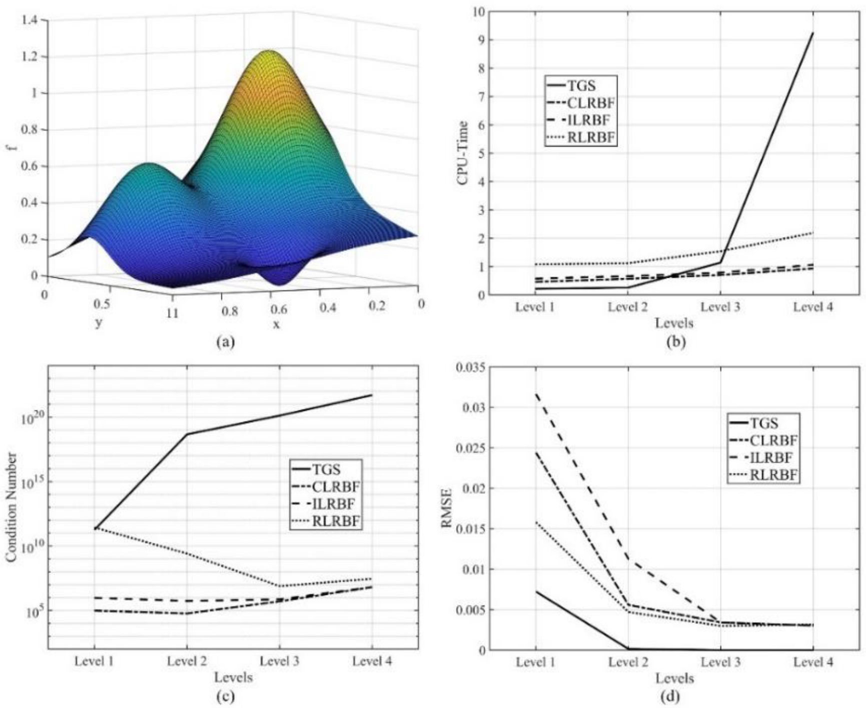

Figure 3. (a) Surface profile of the Franke's function, (b) CPU time spent for each computation involved, (c) the condition number of the interpolation matrix/matrices, and (d) the RMSE for each method.

For this preliminary experiment of this path, the value of k was fixed to 5 for all local schemes and for the whole experiment. Due to the limitation of space, the number of evaluation nodes was fixed at 500. Since it is well-known that there is no such thing as "an optimal shape" for every problem, this numerical investigation started with the attempt to identify a reasonably good shape parameter, ε . Table 1 provides information of comparatively good accuracy obtained using a wide range of ε for all schemes. With this in mind, $\varepsilon = 0.301$ was chosen to be used for the rest of the experiments.

Figure 3 (b), 3(c), and 3(d) illustrate respectively the CPU time spent, the mean condition number of the interpolation matrices, and the root mean square error (RMSE). Note that Level-4 containing 1,774 nodes was added to the experiment to make it clearer for trend analysis. It is clearly shown that with more nodes involved, both condition number and CPU time increased significantly for the global method when compared to the three local schemes, while RMSEs fell in a reasonably acceptable range. This strongly indicates the robust use of the proposed local schemes for more complex applications of the MQ neural network.

Table 1. The range of ε values that give reasonably good accuracy for each method.

Methods	$\varepsilon = 0.201$	$\varepsilon = 0.251$	$\varepsilon = 0.301$	$\varepsilon = 0.351$	$\varepsilon = 0.401$
TGS	1.02E-02	8.60E-03	7.23E-03	6.80E-03	8.08E-03
CLRBF	2.59E-02	2.48E-02	2.44E-02	2.43E-02	2.44E-02
ILRBF	3.86E-02	3.43E-02	3.16E-02	2.99E-02	2.88E-02
RLRBF	1.33E-02	1.62E-02	1.59E-02	4.76E-02	1.49E-01

4. Conclusions

With the shortcomings normally encountered when applying a global RBF neural network in an interpolation application, this work proposes and applies three local MQ-RBF schemes with the suitable shape parameter achieved in an 'ad-hoc' manner. The famous Franke function is then used to test the effectiveness of the methods with the results being compared against the global one. It is discovered in this work that the local schemes are able to produce reasonably good accuracy while keeping both the condition number and CPU time comparatively low. This strongly reveals their potentials for bigger and more complex domains which are the next steps of the study. Additionally, to strengthen the proposed methods, it is our future aim to compare them with some other global schemes.

Acknowledgement

This work was supported by (i) Suranaree University of Technology (SUT), (ii) Thailand Science Research Innovation (TSRI), and (iii) National Science, Research and Innovation Fund (NSRF) (NRIIS number 160336).

References

[1] Bellil W, Amar CB, Alimi AM. Comparison between beta wavelets neural networks, RBF neural networks and polynomial approximation for 1D, 2Dfunctions approximation. International Journal of Aerospace and Mechanical Engineering. 2008;2(1):189-194.

[2] Zhou G, Wang C, Su W. Nonlinear output regulation based on RBF neural network approximation. In Proceedings of International Conference on Control and Automation; 2005. p. 679-684.

[3] Dash CSK, Behera AK, Pandia MK, Dehuri S. Neural networks training based on differential evolution in radial basis function networks for classification of web logs. In Proceedings of International Conference on Distributed Computing and Internet Technology; 2013. p. 183-194.

[4] Thurnhofer-Hemsi K, López-Rubio E, Molina-Cabello MA, Najarian K. Radial basis function kernel optimization for Support Vector Machine classifiers. Preprint of the IEEE for possible publication. 2020.

[5] Kaennakham S, Chuathong N. Numerical solution to coupled burgers' equations by gaussian-based hermite collocation scheme. Journal of Applied and Mathematics. 2018:18.

[6] Kaennakham S, Chanthawara K. Numerical solution to nonlinear transient coupled-PDE by the modified multiquadric meshfree method. Romanian Journal of Physics. 2021;66(108):1-14.

[7] Yao G, Duo J, Chen CS. Implicit local radial basis function interpolations based on function values. Applied Mathematics and Computation. 2015;265:91-102.

[8] Ritthison D, Tavaen S, Kaennakham S. A modified local distance-weighted (MLD) method of interpolation and its numerical performances for large scattered datasets. Current Applied Science And Technology. 2022.

[9] Uhlir K, Skala V. Reconstruction of damaged images using radial basis functions. 13th European Signal Processing Conference; 2005. p. 1-4.

[10] Franke R. Scattered data interpolation: tests of some methods. Mathematics of computation. 1982; 38(157):181-200.