Multilevel Adaptive Radial Basis Function Approximation using Error Indicators

Thesis submitted for the degree of

Doctor of Philosophy

at the

University of Leicester

by

Qi Zhang

Department of Mathematics

University of Leicester

September 2016

Abstract

Multilevel Adaptive Radial Basis Function Approximation using Error Indicators

Qi Zhang

In some approximation problems, sampling from the target function can be both expensive and time-consuming. It would be convenient to have a method for indicating where the approximation quality is poor, so that generation of new data provides the user with greater accuracy where needed.

In this thesis, the author describes a new adaptive algorithm for Radial Basis Function (RBF) interpolation which aims to assess the local approximation quality and adds or removes points as required to improve the error in the specified region. For a multiquadric and Gaussian approximation, one has the flexibility of a shape parameter which one can use to keep the condition number of the interpolation matrix to a moderate size. In this adaptive error indicator (AEI) method, an adaptive shape parameter is applied.

Numerical results for test functions which appear in the literature are given for one, two, and three dimensions, to show that this method performs well. A turbine blade design problem form GE Power (Rugby, UK) is considered and the AEI method is applied to this problem.

Moreover, a new multilevel approximation scheme is introduced in this thesis by coupling it with the adaptive error indicator. Preliminary numerical results from this Multilevel Adaptive Error Indicator (MAEI) approximation method are shown. These indicate that the MAEI is able to express the target function well. Moreover, it provides a highly efficient sampling.

Acknowledgements

During these three years and a half of my Ph.D. life, there has been happiness, hesitations, excitements, frustrations and much more. I has learnt a lot from this precious experiences and would like to express my sincere gratitude to many people; people helped me to finish my Ph.D. degree and helped me to achieved something if I had.

First and foremost, I would like to express my deepest gratitude to my parents. Like the majority of Chinese people, I never took the chance to thank my parents properly. I would like to salute and express my gratitude to my caring and loving parents, Mr Minghua Zhang and Mrs Jinping Ma, for their fostering, support, encouragement and patient since I was a child. I could not finish my degree without their steady stream of support in many ways. I would like to dedicate this thesis to my parents.

Secondly, I would like to express utmost gratitude to my supervisor, instructor and mentor, Prof. Jeremy Levesley. Prof. Jeremy Levesley had provided continuous support, patience, motivation and immense helpful advice of my Ph.D. study and related research. Not only in the academic perspective, but also in many fields, Prof. Jeremy Levesley provided many wise pieces of advice for me. I could not think of a person whom could be a better tutor for me than Prof. Jeremy Levesley.

Thirdly, I would like to thanks my peer Ph.D. students; it was a great pleasure to work with them. I would like to thank especially Mr Zhaonan Dong, known as Peter Dong, and Mr Yangzhang Zhao. Peter was always ready to deliver helpful and inspiring bits of advice wholeheartedly at request since we are aware each other. To some extents, Yangzhang is my partner in research, and he is helpful and motivated for all the time. It was wonderful to have Peter and Yangzhang as my close friends.

Fourthly, I would like to thank Prof. Emmanuil (Manolis) Georgoulis as my seconder advisor, and he has been a great source of learning for me and despite his busy schedule, he always managed time for my research. I am also grateful to all the staff members of the mathematics department and the administration of the University for their help and making so many things easy for me.

Last but not least, I would like to thank GE Power (Rugby, UK) for their financial support and technical support. The University of Leicester has a close cooperation with GE Power (Rugby, UK), which provides great opportunities for researchers to communicate and to get inspired. I would thank especially Dr. Andrew Pike and Dr. Sarah Davis for their encouragement and support.

Contents

List of Figures

List of Tables

Abbreviations

AEI = Adaptive Error Indicator

ARBF = Adaptive Radial Basis Function

CFD = Computational Fluid Dynamic

KdV = KortewegâĂŞde Vries

LOO = Leave One Out

MAEI = Multilevel Adaptive Error Indicator

MQ = Multiquadric

RBF = Radial Basis Function

RS = Residual Sub-sampling

VSK = Variably Scaled Kernels

Chapter 1

Introduction

1.1 Background

The multivariate interpolation problem appears in many branches of science research and engineering. Let $\mathbf{X} \in \mathbb{R}^d$ be a discrete set, where $d > 1$ with the real number $\{f_{\mathbf{x}}\}_{\mathbf{x}\in\mathbf{X}}$. The task is to construct a continuous or sufficiently differentiable function $S : \mathbb{R}^d \to \mathbb{R}$ such that

$$S(\mathbf{x}) = f_{\mathbf{x}}, \quad \mathbf{x} \in \mathbf{X}, \tag{1.1}$$

and one could say that the S interpolates the data $\{(\mathbf{x}, f_{\mathbf{x}}), \mathbf{x} \in \mathbf{X}\}$. Interpolants could be highly useful. One may need to approximate a function whose values are known only at the given points, that is one is ignorant about function's behaviour outside $\mathbf{X}$. Otherwise, the target function might be far too expensive to evaluate at a large number of points. In this case the aim is to choose an interpolant that is cheap to compute, which could provide as much as needed accuracy at low computational cost. Then one can use the interpolant, the alternation of the target function, for other purposes, for example, for calculating the derivative of the target function.

Another application of interpolation could be data compression, where the initial data $\{(\mathbf{x}, f_{\mathbf{x}}), \mathbf{x} \in \tilde{\mathbf{X}}\}$ exceeds the available storage capacity in certain algorithms.

In this case, one needs to determine a subset of $\tilde{\mathbf{X}}$, $\mathbf{X}$ and construct an interpolant by it, then using this interplant to approximate the remaining values.

It is crucial to notice that in most cases $\mathbf{X}$ will consist of scattered points, in other words its elements can be in irregular locations. Thus interpolation algorithms that can cope with mesh-free data are needed. A mesh free method is a numerical method used to construct a system of algebraic equations for the whole domain of the approximation problem without using a pre-decided mesh for the domain and/or boundary discretization. Mesh generation can be one of the most time-consuming stages of any mesh-based numerical approximation methods.

There are many applications of multivariate interpolation, but the author prefer to treat a particular application in some detail rather than provide a list. Therefore the author consider the following interesting example of a turbine cascade case which was originally studied in [74] by Perdichizzi.

In electric power generation, the operating condition for a turbine could typically comes a wide envelope. The energy demand mainly determines the turbine load. Figure 1.1 shows a multi-stage turbine inside a BrayTon cycle power generator. The turbine output power is determined by the shaft rotational speed and axial flow rate, which changes the blade incidence angle and the expansion ratio [10].

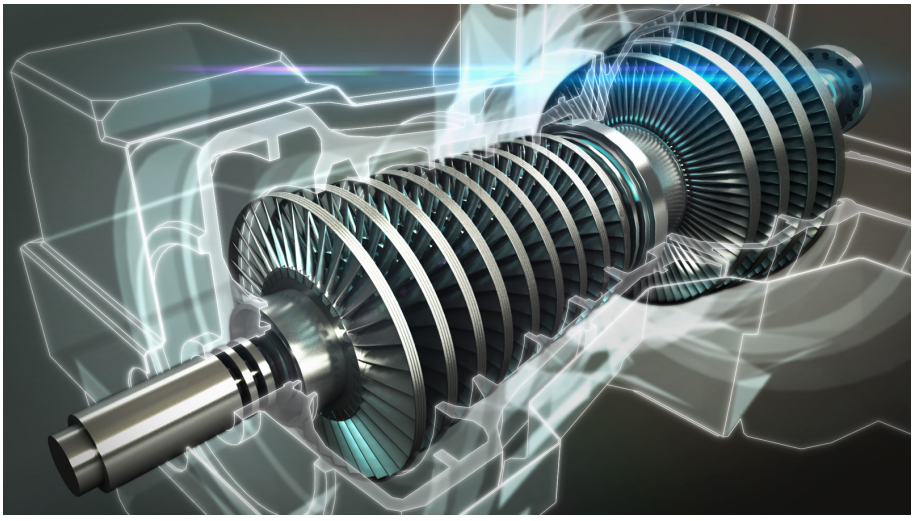

FIGURE 1.1: Schematic of an multi-stage axial compressor turbine assembly inside a power generator.

Furthermore, different pitch-chord ratios are commonly used in the various stage of axial turbines with the same profile [74]. The pitch-chord ratio is key dimension of the blade in a axial turbine. Figure 1.2 [74] shows could see two blades on the cascade plant, and the ratio s/c is defined as the pitch-chord ratio.

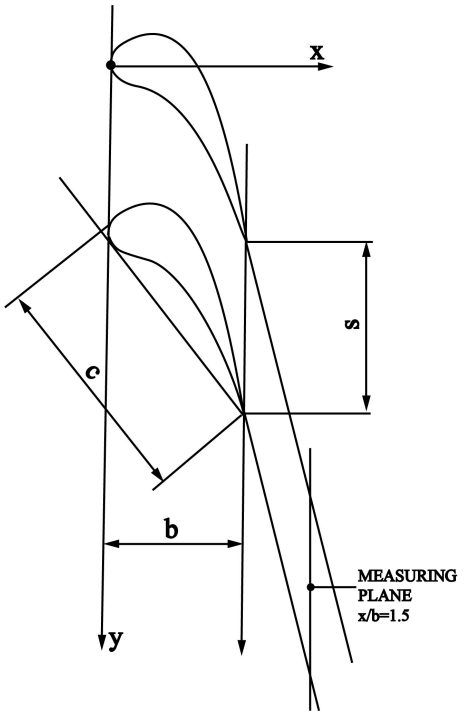

FIGURE 1.2: Dimensions of blades in the cascade plant of an axial turbine.

Perdichizzi and Dossena [74] states that "in the aerodynamic development of a multistage turbine, the awareness of blade row performance in terms of loss of outlet total pressure and secondary flow angle distribution is of extreme importance, both for optimizing the design of the turbine and predicting the overall efficiency at part loads for off-design conditions". The information which GE Power (Rugby, UK) is interested in, namely the change in stage efficiency with the operating conditions, can be obtained by experiment and/or by computational fluid dynamic (CFD). However, one single scattered data sample generated by this approach may take many hours.

Therefore, the intention is to generated an interpolated function that describes the relationship between profile efficiency (profile losses) and incidence angle at different flow condition and pitch-chord settings. For more information about the turbine efficiency, the author refers to [10]. In the study [74], the pitch-chord ratio

keep constant in all the cases. There are three varying input variables that define the design space, these three variables relate to the operating conditions of the machine and are :

- Incidence angle, α_1.
- Inlet total pressure, Po_1.
- Fluid viscosity, ν.

The objective of this study is to find the optimal profile efficiency of this turbine profile, in order words, maximizing the efficiency coefficient, which it is related to the profile losses. In particular, the lower the loss coefficient the higher the efficiency. In general, the loss coefficient function $L(\alpha_1, Po_1, \nu)$ describes the efficiency of this particular turbine.

Specifically, let $(\alpha_j, Po_{1_j}, \nu_j)_{j=1}^n$ be the test conditions in one situation, and let the corresponding observed loss coefficient be $L_{j=1}^n$. Function $S : \mathbb{R}^3 \to \mathbb{R}$ is needed such that

$$S(\alpha_j, Po_{1_j}, \nu_j) = L_j \quad for \quad j = 1, 2, \cdots, n. \tag{1.2}$$

Thence one sees the scattered data interpolation problem comes up spontaneously as the one attempts to construct an approximation model for this turbine case which describes its loss characteristic.

There are some multivariate approximation schemes. They are polynomial interpolation, tensor product methods, multivariate splines and finite element methods. For a thorough overview of these methods, the author refer the reader to de Boor [13], Franke [37] and Hayes [44]. Moreover, it is interesting to see the comparison among these methods with the radial basis function interpolation. The report of Franke [36] is designed for this comparison purpose; it has some numerical examples utilising some methods, including radial basis function and has provided numerical results that radial basis function interpolation provides greater accuracy than given comparisons when interpolating scattered data.

1.2 Radial Basis Function

Radial basis function (RBF) methods are not bound together with a grid or a mesh. This fall into the category of mesh free methods. A RBF approximation takes the form:

$$S(\bar{\mathbf{x}}) = \sum_{\mathbf{x}\in\mathbf{X}} \alpha_{\mathbf{x}} \phi(||\bar{\mathbf{x}} - \mathbf{x}||), \quad \bar{\mathbf{x}} \in \mathbb{R}^d, \tag{1.3}$$

where $\phi : [0, \infty) \to \mathbb{R}$ is a particular univariate function and the coefficients $(\alpha_{\mathbf{x}})_{\mathbf{x}\in\mathbf{X}}$ are real numbers. There is no restriction on the norm $||.||$ due to the method itself and the user of the RBF interpolation can freely define the norm. However, it is conventional to apply the Euclidean norm. Consequently, the approximation S is a linear combination of univariate functions, which the univariate function ϕ translates the radial distance $||\bar{\mathbf{x}}-\mathbf{x}||$ into a real number with respect to the given norm. For more details about RBF interpolation, please look in Chapter 2.

The author thinks that one of most attractive characteristics of the radial basis function method is the uniqueness of the interpolant is often exists with rather less strict conditions on the scattered data set. As straightforward as one could have, the only restraints are there at least two scattered points and they are distinct to each other. This condition is necessary for the unique interpolant.

Beside the above mentioned advantages that RBF has, it also has other advantages [8, 19, 85]. Due to these advantages, the application of RBF methods have been prosperous in last decades, Pena list some RBF applications (Table 1) in [73]. Generally speaking, the desirable features of RBF methods are high-order accuracy, easy implementation, mesh free and easy application in high dimension. Below, the author would like develop and support this argument by summarising some previous study and applications.

- **High-order accuracy**

  In [36], it numerically states the good accuracy for interpolation cases. Moreover, Larson and Fornberg [31] state that the RBF based method is more accurate than the standard second-order finite difference method and the

Fourier-Chebyshev pseudospectral method when dealing with elliptic problems. Buhmann et al. [65] and Madych et al. [7] proved that infinitely smooth RBF deliver spectral order of convergence which is more swift that any polynomial order.

- **Easy application in high dimension**

  Cecil et al. [11] demonstrated the example of solving the Hamilton Jacobi equation up to 4 dimensions by using RBF. In [15], Levesley et al. propose a multilevel algorithm based on directionally scaled tensor-product Gaussian kernels on structured sparse grids for interpolation of high-dimensional functions up to 10 dimensions. This algorithm is based on [38] which developed by Georgoulis, Levesley and Subhan.

- **Easy implementation**

  In [73], Pena demonstrates this point by providing a recipe-like algorithm for pricing different financial options. It also shows that Greek (derivative of options) are easy to calculate due to the form of the RBF interpolant. The form of RBF interpolant, the summation of univariate function ϕ, and the input of ϕ, the norm of two scattered points $||\bar{\mathbf{x}} - \mathbf{x}||$, together make it easy to implement.

- **Mesh free**

  RBF methods (interpolation, collocation, et cetera) only require a scattered data set which does not require the specially designed mesh or grid which contributes to easy implement and easy application in high dimension. This feature make RBF method capable dealing with American option pricing [22, 55] and the medical case which measures a tumour [2] where generating pre-designed mesh is arduous. However, there could be some loose "mesh" when fast evaluation is needed. Also, mesh free does not meaning whatever scattered data sample could provide enough accuracy, especially in high dimensional cases. Some certain requirements for the scattered data set can help contributing approximation accuracy. In Chapter 4, it shows a loosely determined mesh which is totally decided by the algorithm and the error indicator together.

In this thesis, the multiquadric (MQ) RBF is a mainly used raidal basis function. It is the first type of radial basis function that was introduced by Hardy in 1968 and presented in [42]. Hardly also summarised the development process of MQ method over 1968 to 1988 in [43]. This MQ interpolation scheme has gained few attention until 1979. Then a study conducted by Franke [34] and summarised in a report [37] that concluded that the MQ method is the best method for scattered data interpolation problem and he speculates the system matrix of this method is invertible and that this method is well-posed. By that time, a theoretical consolidation still did not exist. Then Micchelli [69] proved that the system matrix of the MQ method, and of also many other RBF methods, was invertible.

The first time that MQ RBF was applied to solve PDE problems was by Kansa [53, 54]. After that, utilising MQ in solving PDE problem become common [12, 24, 59, 79]. The MQ method for solving PDEs and especially for the Black-Scholes equations have been investigated in [47, 48, 49] by Hon et cetera. In [79], Sarra and Kansa summaries the methods and properties for MQ methods. For another commonly used RBF, the Gaussian RBF, please see [20, 30, 46].

1.3 The RBF based adaptive methods

Having above mentioned applications and advantages of radial basis function methods, one should notice that the locations or coordinates of the scattered data points are crucial. Both the approximation quality (the result quality) and stability mainly depends on the distribution of scattered points $\mathbf{X}$. In order to have good approximation quality and stability, many methods have been constructed to have the near optimal scattered points for the radial basis function interpolations. These methods could be called adaptive methods. Their aims are choosing the scattered data set $\mathbf{X}$ by various criteria. Much research [27, 28, 51, 66, 67, 87] has been done on developing useful adaptive schemes and in [40] the authors summarise them and classify them into different categories.

Based on the theoretical study [81] on the error estimates and condition number of RBF interpolation and others numerical experiments, reaching a balance between

stability and approximation accuracy is mostly the consequence of having a evenly distributed or near evenly distributed scattered points in the domain, which the author is interested in. In order to measure the uniformity of data $\mathbf{X}$ in the domain Ω, the ratio of the fill distance and separation distance is used, which is defined :

$$\rho_{\mathbf{X},\Omega} = \frac{q_{\mathbf{X}}}{h_{\mathbf{X},\Omega}}, \tag{1.4}$$

where $q_{\mathbf{X}}$ is the separation distance and $h_{\mathbf{X},\Omega}$ is the fill distance; for their definition please see Chapter 2. A thinning algorithm is one of the adaptive algorithms that removes points from scattered data set $\mathbf{X}$. The objective of the thinning algorithm is to achieve a subset of $\mathbf{X}$ that has a certain size and the subset should have good uniformity. The algorithm below [26] generates a sequence of subsets,

$$\mathbf{X}_1 \subset \mathbf{X}_2 \subset \cdots \subset \mathbf{X}_{n-1} \subset \mathbf{X}_n = \mathbf{X};$$

each of the subsets $\mathbf{X}_i$ is generated by the standard that removing a point $\mathbf{x}$ is feasible if and only if the removed would maximizes the measurement of uniformity $\rho_{\mathbf{X},\Omega}$. The algorithm is the following

1. Given a scattered data set $\mathbf{X}$, let $\mathbf{X}_n = \mathbf{X}, i = n$.
2. Choose a removable point $\mathbf{x} \in \mathbf{X}_i$.
3. Remove the point, $\mathbf{X}_{i-1} = \mathbf{X}_i \setminus \{\mathbf{x}\}$.
4. Set $i = i - 1$.
5. Stop if $i = 2$, otherwise return to Step 2.

In [28], the dual algorithm of the above thinning algorithm is proposed, it is an adding algorithm. This adding algorithm iteratively adds a chosen point $\mathbf{x}$ into $\mathbf{X}_i$. The adding algorithm is below.

1. Set $\mathbf{X}_0 = \emptyset, i = 0$.
2. Choose a point suitable for inserting $\mathbf{x} \in \mathbf{X} \setminus \mathbf{X}_i$.

3. Insert the point, $\mathbf{X}_{i+1} = \mathbf{X}_i \cup \{\mathbf{x}\}$.

4. Set $i = i + 1$.

5. Stop if $i = n - 1$, otherwise return to Step 2.

De Marchi, Schaback and Wendland [66, 67] have constructed a geometric greedy algorithm which selects the near optimal scattered data set for RBF interpolation. The geometric greedy algorithm is :

1. Let $\Omega \in \mathbb{R}^d$ be a compact set, and let $\mathbf{X}_1 = \{\mathbf{x}_1\}$, where $\mathbf{x}_1$ is at the boundary of Ω. Note, there is total n elements in set $\mathbf{X}_n$, that is $\mathbf{X}_n = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$.

2. For $n \geq 1$, choose $\mathbf{x}_{n+1} \in \Omega \setminus \mathbf{X}_n$ which has maximized the distance to $\mathbf{X}_n$. That is,

$$\mathbf{x}_{n+1} = \arg\max\{d(\mathbf{x}, \mathbf{X}_n) : \mathbf{x} \in \Omega \setminus \mathbf{X}_n\},$$

   where $d(\mathbf{y}, \mathbf{X}_n) = \max\{||\mathbf{y} - \mathbf{x}||_2 : \mathbf{x} \in \mathbf{X}_n\}$.

   Then let

$$\mathbf{X}_{n+1} := \mathbf{X}_n \cup \{\mathbf{x}_{n+1}\}.$$

In [66], the optimal points are selected from 10000 random points on the square $[-1, 1] \times [-1, 1]$; the total number of the point that are inside the optimal set $\mathbf{X}_n$ is decided by the user of this algorithm.

The K-Means Clustering algorithm, also known as Lloyd's algorithm, is another adaptive algorithm which is frequently applied in radial basis function neural networks. It is easy to implement and has good performance [57]. Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$ be the data set and $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_k\}$ be the set of k clustering centroids. The algorithm follows this procedure:

1. Randomly select k points from $\mathbf{X}$ as k cluster centroids.

2. Calculate the distance between each data point and cluster centroids.

3. Label the data point to the cluster whose distance from the cluster center is minimum of all the cluster centroids.

4. Recalculate the new cluster centroid using:

$$\mathbf{v}_i = \frac{1}{m_i} \sum_{j=1}^{m_i} \mathbf{x}_j,$$

   where m_i stands for the number of points in ith cluster.

5. Recalculate the distance between each data point and new obtained cluster centroids.

6. If no data point was reassigned then stop, otherwise repeat from Step 3.

Figure 1.3 shows the given data set $\mathbf{X}$ have been divided into 3 clusters which achieve by K-means clustering algorithm. Each colour stands for one cluster and the centroids are marked as black crosses.

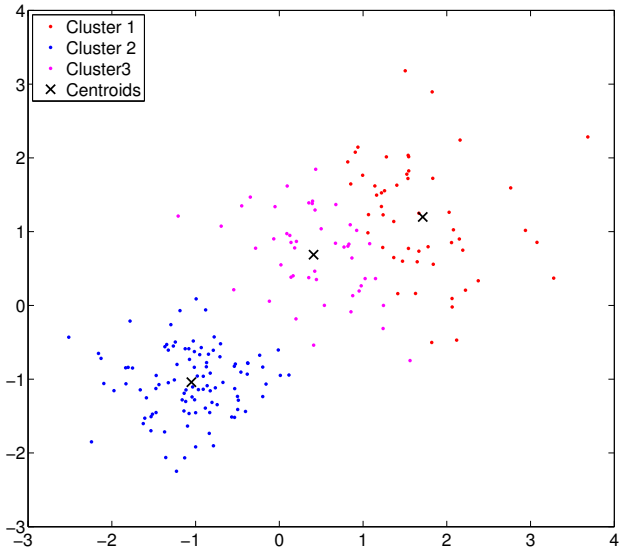

FIGURE 1.3: Cluster assignments and centroids for given data set $\mathbf{X}$

In [41], in order to take the target function value at the scattered points $f(\mathbf{X})$ into consideration, not only the locations of these points $\mathbf{X}$, it proposes the following weighted sequence scheme:

1. Having the scattered data point set and its corresponding function values, $\{(\mathbf{x}_i, f(\mathbf{x}_i)) : \mathbf{x}_i \in \mathbf{X}\}$.

2. Find $\mathbf{x}_i$ that satisfying

$$\mathbf{x}_1 = \arg\max\{f(\mathbf{x})||\mathbf{x}||_2 : \mathbf{x} \in \mathbf{X}\}$$

3. Then for $n = 2, 3, \cdots$

$$\mathbf{x}_n = \underset{\mathbf{x}_j \in \mathbf{X}\setminus\{\mathbf{x}_1,\cdots,\mathbf{x}_{n-1}\}}{\arg\max} \min_{1\leq k\leq n-1} f(\mathbf{x}_j)||\mathbf{x}_j - \mathbf{x}_k||_2.$$

[41] demonstrates an example of selecting 129 points from 40001 points in $[-2\pi, 2\pi]$, then using this selected 129 points to interpolate $f(x) = \sin(\frac{x}{2})$. It reported the error is in 10^{-8} magnitude.

Common sense suggests that in order to represent a target function with some finite discrete sampling points, more points are needed where the target function has more oscillations and less point should be placed where the target function is more regular. Driscoll and Heryudono [17] have developed the residual subsampling method of interpolation, used in boundary-value and initial-value problems with rapidly changing local features. Their method works as follows:

1. Approximates the unknown target function via RBF interpolation on uniformly distributed points.

2. Then the error is evaluated at intermediate points; this stage could be called the indication stage.

3. When the error exceeds a pre-set refinement threshold, corresponding points are added to the data set, and when the error is below a pre-set coarsening threshold, corresponding point are removed from the data set.

By applying this method, a pre-set threshold is needed, which is decided according to the user's demand of accuracy. The algorithm automatically stops when there is no adding or removing points, which means that the interpolant has reach the pre-set accuracy. In this method, knowledge of the target function is assumed, at least

the assumption that evaluation from the target function is very straight-forward and of low-cost.

Behrens and Iske et al. [3] have combined an adaptive semi-Lagrangian method with local thin-plate splines interpolation with leave-one-out (LOO) principle. The local interpolation gives out the fundamental rule of adaptation and it is crucial for approximation accuracy and computational efficiency. Naqvi [71] applied the LOO principle to construct an error indicator in order to solve one dimensional time dependent KortewegâĂŞde Vries (KdV) eaquation. In the BENCHOP project for pricing financial options [83], Li considered time as one spatial dimensional and applied the error indicator, which was also constructed by the LOO principle to solve Black-Scholes equations in both one and two dimensions.

Having the above adaptive RBF interpolation methods in mind, one should be aware that the thinning algorithm (adding algorithm) and the geometric greedy algorithm only focus on the location of the scattered points $\mathbf{X}$. The weighted sequence scheme, residual subsampling method and the leave-one-out adaptive interpolation method also take the function value $f(\mathbf{x}_i), \mathbf{x}_i \in \mathbf{X}$ into consideration to construct the near optimal data set. All the above mentioned algorithms select the "optimal" set $\hat{\mathbf{X}}$ from the scattered data set $\mathbf{X}$ which has considerable amount of points in it.

1.4 Motivation

The author is aware that, in most applications, data is generated with no knowledge of a function from which it was derived, so that an approximation model is needed. When sampling from the target function is expensive and time-consuming, a model that can indicate the location for generating the next samples and can provide enough accuracy with as few as possible samples is very desirable. Such examples include industrial processes, such as engine performance, where one experiment for a different set of (potentially many) parameters might take hours or days. The turbine profile efficiency study [74] that is discussed in Section 1.1 is such a case.

Adaptive radial basis function (ARBF) interpolation is suitable for such problems, mainly due to its ease of implementation in the multivariate scattered data setting. Moreover, it would be convenient to have a method for indicating where the approximation quality is poor, so that the generation of new data provides the user with greater accuracy where needed. By having this desirable method, in expensive approximation applications, the user could save significant time and resources.

1.5 Main Achievements

In this thesis, the author describe a new method for adaptive RBF interpolation which could be a suitable solution for the kind of problems mentioned in the Section 1.4. As the numerical examples show, the method can indicate the best location to generate the next sample and can provide sufficient accuracy with fewer samples than more established techniques.

The goal is achieved by the use of an error indicator, a function which indicates the approximation quality at nodes inspected by the algorithm. The error indicator compares a global RBF interpolant and a local RBF interpolant. The advantage of this error indicator is that it requires no extra function evaluation and indicates regions where the approximation error is high, so that it generates sets of points which are good candidates for optimally reducing the global error. This is the key differences between this method and the subsampling method in [17], which needs to sample the target function at each indication stage. This method is called adaptive error indicator (AEI) RBF interpolation.

The author also present some preliminary results achieved by a multilevel adaptive error indicator (MAEI) RBF approximation method. This method is a hybrid method of the multilevel level approximation and the adaptive error indicator method using a domain decomposition approach.

The author has applied this error indicator adaptive (AEI) RBF interpolation method to the same example $f(x) = \sin(\frac{x}{2})$ in $[-2\pi, 2\pi]$ of [41]. By using a

total 870 evaluations from the target function and a total of 721 in the final approximation, the max error of interpolation is 2.1×10^{-8}. Figure 1.4 shows the error distribution on the domain.

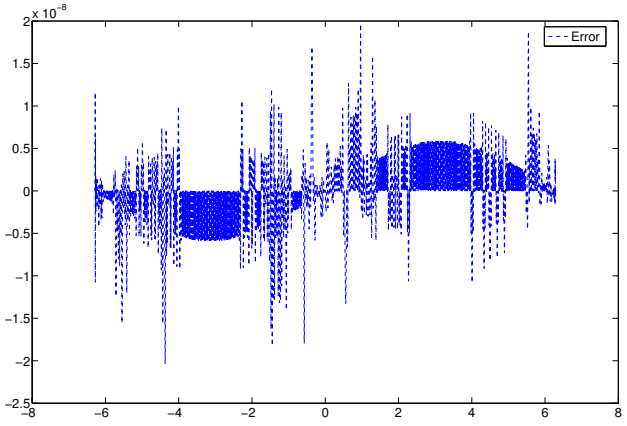

FIGURE 1.4: The final approximation error of $f(x) = \sin(\frac{x}{2})$ on $[-2\pi, 2\pi]$ which achieved by AEI method.

Then the author applied the multilevel adaptive error indicator (MAEI) scheme to approximate $f(x) = \sin(\frac{x}{2})$. By using a total 201 evaluations from the target function, the max error of the MAEI approximationis 1.9×10^{-8}. Table 1.1 summarizes the results of different approximation to the target function $f(x) = \sin(\frac{x}{2})$, N_{total} stands for the total evaluation times from the target function.

Method	Max Error	N_{total}
weighted-sequence	4.0E(-8)	40001
AEI	2.1E(-8)	870
MAEI	1.9E(-8)	201

TABLE 1.1: Summary results of approximation of $f(x) = \sin(\frac{x}{2})$.

In terms of times of total evaluations from the target function, the multilevel adaptive error indicator (MAEI) approximation and the adaptive error indicator (AEI) interpolation method can give massive savings when compared to the weighted-sequence method while delivering the same level of approximation quality. In the following part of the thesis, this point is developed with more numerical examples.

In Chapter 2, the basis definition of radial basis function and RBF interpolation are introduced. Chapter 3 describes the error indicator adaptive interpolation

(AEI) method and some numerical examples to demonstrate the ability of this method. In Chapter 4, the author introduces preliminary results of the multilevel adaptive error indicator interpolation (MAEI) method, also some numerical results are shown. Chapter 5 summarises this thesis and point out some possible future research directions.

Chapter 2

Radial basis functions

2.1 Scattered data interpolation problem

Definition 2.1 (Scattered data interpolation problem)**.** Let $\mathbf{X} \in \mathbb{R}^d$, given the data $(\mathbf{x}_i, f(\mathbf{x}_i)), i = 1, ..., N$, where $f : \mathbb{R}^d \to \mathbb{R}$, the multivariate scattered data interpolation problem is to find a function $S : \mathbb{R}^d \to \mathbb{R}$ such that $S(\mathbf{x}_i) = f(\mathbf{x}_i); i = 1, ..., N$ where S is called the interpolant to the data.

The data set $(\mathbf{x}_i, f(\mathbf{x}_i))$ could also be called sample set, $\mathbf{x}_i, i = 1, \cdots, N$, are the locations of the scattered points, and $f(\mathbf{x}_i)$ are the corresponding target function values at these locations. The author will assume that these values are obtained by sampling the target function f at data locations.

A convenient approach for solving the scattered data problem is to make the assumption that the interpolant $S(.)$ is a linear combination of certain basis functions, $\phi(\mathbf{x}, \mathbf{y})$, i.e.,

$$S(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i \phi(\mathbf{x}, \mathbf{x}_i) \quad \text{where} \quad \mathbf{x} \in \mathbb{R}^d. \tag{2.1}$$

Solving the interpolation problem under this assumption

$$S(\mathbf{x}_i) = f(\mathbf{x}_i), i = 1, 2, \cdots, N, \tag{2.2}$$

leads to the system of linear equations of the form

$$A\alpha = f, \tag{2.3}$$

where the entries of the interpolation matrix A are given by

$$A_{j,k} = \phi(\mathbf{x}_j, \mathbf{x}_k), \quad \text{where} \quad j,k = 1, 2, \cdots, N, \tag{2.4}$$

and $\alpha = [\alpha_1, \cdots, \alpha_N]^T, f = [f_1, \cdots, f_N]^T$.

As long as the matrix A is non singular, the unique solution of the problem exists. The non-singularity of the matrix A is guaranteed under some mild restrictions i.e., constant shape parameters and usually by adding a low order polynomial [69]. Shape parameters are user decided parameters. The methods to decide the shape parameters are introduced in Chapter 3. In this thesis, the radial basis functions are applied to solve scattered data interpolation problem.

2.2 Introductory concepts

In this section, some concepts that are related to radial basis function interpolation are introduced. For detail contents and a thorough description, please see Fasshauer [21] and Wendland [85].

The π_m^d denotes the space of d-variate polynomials whose degree does not exceed m. The dimension of the polynomial space π_m^d is stated as $\dim(\pi_m^d) = \binom{m+d}{d}$.

Definition 2.2 (Condition of unisolvancy)**.** The data sites $\mathbf{X} \subset \mathbb{R}^d$ with $N \geq M = \dim(\pi_m^d)$ are called π_m^d -unisolvent if the zero polynomial is the only polynomial from the space π_m^d that vanishes on all of them.

Example 2.1. *Three collinear points in* $\mathbb{R}^2$ *are not* π_1^1*-unisolvent since a linear interpolant, i.e., a plane through three arbitrary heights at these* 3 *collinear points is not uniquely determined. On the other hand, if a set of points in* $\mathbb{R}^2$ *contains* 3 *non-collinear points, then it is* π_1^1*-unisolvent.*

Definition 2.3 (Positive definite matrix)**.** A real symmetric $N \times N$ matrix A is called positive semi-definite if its associated quadratic form is non-negative, i.e.,

$$\sum_{j=1}^{N}\sum_{k=1}^{N}\lambda_j\lambda_k A_{jk} \geq 0,$$

for $\lambda = [\lambda_1, \cdots, \lambda_N]^T$. If the only vector λ that turns the above quadratic form into an equality is the zero vector, then A is called positive definite.

The positive definite matrix A has an inverse matrix A^{-1}, because its determinant is not zero.

Definition 2.4 (Positive definite function)**.** A real valued continuous function $\phi : \mathbb{R}^d \to \mathbb{R}$ is positive definite on $\mathbb{R}^d$ if and only if it is even and

$$\sum_{j=1}^{N}\sum_{k=1}^{N}\lambda_j\lambda_k\phi(\mathbf{x}_j - \mathbf{x}_k) \geq 0,$$

for any N pairwise different points $\{\mathbf{x}_1, \cdots, \mathbf{x}_N\} \subset \mathbb{R}^d$, and $\lambda = [\lambda_1, \cdots, \lambda_N]^T$.

The function ϕ is strictly positive definite on $\mathbb{R}^d$ if the only vector λ that turns the above into the equality is the zero vector.

Definition 2.5 (Conditionally positive functions)**.** A continuous real even function $\phi : \mathbb{R}^d \to \mathbb{R}$ is said to be conditionally positive definite of order m on $\mathbb{R}^d$ if and only if

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_i\lambda_j\phi(\mathbf{x}_i - \mathbf{x}_j) \geq 0,$$

holds for all possible pairs $(\lambda, \mathbf{X})$ of choices $\lambda = [\lambda_1, \cdots, \lambda_N]$ and $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\} \subset \mathbb{R}^d$ satisfying the vanishing moment conditions

$$\sum_{j=1}^{N}\lambda_j P(\mathbf{x}_j) = 0,$$

for all $P \in \pi_{m-1}^d$. It is not possible to list all of the analogues and generalizations of positive definite functions but some of the properties of the positive definite and conditionally positive definite functions are listed below (details can be found in [82]).

- If $\phi_1, \cdots, \phi_N$ are positive definite and $\lambda_j > 0$, then $\Phi := \sum_{j=1}^{N} \lambda_j \phi_j$ is also a positive definite function.
- $\Phi(0) \geq 0$.
- Any positive definite function is bounded. In fact

$$|\phi(x)| \leq \phi(0).$$

- If ϕ is positive definite with $\phi(0) = 0$, then $\phi \equiv 0$.
- The product of two positive definite functions is positive definite.

Definition 2.6 (Fill distance)**.** The fill distance corresponding to the data set $\mathbf{X}$ in Ω is defined as

$$h_{\mathbf{X},\Omega} = \sup_{\mathbf{x}\in\Omega} \min_{\mathbf{x}_j\in\mathbf{X}} ||\mathbf{x} - \mathbf{x}_j||_2.$$

This is also called the covering radius. The geometrical interpretation of fill distance is the largest possible empty sphere amongst the points in this data set. It is a measure of the data distribution and can indicate how well the domain Ω is filled with the data in the set $\mathbf{X}$. In Figure 2.1, it shows the fill distance of this centers set.

Definition 2.7 (Separation distance)**.** The separation distance is defined as

$$q_{\mathbf{X}} = \frac{1}{2} \min_{i\neq j} ||\mathbf{x}_i - \mathbf{x}_j||_2.$$

This is also called as packing radius. The geometrical interpretation of the separation distance is that no two balls of radius "$q_{\mathbf{X}}$" centered at each center will overlap. In Figure 2.1, it shows the separation distance of this centers set.

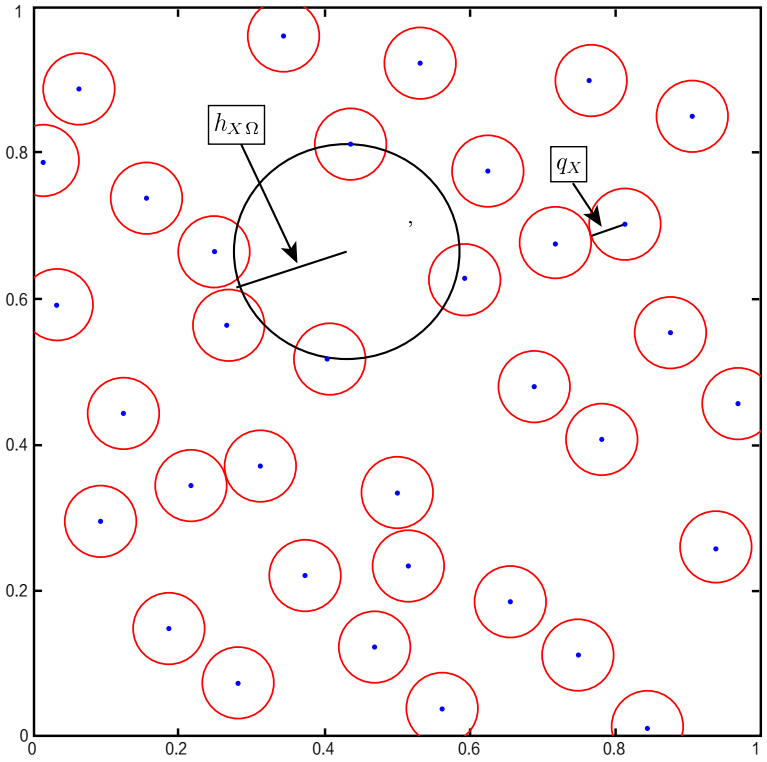

FIGURE 2.1: Fill distance and Separation distance for this sample centers set.

Definition 2.8 (Condition number)**.** The condition number of a matrix A with respect to any matrix norm $||.||$ is defined as

$$\kappa(A) = ||A||||A^{-1}||.$$

The condition number of an RBF interpolation matrix A depends on the separation distance (see Definition 2.7) and the fill distance (see Definition 2.6) and provides information on the numerical stability of the interpolation process. To do so one has to investigate both the maximum and the minimum eigenvalues. A condition number is used to quantify the sensitivity to perturbations of a linear system, such as Equation 2.3, and to estimate the accuracy of a computed solution. Using the 2-norm, the matrix condition number is

$$\kappa(A) = ||A||_2||A^{-1}||_2 = \frac{\sigma_{\max}}{\sigma_{\min}}, \tag{2.5}$$

where $\sigma_{\max}$ and $\sigma_{\min}$ are the largest and smallest eigenvalues (in absolute size) of the symmetric matrix A.

A well-conditioned matrix will have a small condition number $\kappa(A) \approx 1$, while an ill-conditioned matrix will have a much larger condition number. A system of equations, like Equation 2.3, is considered to be well-conditioned if a small change in f results in small change in α, while small change in f could results in large change in α in ill-conditioned equations. The reason the condition number should be keep low is that, theoretically one less digit of accuracy will be obtained in the computed solution as the condition number increases by a factor of 10.

2.3 Radial basis function

Over the last two decades, due to the numerous advantages RBF offer [8, 19, 85], the application of RBF has had a fast development in many research fields. Pena listed some RBF applications (Table 1) in [73].

2.3.1 Radial basis function interpolation

Definition 2.9 (Radial function)**.** A function $\Phi : \mathbb{R}^d \to \mathbb{R}$ is said to be radial if there exist a univariate function $\phi : [0, \infty) \to \mathbb{R}$ such that $\Phi(\mathbf{x}) = \phi(r)$, where $r = ||\mathbf{x}||$ and $||.||$ is some norm on $\mathbb{R}^d$ (usually the Euclidean norm).

The definition can be explained as that for a finite set of distinct points $\mathbf{X} \subset \mathbb{R}^d$ called the centers, that the function value $\Phi(\mathbf{x}), \mathbf{x} \in \mathbf{X}$, solely depends on $||\mathbf{x}||$ but not on the coordinates of $\mathbf{x}$. The function Φ is radially symmetric about the origin. The reason that makes radial functions most useful for applications is the fact that the interpolation problem becomes insensitive to the dimension d of the space in which the data set lies. Instead of having to deal with a multivariate function, whose complexity will increase with increasing space dimension d, one can work with the same univariate function ϕ for all choices of d.

Given the centers set $\mathbf{X} = \{\mathbf{x}_i, i = 1, \cdots, N\}$, which is a set of distinct points in $\mathbb{R}^d$, and the function values $f(\mathbf{x}_i), i = 1, \cdots, N$. The RBF approximation to the

function f is of the form

$$S(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i \phi(||\mathbf{x} - \mathbf{x}_i||_2). \tag{2.6}$$

The coefficients, α, are determined by enforcing the interpolation condition

$$S(\mathbf{x}_i) = f(\mathbf{x}_i), \tag{2.7}$$

at the center set $\mathbf{X}$. Enforcing the interpolation condition at N centers results in a $N \times N$ linear system

$$A\alpha = f, \tag{2.8}$$

where $A_{ij} = (\phi(||\mathbf{x}_i - \mathbf{x}_j||_2)), 1 \leq i, j \leq N$.

The above linear system has to be solved for the coefficients α. The matrix A is called the interpolation matrix or the system matrix and consist of the functions serving as the basis of the approximation space. To evaluate the interpolant at points $\mathbf{Y} = \{\mathbf{y}_i, i = 1, 2, ..., M\}$ using Equation 2.6, $M \times N$ evaluation matrix H is formed with entries

$$H_{i,j} = \phi(||\mathbf{y}_i - \mathbf{x}_j||_2), \quad i = 1, \cdots, M. \quad j = 1, \cdots, N.$$

Then the interpolant is evaluated at the M points by the matrix multiplication

$$S(\mathbf{Y}) = H\alpha.$$

Sometimes the assumption on the form Equation 2.1, for solving to the scattered data interpolation problem (see Definition 2.1) is extended by adding certain polynomials to the expansion, i.e., $S(\mathbf{x})$ is now assumed to be of the form

$$S(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i \phi(||\mathbf{x} - \mathbf{x}_i||_2) + P(\mathbf{x}), \tag{2.9}$$

where $\mathbf{x} \in \mathbb{R}^d$ and $P \in \pi^d_{m-1}$. Equation 2.9 can be written as

$$S(\mathbf{x}) = \sum_{i=1}^{N} \alpha_j \phi(||\mathbf{x} - \mathbf{x}_i||_2) + \sum_{j=1}^{q} \beta_j p_j(\mathbf{x}), \tag{2.10}$$

where the polynomials $p_1, \cdots, p_q$ form a basis for the $q = \binom{d+m-1}{d}$-dimensional linear space π^d_{m-1} of polynomials of total degree less than or equal to $m-1$ in d variables. The coefficients $\alpha_i, i = 1, 2, \cdots, n$ and $\beta_j, j = 1, 2, \cdots, q$, are to be determined by the interpolation condition and the additional condition of Equation 2.12.

Enforcing the interpolation conditions $S(\mathbf{x}_i) = f(\mathbf{x}_i), i = 1, ..., N$, leads to a system of N linear equations in $N + q$ unknowns α_i and β_j , one usually adds the q additional conditions to ensure a unique solution. Imposing the interpolation conditions on the interpolant $S(.)$

$$\sum_{i=1}^{N} \alpha_i \phi(||\mathbf{x}_k - \mathbf{x}_i||) + \sum_{j=1}^{q} \beta_j p_j(\mathbf{x}_k) = f(\mathbf{x}_k), \quad k = 1, \cdots, N. \tag{2.11}$$

Equation 2.11 is a linear system of N equations in $N + q$ unknown variables in coefficient vector $\alpha = [\alpha_1, \cdots, \alpha_N]^T$ of the major part and $\beta = [\beta_1, \cdots, \beta_q]^T$ of the polynomial part of the interpolant. However, this leave the author with q free coefficients to find, so some extra conditions are needed. Mimicking the natural conditions for cubic splines gives the following additional condition:

$$\sum_{i=1}^{N} \alpha_i p_j(\mathbf{x}_i) = 0, \quad j = 1, 2, \cdots, q, \tag{2.12}$$

where $p \in \pi^d_{m-1}$ and m is the order of the basis function ϕ.

Combining the interpolation condition and side condition together, the system can be written as

$$\begin{pmatrix} A & P^T \\ P & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}, \tag{2.13}$$

where

$$A_{ij} = (\phi(\|\mathbf{x}_i - \mathbf{x}_j)\|)), \quad 1 \leq i, j \leq N,$$

$$P_{ij} = (p_j(\mathbf{x}_i)), \quad 1 \leq i \leq N, \quad 1 \leq j \leq q,$$

$$0 = \text{zero matrix} \in \mathbb{R}^{q \times q}.$$

Schaback [80] discusses the solvability of the above system, which is guaranteed by the requirement that $\text{rank}(P) = q \leq N$ and

$$\lambda \|\alpha\|^2 \leq \alpha^T A \alpha \tag{2.14}$$

for all $\alpha \in \mathbb{R}^N$ with $P\alpha = 0$, where λ is a positive constant. The last condition is a condition on the function ϕ, and functions which satisfy this condition, irrespective of the choice of the points in $\mathbf{X}$, are called *conditionally positive definite of order* m. The condition $rank(P) = q \leq N$ is called π_m^d-unisolvent of $\mathbf{X}$, because such sets of polynomials are uniquely determined by their values on the set $\mathbf{X}$.

Moreover, if the data come from a polynomial of total degree at the most $m-1$, then they are fitted exactly by the interpolant. As commented in [1], the addition of polynomial terms does not improve greatly the accuracy of approximation for non-polynomial functions.

The classical choice for radial basis function ϕ along with their order m are shown in Table 2.1.

Name of RBFs	$\phi(r) =$	Parameters	Order m
Gaussians	$e^{-(cr)^2}$	$c > 0$	$m \geq 0$
Multiquadric (MQ)	$(1 + c^2 r^2)^{\frac{v}{2}}$	$v > 0, v \notin 2\mathbb{N}, c > 0$	$m \geq \lceil \frac{v}{2} \rceil$
Inverse Multiquadric (IMQ)	$(1 + c^2 r^2)^{\frac{v}{2}}$	$v < 0, c > 0$	$m \geq 0$

TABLE 2.1: Examples of some radial basis functions.

2.4 Well-posedness of the RBF interpolation problem

Discussion of the existence of a unique solution to the above interpolation problem is presented in below.

Theorem 2.10 (Bochner's Theorem)**.** *A continuous function $\Phi \in C(\mathbb{R}^d)$ is positive definite on $\mathbb{R}^d$ if and only if it is the Fourier transform of a finite non-negative Borel measure μ on $\mathbb{R}^d$, i.e.*

$$\Phi(\boldsymbol{x}) = \hat{\mu}(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^d}} \int_{\mathbb{R}^d} e^{-i\boldsymbol{x}^T \boldsymbol{y}} d\mu(\boldsymbol{y}), \quad \boldsymbol{x} \in \mathbb{R}^d$$

Lemma 2.11. *The Gaussian $\Phi(r) = \exp(-\alpha\|r\|^2)$, $\alpha > 0$, is positive definite on $\mathbb{R}^d$.*

Proof. The above Lemma is true as the Fourier transform of the Gaussian is essentially the Gaussian. For example, The Gaussian $\Phi(r) = \exp(-\frac{\|r\|^2}{2})$, has a Fourier transform

$$\hat{\Phi}(r) = (2)^{-d}(\alpha\pi)^{-\frac{d}{2}} \int_{\mathbb{R}^d} e^{-\frac{\|r\|^2}{4\alpha}} e^{-i\mathbf{x}^T w} dw, \tag{2.15}$$

this means that Φ is positive definite, and by Bochner's Theorem: every positive definite function is the Fourier transform of a positive function. If $\alpha = \frac{1}{\sqrt{2}}$ then $\hat{\Phi} = \Phi$. □

Definition 2.12 (Completely monotone)**.** A function φ is completely monotone on $[0, \infty)$ if :

1. $\varphi \in \mathbf{C}[0, \infty)$.
2. $\varphi \in \mathbf{C}^\infty(0, \infty)$.
3. $(-1)^l \varphi^{(l)}(r) \geq 0$ where $r > 0$ and $l = 0, 1, \cdots$

Theorem 2.13 (Micchelli)**.** *Let $\varphi(r) = \phi(\sqrt{r}) \in C[0, \infty)$ and $\varphi(r) > 0$ for $r > 0$. be such that φ' is completely monotonic but not constant. Then for any set of N*

distinct centers $\{\mathbf{x}_j\}_{j=1}^N$*, then the* $N \times N$ *matrix* A *with entries* $A_{jk} = \phi(||\mathbf{x}_j - \mathbf{x}_k||_2)$ *is invertible.*

For the MQ $\phi(r) = \sqrt{(1 + c^2 r^2)}$ one has

$$\varphi(r) = \phi(\sqrt{r}) = \sqrt{(1 + c^2 r)}$$

and

$$\varphi^{'}(r) = \frac{c^2}{2\sqrt{1 + c^2 r}},$$

$$\varphi^{''}(r) = \frac{-c^4}{4(1 + c^2 r)^{3/2}},$$

$$\varphi^{(3)}(r) = \frac{3c^6}{8(1 + c^2 r)^{5/2}},$$

$$\varphi^{(4)}(r) = \frac{-15c^8}{16(1 + c^2 r)^{7/2}},$$

$$\vdots \quad = \quad \vdots \quad ,$$

Thus

$$(-1)^l \varphi^{(l)}(r) \geq 0$$

and $\varphi^{'}(r)$ is completely monotone and the invertibility of the MQ interpolation matrix A is proofed.

The proof of invertibility of Gaussian and MQ are directly taken from [85] and [79], respectively.

2.4.1 Convergence and Error Bound

Schaback et al. used symmetric interpolation matrix to give the error bound [35] and Fornberg et al. found the convergence rate of RBFs in 1-D of equally spaced points [29]. Fornberg et al. also noted that the RBF interpolation error structure oscillates around zero mean in their experiment ($\cos(wx)$). A convergence analysis of this adaptive algorithm appears not to be available in the literature. Before

going any further, the author introduces some relevant and necessary definitions needed in the analysis of convergence of this adaptive method in the future.

From the work of [65, 85], smooth RBFs, such as Gaussian, inverse multiquadratics (IMQ) and MQ on native spaces have been known to converge exponentially [7, 65]. Furthermore, Gaussian basis functions have small native space with analytical function [75], the proof of convergence were also given by Madych and Nelson in [64] that was based on the fundamental work of [39]. Subsequently this has been extended by [58, 69, 76, 77, 81].

The convergence in scattered data interpolation was proved by Wu in [86] and the convergence rate of the RBFs method for the solution of PDEs was investigated in [35, 84]. For the shape parameter, Driscoll and Fornberg showed some result of convergence in both 1-D (converge to polynomial) and 2-D for smooth functions when c tends to zero with small number of N [16]. At equidistant distribution of center nodes, Platte and Driscoll applied the variable change, and found that there was a connection between polynomials and Gaussian interpolation [18]. For a polynomial interpolation, the rate of convergence can be denoted by the method of classical approximation theory, such as the stability of the interpolation problem of the Runge phenomenon analysis, which indicates the optimal distribution set of nodes. In a Hermite interpolation, the investigation of the rate of convergence of the Hermite interpolation has been done by Luo and Levesley [56] with a modification method of variational approach of Madych and Nelson [63, 64] by a fixed conditional positive definite function.

Chapter 3

Adaptive Error Indicator (AEI) Method

3.1 Adaptive RBF interpolation scheme

The backbone of most adaptive methods for solving approximation problems is a cyclic process of three procedures:

$$\cdots \quad \textit{Solve} \text{ — } \textit{Estimate} \text{ — } \textit{Refine/Coarsen} \cdots$$

and the cycle is terminated when a certain stopping criterion is achieved. In this chapter, the author introduces an adaptive RBF method that follows this above outline. In a scattered data approximation problem, the procedure "Solve" means solutions are obtain by solving a system of a linear equations either by direct or iterative methods in which the linear system is the RBF interpolation system. Once some approximated solutions are known, the procedure "Estimate" gathers information about errors, defined by the difference between the approximation and the real target function values, at given check points. These errors are used as criteria to decide whether the degrees of freedom must be added or reduced. Then these decisions are passed along to the "Refine/Coarsen" stage, in the "Refine" stage more samples are added and in the "Coarsen" stage some samples are removed.

The error determines how close the approximate solution is to the known exact solution. With that knowledge, convergence of the method can be guided and unnecessary cycles can be avoided. However, in most cases, finding error estimates can be very challenging. Rather than pursue rigorous error estimates for particular problems, the author has taken the practical and effective step of using an error indicator function to suggest where changes should be made. In other words, the error indicator function can be used in the "Estimate" stage to effectively estimate the error at given check points.

This adaptive error indicator RBF interpolation method will be explained in the following sections. In Section 3.2, the author define the error indicator function and specify the refinement and coarsening rules. Section 3.3 gives the details of adaptive data structure and adaptive shape parameters matching the error indicator function. Section 3.4 summarise the procedure of adaptive error indicator (AEI) interpolation. In Section 3.5, numerical results for one, two and three dimensions are shown.

3.2 The error indicator function

In this section, the basic features of the error indicator function are explained. Consider a function $f : \mathbb{R}^d \to \mathbb{R}$, a real valued function of d variables, that is to be approximated by $S_{\mathbf{X}} : \mathbb{R}^d \to \mathbb{R}$, given the centers $f(\mathbf{x}_i) : i = 1, 2, 3, ..., N$, where $\mathbf{x}_i : i = 1, 2, 3..., N$ is a set of distinct points in $\mathbb{R}^d$.

The approximation to the function f is of the form

$$S_{\mathbf{X}} = \sum_{i=i}^{N} \alpha_i \phi_i(||\mathbf{x} - \mathbf{x}_i||_2) + \sum_{j=1}^{q} \beta_j p_j(\mathbf{x}), \tag{3.1}$$

where $\phi_i : \mathbb{R}_+ \to \mathbb{R}$ is a radial basis function, $||.||_2$ denotes the Euclidean norm on $\mathbb{R}^d$. This form of the approximation is different to the standard form in Chapter 2, in which the basis function ϕ_i is the same for all i. The author is leaving ourselves the flexibility of changing the basis function, via a different choice of

shape parameter, depending on the density of data points in a particular region. The author will comment later on how to do this.

In this method, it generates a sequence of sets $\mathbf{X}_0, \mathbf{X}_1, \cdots$, where it generate $\mathbf{X}_{k+1}$ from $\mathbf{X}_k$ via a refinement and coarsening strategy which will be described below. In contrast with e.g. Iske and Levesley [52], this method do not use a nested sequence of points. The strategy for including or removing points depends on an error indicator. The author follow the idea of Behrens et al. [3] who wished to decide on the positioning of points in a semi-Lagrangian fluid flow simulation. They compared a local interpolant with some known function and refined where the error was large, and coarsened where small.

This error indicator is based on the principle that in a region which is challenging for approximation, two different approximation methods will give significantly different results, when compared with regions where approximation is more straightforward. The first approximation method is the current interpolant $S^{\text{multi}}_{\mathbf{X}_k}$ at level k. The second approximation method will be via a polyharmonic spline interpolant based on values of the approximation on a local set of points. Then a function $\eta(\mathbf{x})$ with domain in the current indication set, assigns a positive value to each indication point ξ. This value indicates the local approximation quality at each indication nodes, and serves to determine where the approximate solution $S^{\text{multi}}_{\mathbf{X}_k}$ requires more accuracy at these specified indication nodes, and requires no extra function evaluation. Below, the author gives the definition of the error indicator which is proposed in this thesis.

Definition 3.1 (Error indicator 1)**.** For $k \geq 0$, let the indication set Ξ_k, corresponding to $\mathbf{X}_k$, be a set of scattered points, at which one wants to know the approximation quality.

The error indicator function $\eta(\xi)$ is defined by

$$\eta(\xi) = |S^{\text{multi}}_{\mathbf{X}_k}(\xi) - S^{\text{ps}}_{N_\xi}(\xi)|, \quad \xi \in \Xi_k. \tag{3.2}$$

The value $S^{\text{multi}}_{\mathbf{X}_k}(\xi)$ is the multiquadric radial basis function approximation of the target function at ξ by the center set $\mathbf{X}$. The value $S^{\text{ps}}_{N_\xi}(\xi)$ is a polyharmonic spline

radial basis function reconstruction which matches the target function value at ξ by a scattered point set N_ξ in a neighbourhood around ξ. The specified polyharmonic spline depends on the dimension of the problem. N_ξ is a subset of the current centers set $\mathbf{X}_k$. One could call N_ξ the neighbourhood set of ξ, the elements in N_ξ are the M nearest neighbour points to ξ from the center set $\mathbf{X}_k$. Then

$$S^{\mathrm{ps}}_{N_\xi}(v) = f(v) \quad for \quad v \in N_\xi. \tag{3.3}$$

For $k = 0$, the indication set Ξ_0 is determined by $\mathbf{X}_0$, for $k > 0$, the indication set Ξ_k is determined by $\mathbf{X}_k$ and $\mathbf{X}_{k-1}$. The details of the relationship between Ξ_k and $\mathbf{X}_k$ is explained in the algorithm flow steps.

The polyharmonic splines have the following form:

$$\phi_{d,k}(r) = \begin{cases} r^{2k-d}\log(r), & \text{if } d \text{ is even,} \\ r^{2k-d}, & \text{if } d \text{ is odd,} \end{cases} \tag{3.4}$$

where k is required to satisfy $2k > d$.

For $d = 1$, with $m = k = 3$, the neighbourhood set of ξ is $N_\xi = \{x_1, x_2, ..., x_M\}$ and the local approximation is

$$S^{\mathrm{ps}}_{N_\xi}(x) = \sum_{i=1}^{M} \alpha_i (||x - x_i||_2)^5 + \beta_1 + \beta_2 x + \beta_3 x^2, \tag{3.5}$$

which is conditionally positive definite of order 3.

For $d = 2$, with $m = k = 2$, the neighbourhood set of ξ is $N_\xi = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M\}$ with $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$, and

$$S^{\mathrm{ps}}_{N_\xi}(\mathbf{x}) = \sum_{i=1}^{M} \alpha_i (||\mathbf{x} - \mathbf{x}_i||_2^2) \log(||\mathbf{x} - \mathbf{x}_i||_2) + \beta_1 + \beta_2 x_1 + \beta_3 x_2, \tag{3.6}$$

which is conditionally positive definite of order 2.

For $d = 3$, with $m = k = 3$, the neighbourhood set of ξ is $N_\xi = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M\}$ with $\mathbf{x} = (x_1, x_2, x_3) \in \mathbb{R}^3$, and

$$S^{\text{ps}}_{N_\xi}(\mathbf{x}) = \sum_{i=1}^{M} \alpha_i(||\mathbf{x} - \mathbf{x}_i||_2) + \beta_1 + \beta_2 x_1 + \beta_3 x_2 + \beta_4 x_3 + \beta_5 x_1 x_2 + \beta_6 x_2 x_3 + \beta_7 x_1 x_3 + \beta_8 (x_1)^2 + \beta_9 (x_2)^2 + \beta_{10} (x_3)^2, \quad (3.7)$$

which is conditionally positive definite of order 3. The M is specified in the numerical examples. The length of the polynomial part in $S^{\text{ps}}_{N_\xi}$ will increase with the increment of dimensional number d, this will increase the computational complexity in high dimensions.

The error indicator defined above measures the deviation between a global approximation and a local approximation at the point ξ. The intuition inside this method is simple, when ξ lies in a smooth region of the function, two different approximation should give out similar results, then the error indicator $\eta(\xi)$ is expected to be small, whereas in the region of less regularity for f, or around discontinuities, the error indicator $\eta(\xi)$ is expected to be large.

The author try to present a error bound for this error indicator function $\eta(\xi)$. The error indicator is

$$\eta(\xi) = |S^{\text{multi}}_{\mathbf{X}_k}(\xi) - S^{\text{ps}}_{N_\xi}(\xi)|,$$

$$\eta(\xi) = |S^{\text{multi}}_{\mathbf{X}_k}(\xi) - f(\xi) + f(\xi) - S^{\text{ps}}_{N_\xi}(\xi)|,$$

$$\eta(\xi) \leq |f(\xi) - S^{\text{multi}}_{\mathbf{X}_k}(\xi)| + |f(\xi) - S^{\text{ps}}_{N_\xi}(\xi)|. \quad (3.8)$$

Fasshauer's book [21] (Page 125) gives the error bound for stationary interpolation of MQ RBF and polyharmonic splines RBF. They are :

$$|f(\xi) - S^{\text{multi}}_{\mathbf{X}_k}(\xi)| \leq C_1 \exp(-K_1/h_{\mathbf{X}_k,\Omega}),$$

$$|f(\xi) - S^{\text{ps}}_{N_\xi}(\xi)| \leq C_2 h^{K_2}_{N_\xi,\Omega},$$

where K_1, K_2, C_1, C_2 are constants. Then Equation 3.8 becomes

$$\eta(\xi) \leq \exp(-K_1/h_{\mathbf{X}_k,\Omega}) + C_1 h^{K_2}_{N_\xi,\Omega}. \tag{3.9}$$

Then the error indicator $\eta(\xi)$, $\xi \in \Xi$ is used to flag points $\xi \in \Xi$ as "to be refined" or its corresponding center $\mathbf{x}$ "to be coarsened" according to the following definition.

Definition 3.2 (Refine and coarsen)**.** Let θ_{coarse}, θ_{refine} be two tolerance values satisfying $0 < \theta_{\text{coarse}} < \theta_{\text{refine}}$. One refines a point $\xi \in \Xi$, and place it in $\mathbf{X}_{\text{refine}}$, if and only if $\eta(\xi) > \theta_{\text{refine}}$, and one moves a point from the active center set $\mathbf{X}$ into the coarse set $\mathbf{X}_{\text{coarse}}$, if and only if corresponding $\eta(\xi) < \theta_{\text{coarse}}$.

These two parameters $\theta_{\text{coarse}}, \theta_{\text{refine}}$ should be specified according to the user's need. Thus one has two processes: coarsening where a coarse set $\mathbf{X}_{\text{coarse}}$ is removed from the current center set $\mathbf{X}$, that is the new center set $\mathbf{X}$ is modified by replacing $\mathbf{X}$ with $\mathbf{X} \setminus \mathbf{X}_{\text{coarse}}$; refinement where a set of nodes $\mathbf{X}_{\text{refine}}$ is added to the current center set where the error is large, in other words, $\mathbf{X}$ is modified by replacing $\mathbf{X}$ with $\mathbf{X} \cup \mathbf{X}_{\text{refine}}$.

When applying this error indicator it require no extra evaluation of the target function so that no extra cost is paid in finding where approximation is likely to be poor. When function evaluation is very costly this is a very positive feature of the method.

3.3 Adaptive data structure

3.3.1 Adaptive point sets

Now the author describes the relationship between the center set $\mathbf{X}_k$ and the corresponding indication set Ξ_k. In one dimensional cases, the initial center set $\mathbf{X}_0 = \{x_1, x_2, \cdots, x_{n_0}\}$ is a set of uniformly distributed nodes in the domain. For $k \geq 0$, the indication nodes in Ξ_k are the middle points of the current centers,

that is $\Xi_k = \{\xi_i = 0.5(x_i + x_{i+1}), i = 1, 2, \cdots, n_k - 1\}$. The author gives out the details of rules of refining and coarsening centers for $d = 1$:

- Each indication point is checked independently by the error indicator η for addition to the center set according to Definition 3.2.
- The author defines the corresponding indication point for x_i is ξ_{i-1} for $i = 2, \cdots, n$, all the corresponding indication points, one indication point for one center in $d = 1$ cases, for center x is checked by error indicator η, then follow Definition 3.2 to coarsen the appropriate centers.

Figure 3.1 shows the example of initial centers set $\mathbf{X}_0$, which are the black nodes, and the corresponding indication set Ξ_0, which are the red nodes. Note that the first and last center are always left intact.

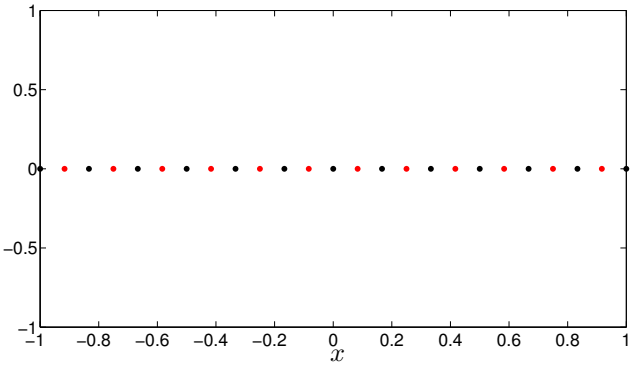

FIGURE 3.1: Initial centers and indication points of an interval.

For two-dimensional cases $d = 2$, one follows the setting in [17]. They begin with a coarse collection of non overlapping regular boxes in $\mathbb{R}^d$ that cover the domain Ω of interest. Each initial box has an RBF center at its midpoint, and the author calls such boxes center boxes and the centers consist of the initial center set $\mathbf{X}$. Each of the 2^d child boxes arising from the natural subdivision (bisection) of a center box is an indication box, and the center of each indication box is an indication point ξ; these indication points form the indication set Ξ. The box structure implies a quadtree, or octree data structure in two, and three dimensions [45].

Figure 3.2 shows the example of initial centers and indication set for a two dimensional square. Note the four corner points are added, these points are always left

alone in "Refine/Coarsen" stage. In order to initialize boxes for each boundary side in the square case, one simply treats each side as an independent one dimensional case. Figure 3.2 shows the indicator set (red nodes) corresponding to the equally spaced points in the square (black nodes). The initial centers that consist of two types : 1) the interior nodes and 2) the boundary nodes which include 4 vertices. The red nodes are grouped to became the indication set Ξ_0.

To summarize, the geometric data structure is nothing more than finding a set of centers $\mathbf{X}$, which more or less represent the domain Ω, and the indication set Ξ which is finer than the set $\mathbf{X}$.

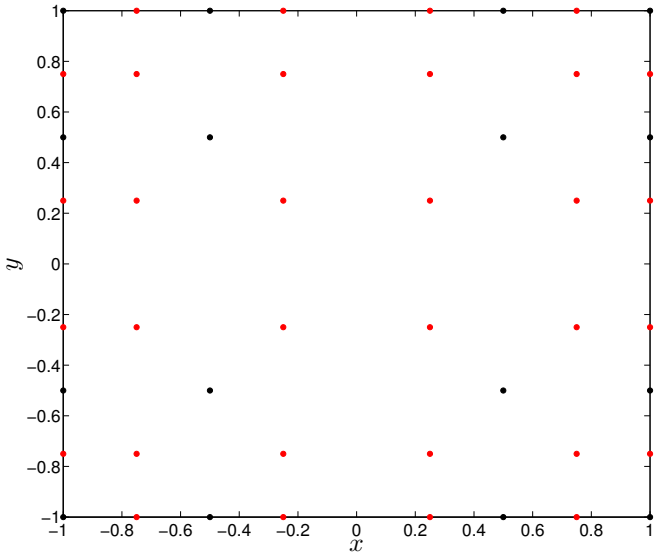

FIGURE 3.2: Initial centers and indication points of a square region, with $n = 2^j, j = 1$.

The interpolant $S_{\mathbf{X}}^{\text{multi}}$ is created using the current set of centers from the center set $\mathbf{X}$. The error indicator η (in this case, the interpolation error) of the interpolant is then evaluated at the indication points Ξ, and adaptation decisions are made. The author gives out the details of rules regarding refining and coarsening centers for $d = 2$:

- That each center $\mathbf{x}$ has 2^d sibling indication points. These 4 sibling indication points are checked by error indicator η independently, then the addition is constructed following Definition 3.2.

- For any center $\mathbf{x}$ whose all sibling indication points have error indicator value η less than θ_{coarse}, then this center $\mathbf{x}$ is removed according Definition 3.2.

The author gives a visible example of refinement and coarsening. Converting the indication points where the error indicator values η are greater that θ_{refine} described as $\bigotimes$ into RBF centers as black dots, see Figure 3.3. Removing current center (black dot in Figure 3.4 left corner) where error indicator value η at all of its sibling indication points, described as $\bigotimes$, are less than the threshold θ_{coarse}.

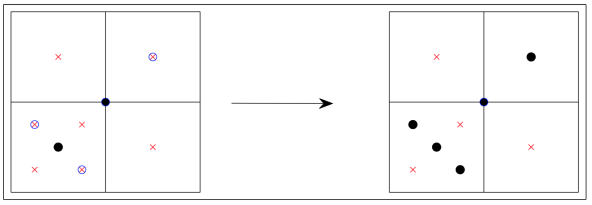

FIGURE 3.3: Examples of refinement in $d = 2$ cases.

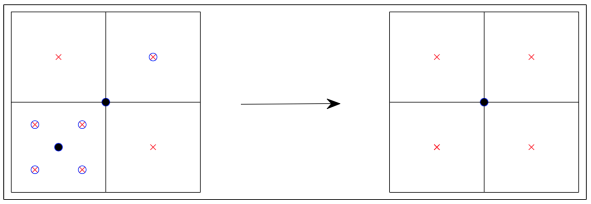

FIGURE 3.4: Examples of coarsen in $d = 2$ cases.

In Algorithm 1, the author gives the process how the centers set $\mathbf{X}_k$ and corresponding indication set Ξ_k evolve according to the error indicator η.

In three dimensional cases, one can extend the two dimensional node scheme, the relationship between center set $\mathbf{X}_k$ and indication set $\Xi_k, k = 0, 1, 2, \cdots$ following the same principal as in two-dimensional cases and Algorithm 1.

3.3.2 Adaptive shape parameters

For the multiquadric RBF and the Gaussian RBF one has a free parameter c which is named the shape parameter, which can be decided by the user. As one increases c the $\phi(r) = \sqrt{(1 + c^2 r^2)}$, the MQ basis function behaves more and more like the function cr, so that one gets a sharp corner near $r = 0$.

Algorithm 1 Calculate $\mathbf{X}_k, \Xi_k, k = 0, 1, 2, \cdots$ in $[-1, 1]^2$.

Initialization: $n = 2^j, j \in \mathbf{N}, h = 2/n, k = 0$.
For interior nodes
$Inodes = (-1 + h/2 + rh, -1 + h/2 + sh), r, s = 0, 1, \cdots, n-1$.

For boundary nodes
$Bnodes = (-1 + h/2 + rh, \pm 1), (\pm 1, -1 + h/2 + rh), r = 0, 1, \cdots, n-1$.
$Pnodes = (\pm 1, \pm 1)$

Initial center set $\mathbf{X}_0 \leftarrow Inodes \cup Pnodes \cup Bnodes$.
$endflag = true$

while *endflag* **do**
$k \leftarrow k + 1$.
$h \leftarrow h/2$.

$Inodes_{new} \leftarrow$ Form squares centered at each node in $Inodes$ with boundary length h_k. Put the vertices of each square into $Inodes_{new}$.

$Bnodes_{new} \leftarrow$ Choose points on the boundary which are $0.5h$ away from the current nodes in $Bnodes$. Add these to $Bnodes_{new}$.

$\Xi_k \leftarrow Inodes_{new} \cup Bnodes_{new}$
Use error indicator to decide the points at which to refine $\mathbf{X}_{\text{refine}} \subset \Xi_k$.
$Inodes \leftarrow Inodes \cup (\mathbf{X}_{\text{refine}} \cap Inodes_{new})$.
$Bnodes \leftarrow Bnodes \cup (\mathbf{X}_{\text{refine}} \cap Bnodes_{new})$.

Use error indicator to locate the points need to be coarsen $\mathbf{X}_{\text{coarse}}$ in $\mathbf{X}_{k-1}$.
$\mathbf{X}^*_{k-1} \leftarrow \mathbf{X}_{k-1} \setminus \mathbf{X}_{\text{coarse}}$.
$\mathbf{X}_k \leftarrow \mathbf{X}^*_{k-1} \cup Inodes \cup Bnodes$.

if then $\mathbf{X}^k_{\text{refine}} \cup \mathbf{X}^k_{\text{coarse}} = \emptyset$
$endflag = flase$
end if
end while

To keep the growth of the condition number of interpolation matrix $\kappa(A)$ in a moderate pace, one needs to keep the minimum eigenvalue $\sigma_{\min}$ and the maximum eigenvalue $\sigma_{\max}$ in a moderate scale.

Gershgorin's theorem [68] states that

$$|\sigma_{\max} - A_{ii}| \leq \sum_{j=1, j \neq i}^{N} |A_{ij}| \tag{3.10}$$

for $\mathbf{X} \in \mathbb{R}^d$ and $\mathbf{X} = \{\mathbf{x}_i, i = 1, \cdots, N\}$. That is,

$$\sigma_{\max} \leq N \max_{i,j=1,\cdots,N} |A_{ij}| = N \max_{\mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}} \phi(||\mathbf{x}_i - \mathbf{x}_j||_2). \tag{3.11}$$

By the properties of positive definite functions,

$$\sigma_{\max} \leq N\phi(0). \tag{3.12}$$

Now, as long as the data $\mathbf{X}$ are not too wildly distributed, N grows as $h_{\mathbf{X},\Omega}^{-d}$, which is in a moderate pace and it is acceptable.

Then the remaining work for having a bound of $\kappa(A)$ lies in finding the lower bound for $\sigma_{\min}$. Different lower bound of $\sigma_{\min}$ for different RBF are summarised in [21].

For generalized multiquadric $\phi(\mathbf{x}) = (1 + c^2||\mathbf{x}||_2^2)^\beta, \beta \in \mathbb{R} \setminus \mathbb{N}_0$, one has

$$\sigma_{\min} \geq C(d, \beta, c) q_{\mathbf{X}}^{\beta - d/2 + 1/2} \exp(-2M_d/(q_{\mathbf{X}} c)), \tag{3.13}$$

where $C(d, \beta, c)$ and M_d are known constants. One sees that, for a fixed shape parameter c, the lower bound for $\sigma_{\min}$ goes to zero exponentially as the separation distance $q_{\mathbf{X}}$ decrease. Since one knows the condition number of interpolation matrix $\kappa(A)$ is the ratio of largest and smallest eigenvalues. Moreover, the growth of $\sigma_{\min}$ is of order N. Then, one sees that the condition number of interpolation matrix $\kappa(A)$ grows exponentially with decreasing separation distance. On the other hand, if one keeps the separation distance fixed (normally achieved by keeping the number of centers N fixed) and decreases the shape parameter c, then the condition number $\kappa(A)$ behaves in almost the same manner, which is exponential.

From the perspective of accuracy, one has the error bound for the multiquadric interpolant [65]

$$|f(\mathbf{x}) - S_{\mathbf{X}}^{\text{multi}}| \leq K\eta^{\frac{1}{ch_{\mathbf{X},\Omega}}}, \tag{3.14}$$

where K is a constant, c is the shape parameter, $h_{\mathbf{X},\Omega}$ is the fill distances and $0 < \eta < 1$. One sees that spectral convergence is achieved as either the fill distance or the shape parameter go to zero.

Thus, it leads to the uncertainty principle from [80], that is that the good accuracy is obtained by a small shape parameter and a small fill distance (centers close together) while the small condition number requires the shape parameter and separation distance be large (centers far apart). Apparently, both situations can not occur at the same time. Moreover, there is not an ideal method in practical and theoretical term for having the optimal shape parameter in applications. A lot of researches have been done and some useful methods have been proposed for selecting near optimal shape parameters [5, 9, 20, 23, 30, 32, 33, 60, 61, 62, 78].

Thus, in order to maintain the condition number of interpolation matrix $\kappa(A)$ and to achieve good approximation accuracy, an adaptive shape parameter is applied, increasing the shape parameter as the distance between the centers decreases.

In this chapter, the basis functions in the interpolant in 3.1 will be

$$\phi_i(r) = \sqrt{(1 + (c_i r)^2)}, \tag{3.15}$$

so that a different choice of shape parameter will be used at each point $\mathbf{x}, i = 1, \cdots, N$, which is called the adaptive shape parameters. Here the author introduces the two ways to have adaptive shape parameters :

- The shape parameter c_i of each center $\mathbf{x}_i$ is set to be a constant divided by the distance to the nearest neighbour center, that is

$$c_i = \frac{C}{\min(||\mathbf{x} - \mathbf{x}_i||)} \quad \mathbf{x} \in \mathbf{X}. \tag{3.16}$$

- In two and three dimensional cases, each center $\mathbf{x}$ in the initial centers set $\mathbf{X}_0$, which is at level $l = 0$, is assigned a shape parameter value c^0. Every center has 2^d attached indication points, which these indication points are at level $l = 1$. In order words, the centers are at level l, their indication

points are at level $l+1$. When ever an indication point ξ is converted into center $\mathbf{x}$, its shape parameter value is twice its parent's value. Figure 3.5 shows one example of centers and indication points at different levels, the black dots are centers and the red dots are indication points. Based on above description, the shape parameter for centers at level l is

$$c^l = 2^l c^0. \tag{3.17}$$

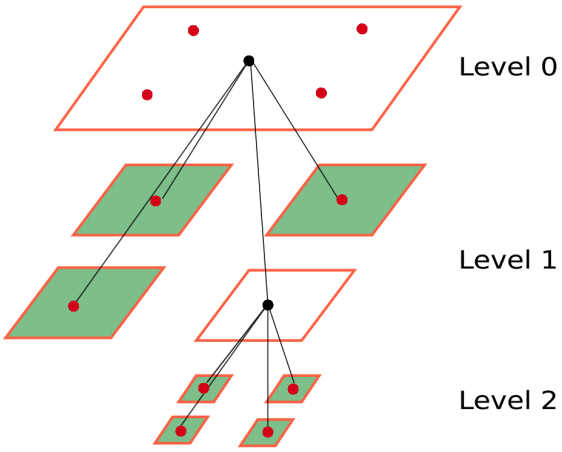

FIGURE 3.5: Examples of levels.

3.4 Adaptive procedure achieved by using an error indicator

In mind of above definitions and the relationship between $\mathbf{X}_k$, Ξ_k and N_ξ, the adaptive error indicator (AEI) RBF approximation for target function f is achieved by following procedure:

1. Center set $\mathbf{X}_k$ and its corresponding indication set Ξ_k are specified.
2. Global RBF approximation $S^{\text{multi}}_{\mathbf{X}_k}$ is generated on the center set $\mathbf{X}_k$, and the neighbourhood sets N_ξ for every ξ in Ξ_k are decided.

3. The local RBF approximation $S^{\text{ps}}_{N_\xi}$ is generated for each ξ, and the error indicator $\eta(\xi)$ is computed.

4. The refinement set $\mathbf{X}_{\text{refine}}$ and the coarse set $\mathbf{X}_{\text{coarse}}$ are generated by error indicator $\eta(.)$ according to the Definition 3.2.

5. The center set $\mathbf{X}_k$ is updated by adding the refinement set $\mathbf{X}_{\text{refine}}$ and deleting the coarse set $\mathbf{X}_{\text{coarse}}$ that is $\mathbf{X}_{k+1} = \{\mathbf{X}_k \cup \mathbf{X}_{\text{refine}}\} \setminus \mathbf{X}_{\text{coarse}}$.

6. When $\mathbf{X}_{\text{refine}} \cup \mathbf{X}_{\text{coarse}} = \emptyset$, the algorithm terminates. Otherwise return to the first step.

3.5 Numerical Results

In this section, the author shows the effectiveness of the error indicator in locating the worst errors, and in the corresponding refinement and coarsening strategy to improve the error. The author does this to one dimensional and two dimensional test functions and compare to results in the paper [17]. Three dimensional examples including the Perdichizzi case have been tested.

3.5.1 One dimensional function approximations

For one dimensional test functions, the author sets the initial center set $\mathbf{X}_0$ to be the uniformly distributed points in the interval $[-1, 1]$ and the indication set Ξ_0 is the set of midpoints of $\mathbf{X}_0$, as explained above. The neighbourhood set N_ξ contains $M = 4$ points. The shape parameter c of each center is set as described as in Equation 3.16, $C = 0.75$. The reason the author sets $C = 0.75$ is simply because this settings works well, this setting could also be find in [17]. The author also tried other settings, suck as $C = 0.5$, this setting also works. A test set T containing 5001 equally spaced nodes is used to test the approximation quality:

$e_{\mathbf{X}} = \max_{t \in T} |f(t) - S_{\mathbf{X}}^{\text{multi}}(t)|$ and the root mean square value in $e_{\mathbf{X}}$, that is

$$\text{RMS}(e_{\mathbf{X}}(f)) = \sqrt{\frac{1}{N_T} \sum_{i=1}^{N_T} |f(t_i) - S_{\mathbf{X}}^{\text{multi}}(t_i)|}. \tag{3.18}$$

3.5.1.1 The Runge function

The author first consider a standard approximation problem, the Runge function $f(x) = (1 + 25x^2)^{-1}$on $[-1, 1]$.

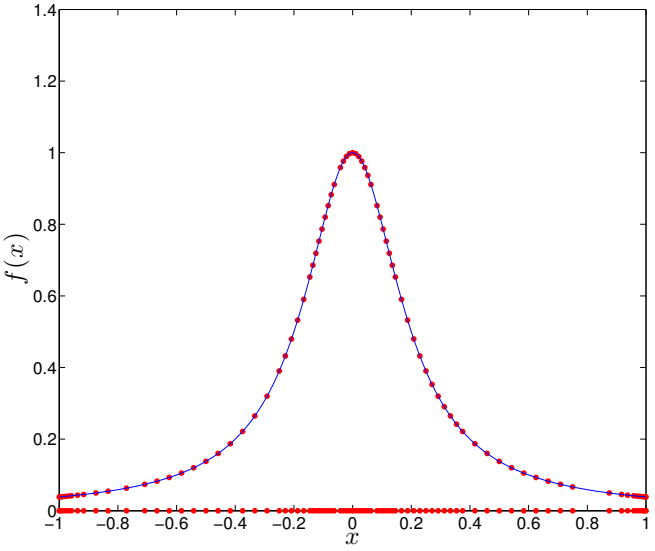

FIGURE 3.6: Runge function with final RBF centers distribution, initial $|\mathbf{X}| = 13$, $\theta_{\text{refine}} = 2.0(-5)$.

In Figure 3.6, one sees the final result obtained by adaptive interpolation, with $|\mathbf{X}| = 13$ initially, refinement threshold $\theta_{\text{refine}} = 2.0(-5) = 2 \times 10^{-5}$ and $\theta_{\text{coarse}} = \theta_{\text{refine}}/200$. One observes that centers cluster near the boundaries where approximation is more challenging due to the one-sided nature of the information, and at the origin, where the target function changes more rapidly. Note that the final maximum error is $1.4(-5)$ which is very close to θ_{refine} suggesting that the error indicator is working well. The largest condition number of during the iterative process in this case is $3.1(+6)$.

Table 3.1 presents the results of the adaptive process, which stops after 8 iterations. The final interpolant $S_{\mathbf{X}}^{\text{multi}}$ has 83 centers and the whole process computed a total of 85 evaluations of the target function. At each stage $N_{\text{refine}}, N_{\text{coarse}}$ are respectively the numbers of nodes to be added/removed from the center set, $\kappa(A)$ is the condition number of interpolation matrix A.

If one uses the full center set with 85 points to construct an interpolant one gets L-infinity and root mean square errors $1.4(-5)$ and $1.4(-6)$ respectively, a small improvement on the error with 83 centers.

It	N_{total}	$\lvert\mathbf{X}\rvert$	N_{coarse}	N_{refine}	$e_{\mathbf{X}}(f)$	$\text{RMS}(e_{\mathbf{X}}(f))$	$\kappa(A)$
1	13	13	0	12	1.2(-2)	5.1(-3)	3.2(+3)
2	25	25	0	22	4.9(-4)	1.3(-4)	1.2(+4)
3	47	47	0	16	1.1(-4)	1.7(-5)	1.0(+5)
4	63	63	0	13	5.6(-5)	6.3(-6)	2.6(+5)
5	76	76	0	5	2.8(-5)	3.3(-6)	5.1(+5)
6	81	81	2	3	2.1(-5)	3.4(-6)	1.0(+6)
7	84	82	0	1	1.3(-5)	2.5(-6)	3.0(+6)
8	85	83	0	0	1.4(-5)	2.6(-6)	3.1(+6)

TABLE 3.1: Iterative process of adaptive error indicator interpolation of Runge function with $\theta_{\text{refine}} = 2(-5)$.

Table 3.2 shows results according to different θ_{refine}, $\kappa(A)_{\max}$ means the largest $\kappa(A)$ during the adaptive process.

θ_{refine}	N_{total}	$\lvert\mathbf{X}\rvert$	$e_{\mathbf{X}}(f)$	$\text{RMS}(e_{\mathbf{X}}(f))$	$\kappa(A)_{\max}$
2.0(-5)	85	83	1.4(-5)	2.6(-6)	3.1(+6)
2.0(-6)	155	151	1.6(-6)	1.4(-7)	2.4(+7)
2.0(-7)	276	244	2.4(-7)	3.5(-8)	2.7(+9)
2.0(-8)	710	646	1.8(-8)	3.5(-9)	1.1(+11)

TABLE 3.2: Adaptive error indicator interpolation results of Runge function with different θ_{refine}.

Figure 3.7 shows how the error decreases with the number of points in the set $\mathbf{X}$, starting at 13, and finishing with 646 centers, N_{total} is the total centers that were sampled from target function, staring at 13 and finishing with 710. The final interpolant $S_{\mathbf{X}}^{\text{multi}}$ used 646 centers, with $e_{\mathbf{X}}(f) = 1.8(-8)$ and $\text{RMS}(e_{\mathbf{X}}(f)) = 3.5(-9)$. Using all the 710 centers, the interpolant $S_{N_{\text{total}}}^{\text{multi}}$ provide $e_{N_{\text{total}}}(f) =$

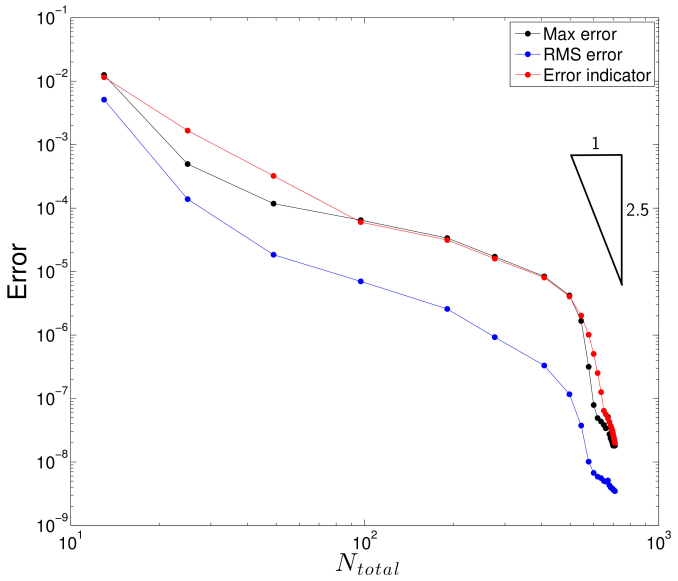

FIGURE 3.7: Runge function interpolation error for each iteration, N_{total} is the total number of samples of the target function, with $\theta_{\text{refine}} = 2(-8)$.

$1.8(-8)$ and $\text{RMS}(e_{N_{\text{total}}}(f)) = 3.3(-9)$. The red nodes in Figure 3.7 are the maximum values of the error indicator function at each iteration in absolute value, so one can see that the error indicator is a good measure of approximation error because the measured error (red line) tracks the approximation error (black line). One sees a convergence rate $|e_{\mathbf{X}}(f)_{|\mathbf{X}|}| < C \times |\mathbf{X}|^{-2.5}$.

The condition numbers at each iteration is below $1.1(+11)$ due to the application of this adaptive shape parameter strategy. If one uses the adaptive interpolation algorithm with a constant shape parameter in this example, the condition number of the interpolation matrix increases to $5(+20)$ after one or two iterations. While it has been observed that good approximation can be maintained with very large condition numbers, any user would, quite reasonably, doubt that such poor conditioning could lead to reliable results. The goal is to provide answers that users can trust.

In [17], Driscoll and Heryudono use the residual sub-sampling method on the same example. They record the number of centers $\mathbf{X}$ used in the final interpolant, but the total numbers of function samples computed from the target function is not reported. In Table 3.3, RS stands for residual sub-sampling method and

AEI stands for adaptive error indicator method, it compares the results and the resources needed, the N_{total} for residual sub-sampling method as implemented by the author. One can see that residual sub-sampling achieves a better result marginally, but with a much larger number of function evaluations. The author emphasises that this applications include examples where function evaluation is expensive.

Method	$e_{\mathbf{X}}(f)$	$\lvert\mathbf{X}\rvert$	N_{total}
RS	1.3(-5)	53	285
AEI	1.4(-5)	83	85
RS	8.9(-8)	284	4595
AEI	1.8(-8)	646	710

TABLE 3.3: Error indicator vs Residual sub-sampling for Rnuge function

3.5.1.2 The hyperbolic tan function

In this example, target function $f(x) = \tanh(60x - 0.1)$ is considered. Figure 3.8 shows the result of the adaptive error indicator approximation of this function with $\theta_{\text{refine}} = 2.0(-5)$, one sees how the error indicator distributes centers around the steepest part of f.

Table 3.4 shows the adaptive process of interpolation with threshold $\theta_{\text{refine}} = 2.0(-5)$. The adaptive approximation converges in 9 iterations with final 82 nodes selected from 141 centers at which this method computes the target function.

The final interpolant $S^{\text{multi}}_{\mathbf{X}}$ has error $e_{\mathbf{X}}(f) = 1.1(-5)$ and and $\text{RMS}(e_{\mathbf{X}}(f)) = 1.8(-6)$. Using all the 141 centers, the interpolant $S^{\text{multi}}_{N_{\text{total}}}$ provide $e_{N_{\text{total}}}(f) = 3.2(-6)$ and $\text{RMS}(e_{N_{\text{total}}}(f)) = 1.3(-7)$. Thus, depending on the user, one can have a more compact representation of the target function, guided by θ_{refine} and θ_{coarse}, or for a more accurate approximation using all points at which the target has been evaluated. The condition number grows fast during the first few iterations, but never grows too large.

Table 3.5 shows results according to different θ_{refine}, one could see the final results $e_{\mathbf{X}}(f)$ are close to the threshold θ_{refine}.

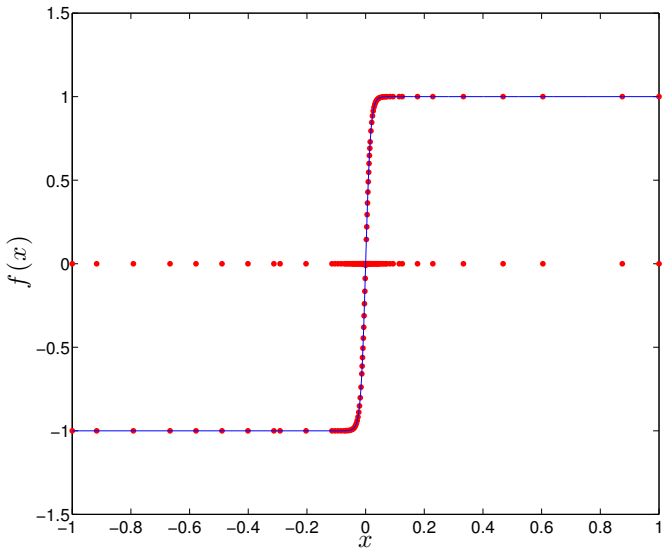

FIGURE 3.8: Graph of $f(x) = \tanh(60x - 0.1)$ function with final RBF centers distribution produced by the algorithm, with initial $|\mathbf{X}| = 13$, $\theta_{\text{refine}} = 2.0(-5)$. The final number of centers used is 82.

It	N_{total}	$\lvert\mathbf{X}\rvert$	N_{coarse}	N_{refine}	$e_{\mathbf{X}}(f)$	$\text{RMS}(e_{\mathbf{X}}(f))$	$\kappa(A)$
1	13	13	0	12	7.2(-1)	1.7(-1)	3.2(+3)
2	25	25	0	22	5.3(-1)	9.7(-2)	1.2(+4)
3	47	47	6	22	2.6(-1)	3.7(-2)	4.7(+4)
4	69	63	25	20	5.3(-2)	6.1(-3)	4.5(+5)
5	89	58	26	17	2.1(-3)	1.7(-4)	2.3(+6)
6	106	49	0	24	1.5(-4)	6.1(-5)	2.2(+7)
7	130	73	1	10	1.2(-5)	2.1(-6)	7.1(+6)
8	140	82	1	1	1.1(-5)	1.8(-6)	9.3(+6)
9	141	82	0	0	1.1(-5)	1.8(-6)	9.7(+6)

TABLE 3.4: Iterative process of adaptive algorithm interpolation of $\tanh(60x - 0.1)$, with $\theta_{\text{refine}} = 2(-5)$.

θ_{refine}	N_{total}	$\lvert\mathbf{X}\rvert$	$e_{\mathbf{X}}(f)$	$\text{RMS}(e_{\mathbf{X}}(f))$	$\kappa(A)_{\max}$
2.0(-5)	141	82	1.1(-5)	1.8(-6)	2.2(+7)
2.0(-6)	232	141	1.8(-6)	2.4(-7)	2.6(+7)
2.0(-7)	320	238	2.0(-7)	3.7(-8)	1.1(+9)
2.0(-8)	726	595	1.7(-8)	2.1(-9)	8.1(+9)

TABLE 3.5: Adaptive error indicator interpolation results of $\tanh(60x - 0.1)$ function with different θ_{refine}.

Figure 3.9 shows the adaptive process with $\theta_{\text{refine}} = 2(-8)$, starting with 13 centers. The algorithm stops with $|\mathbf{X}| = 595$ and $N_{\text{total}} = 726$ in 38 iterations. The final interpolant $S^{\text{multi}}_{\mathbf{X}}$ has $e_{\mathbf{X}}(f) = 1.7(-8)$ and $\text{RMS}(e_{\mathbf{X}}(f)) = 2.1(-9)$, while the interpolant using all the available centers $S^{\text{multi}}_{N_{\text{total}}}$ gives L-infinity and root mean square errors $3.4(-8)$ and $9.8(-10)$ respectively. The condition number at each iteration is below $8.1(+9)$. One still see a convergence rate $|e_{\mathbf{X}}(f)_{|\mathbf{X}|}| < C \times |\mathbf{X}|^{-2.5}$.

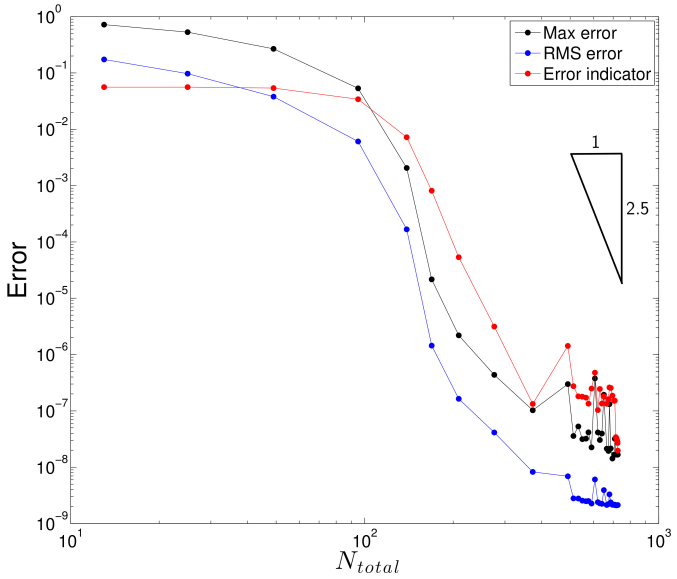

FIGURE 3.9: $\tanh(60x - 0.1)$ function interpolation error for each iteration, N_{total} is the total number of samples of the target function, with $\theta_{\text{refine}} = 2(-8)$.

Table 3.6 compares the results and the total number of function evaluations needed for the error indicator algorithm and residual sub-sampling algorithm. In this example the AEI algorithm achieves a better result with significantly less function evaluation.

Method	$e_{\mathbf{X}}(f)$	$\lvert\mathbf{X}\rvert$	N_{total}
RS	2.5(-5)	129	441
AEI	1.1(-5)	82	141
RS	1.7(-7)	185	5617
AEI	1.7(-8)	595	726

TABLE 3.6: Error indicator vs Residual sub-sampling for function $\tanh(60x - 0.1)$.

3.5.1.3 The shifted absolute value function

The final univariate example is $f(x) = |x - 0.04|$. Figure 3.10 shows the center distributed around the derivative discontinuity of $|x - 0.04|$. The final RBF representation uses 44 centers. Table 3.7 shows the adaptive process starting with 13 uniformly distributed centers and ending with 44 centers. The total number of function evaluations was 121. The final interpolant $S_{\mathbf{X}}^{\text{multi}}$ has L-infinity and root mean square errors $3.8(-5)$ and $2.7(-6)$ respectively, while using all the 121 centers one obtains uniform and root mean square errors $3.8(-5)$ and $6.0(-7)$ respectively.

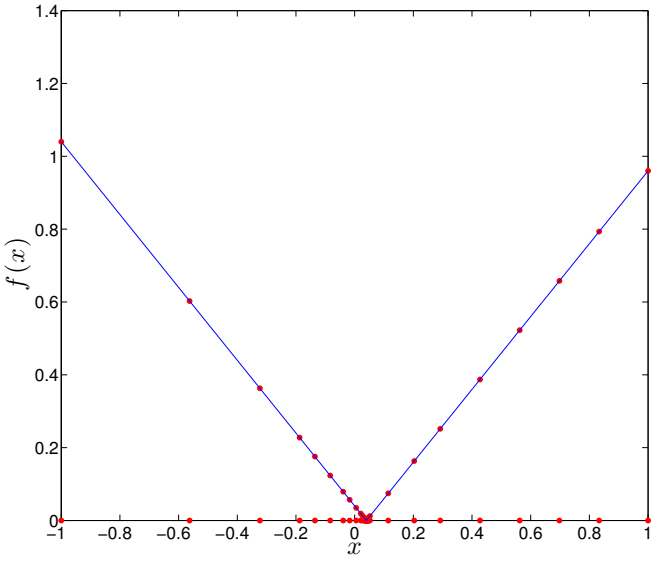

FIGURE 3.10: Final center distribution (44 points) for approximating $f(x) = |x - 0.04|$ with $\theta_{\text{refine}} = 2.0(-5)$.

Table 3.8 shows results according to different θ_{refine}, one could see the final results $e_{\mathbf{X}}(f)$ are still close to the threshold θ_{refine} in this non-smooth function.

Figure 3.11 shows the progress of the adaptive algorithm starting with 13 centers, and $\theta_{\text{refine}} = 2.0(-8)$. The algorithm terminates after 27 iterations with $|\mathbf{X}| = 81$. The total number of points used N_{total} starts at 13 and stops at 459. The final interpolant $S_{\mathbf{X}}^{\text{multi}}$ with 81 centers has maximum error $e_{\mathbf{X}}(f) = 3.8(-8)$

It	N_{total}	$\lvert\mathbf{X}\rvert$	N_{coarse}	N_{refine}	$e_{\mathbf{X}}(f)$	RMS($e_{\mathbf{X}}(f)$)	$\kappa(A)$
1	13	13	0	12	3.7(-2)	6.3(-3)	3.2(+3)
2	25	25	0	16	2.7(-2)	3.0(-3)	1.2(+4)
3	41	41	3	14	7.1(-3)	9.8(-4)	5.3(+4)
4	55	52	18	12	3.3(-3)	2.9(-4)	3.4(+5)
5	67	46	9	15	1.6(-3)	2.1(-4)	1.1(+8)
6	82	52	23	6	1.4(-3)	4.1(-5)	9.2(+6)
7	88	35	6	14	7.7(-4)	9.2(-5)	1.3(+7)
8	102	43	6	5	3.3(-4)	6.3(-6)	2.0(+7)
9	107	42	5	10	1.1(-3)	4.2(-4)	1.0(+9)
10	117	47	3	4	5.2(-5)	2.1(-6)	6.2(+7)
11	121	48	4	0	3.8(-5)	1.8(-6)	8.6(+7)
12	121	44	0	0	3.8(-5)	2.7(-6)	7.5(+7)

TABLE 3.7: Iterative process of adaptive algorithm interpolation of $|x - 0.04|$, with $\theta_{\text{refine}} = 2(-5)$.

θ_{refine}	N_{total}	$\lvert\mathbf{X}\rvert$	$e_{\mathbf{X}}(f)$	RMS($e_{\mathbf{X}}(f)$)	$\kappa(A)_{\max}$
2.0(-5)	121	44	3.8(-5)	2.7(-6)	1.0(+9)
2.0(-6)	266	56	1.6(-6)	1.5(-7)	9.0(+10)
2.0(-7)	352	66	7.6(-7)	4.3(-8)	8.8(+11)
2.0(-8)	459	81	3.8(-8)	5.3(-9)	2.1(+13)

TABLE 3.8: Adaptive error indicator interpolation results of $f(x) = |x - 0.04|$ function with different θ_{refine}.

and RMS($e_{\mathbf{X}}(f)$) = 5.3(−9). The condition number at each iteration is below 2.1(+13). The interpolant using all the available centers $S^{\text{multi}}_{N_{\text{total}}}$ gives L-infinity error 3.8(−8) and the root mean square error 5.3(−10). The condition numbers for this interpolation is 1.8(+13). One still sees a convergence rate $|e_{\mathbf{X}}(f)_{|\mathbf{X}|}| < C \times |\mathbf{X}|^{-2.5}$.

Table 3.9 compares the results and function evaluations required for the error indicator algorithm and the residual sub-sampling algorithm.

Method	$e_{\mathbf{X}}(f)$	$\lvert\mathbf{X}\rvert$	N_{total}
RS	1.5(-5)	53	422
AEI	1.6(-6)	56	266
RS	4.1(-8)	74	2683
AEI	3.8(-8)	81	459

TABLE 3.9: Error indicator vs Residual sub-sampling for function $|x - 0.04|$.

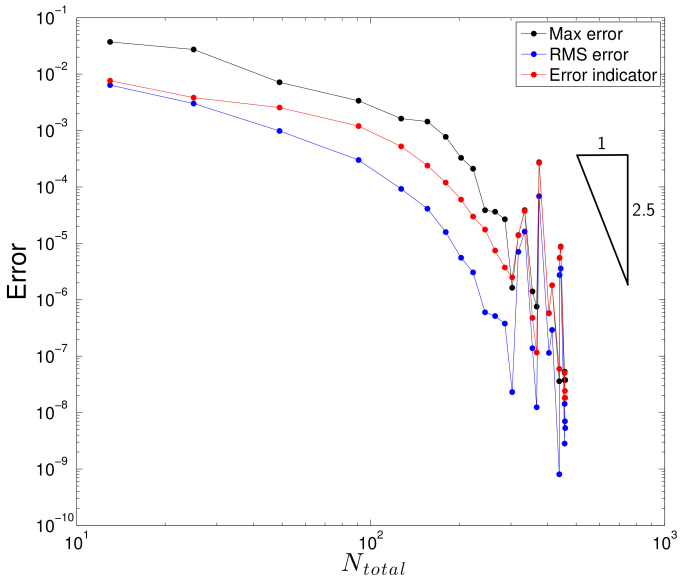

FIGURE 3.11: Error convergence of the adaptive algorithm for $f(x) = |x - 0.04|$ and $\theta_{\text{refine}} = 2.0(-8)$.

For these three approximation problems, one sees from the results that the shifted absolute value function $f(x) = |x - 0.04|$ is more difficult to approximate than the Runge function and the hyperbolic tan function $f(x) = \tanh(60x - 0.1)$. Table 3.10 summaries the $\kappa(A)_{\max}$ for these three functions with different θ_{refine}.

The author observes that the $\kappa(A)_{\max}$ for the shifted absolute value function is larger than the one from other two functions for the same θ_{refine}. That is the $\kappa(A)_{\max}$ might could be considered as an evaluation of the difficulty of the problem in one dimensional cases, the more difficult the approximation problem is, the larger the $\kappa(A)_{\max}$ becomes.

	$\lvert x - 0.04\rvert$	$\tanh(60x - 0.1)$	$(1 + 25x^2)^{-1}$
$\kappa(A)_{\max}, \theta_{\text{refine}} = 2.0(-5)$	1.0(+9)	2.2(+7)	3.1(+6)
$\kappa(A)_{\max}, \theta_{\text{refine}} = 2.0(-6)$	9.0(+10)	2.6(+7)	2.4(+7)
$\kappa(A)_{\max}, \theta_{\text{refine}} = 2.0(-7)$	8.8(+11)	1.1(+9)	2.7(+9)
$\kappa(A)_{\max}, \theta_{\text{refine}} = 2.0(-8)$	2.1(+13)	8.1(+9)	1.1(+7)

TABLE 3.10: $\kappa(A)_{\max}$ for three one dimensional examples.

3.5.2 Two dimensional function approximations

Now three two-dimensional examples are considered, where the node refinement scheme explained above is applied. The author sets $j = 3$ in the initialisation step of Algorithm 1 to achieve the initial center set $\mathbf{X}_0$, with $|\mathbf{X}_0| = 100$, and its indication set Ξ_0. The neighbourhood set N_ξ has $M = 24$ neighbours. A test grid T of 101×101 uniformly spaced nodes on $[-1, 1]^2$ is used to test the approximation quality: $e_{\mathbf{X}}(f) = \max_{t \in T} |f(t) - S^{\text{multi}}_{\mathbf{X}}(t)|$ and the root mean square value in $e_{\mathbf{X}}$, that is $\text{RMS}(e_{\mathbf{X}}(f))$ as Equation 3.18. The shape parameter c of each center is set as described as in Equation 3.16, $C = 0.5$, and $\theta_{\text{coarse}} = \theta_{\text{refine}}/100$.

3.5.2.1 The modified Franke Function

This test function (Figure 3.12)

$$f(x,y) = \exp^{-0.1(x^2+y^2)} + \exp^{-5((x-0.5)^2+(y-0.5)^2)} + \exp^{-15((x+0.2)^2+(y+0.4)^2)} + \exp^{-9((x+0.8)^2+(y-0.8)^2)} \quad (3.19)$$

is a standard test function for a RBF approximation. The author adopts this test function from Driscoll and Heryudono [17], they called this function modified Franke function. With $\theta_{\text{refine}} = 5.0(-4)$ only 14 iterations are needed to reach the stopping criteria. The process is showed in Table 3.11. In this case one has $|\mathbf{X}| = 1318$ centers with L-infinity error $7.2(-4)$ and in the process all the condition numbers are below $2.1(+7)$.

Figure 3.13 shows the final node distribution and demonstrates that the error indicator locates points in regions of rapid variation.

Table 3.12 shows results according to different θ_{refine}, $\kappa(A)_{\max}$ means the largest $\kappa(A)$ during the adaptive process.

Figure 3.14 shows how the maximum and root mean square error decrease with the pre-set threshold, and the number of points required to achieve the given threshold. The red line stands the maximum error, the black line stands for the

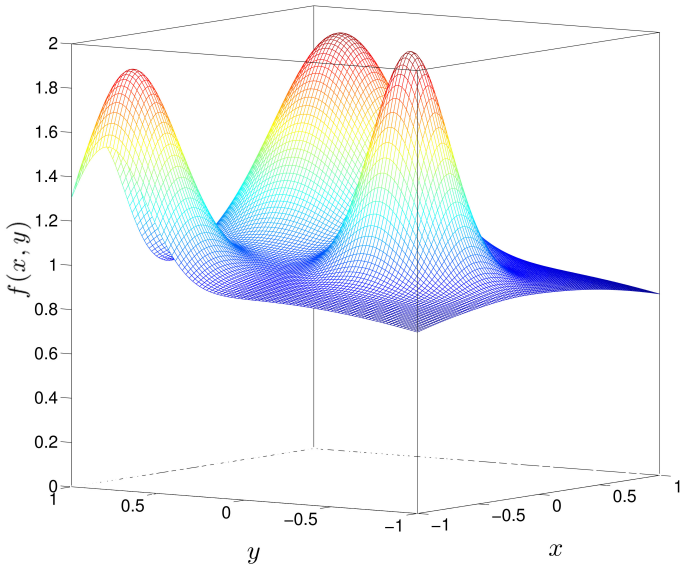

FIGURE 3.12: modified Franke function interpolation.

It	N_{total}	$\lvert\mathbf{X}\rvert$	N_{coarse}	N_{refine}	$e_{\mathbf{X}}(f)$	RMS$(e_{\mathbf{X}}(f))$	$\kappa(A)$
1	100	100	0	280	4.9(-2)	8.9(-3)	1.5(+6)
2	380	380	1	370	2.1(-3)	2.3(-4)	2.5(+6)
3	750	749	0	185	2.2(-3)	1.2(-4)	6.4(+6)
4	935	934	0	153	1.3(-3)	1.0(-4)	1.0(+7)
5	1088	1087	0	98	1.4(-4)	8.2(-5)	1.5(+7)
6	1186	1185	0	65	7.4(-5)	7.1(-5)	1.7(+7)
7	1251	1250	0	42	7.2(-5)	6.5(-5)	1.9(+7)
8	1293	1292	0	13	7.2(-5)	6.4(-5)	2.1(+7)
9	1306	1305	0	5	7.2(-5)	6.4(-5)	2.1(+7)
10	1311	1310	0	3	7.2(-5)	6.4(-5)	2.1(+7)
11	1314	1313	0	3	7.2(-5)	6.4(-5)	2.1(+7)
12	1317	1316	0	1	7.2(-5)	6.4(-5)	2.1(+7)
13	1318	1317	0	1	7.2(-5)	6.4(-5)	2.1(+7)
14	1319	1318	0	0	7.2(-5)	6.4(-5)	2.1(+7)

TABLE 3.11: Iterative process of AEI interpolation of modified Franke function, with $\theta_{\text{refine}} = 5.0(-4)$.

pre-set threshold θ_{refine} and the blue line stands for the root mean square error. One sees a convergence rate $|e_{\mathbf{X}}(f)_{|\mathbf{X}|}| < C \times |\mathbf{X}|^{-1}$.

In Figure 3.15, the author compares the adaptive error indicator (AEI) interpolation and residual sub-sampling (RE) method on approximating the modified Franke function. One could see clearly that AEI method could reach the desirable

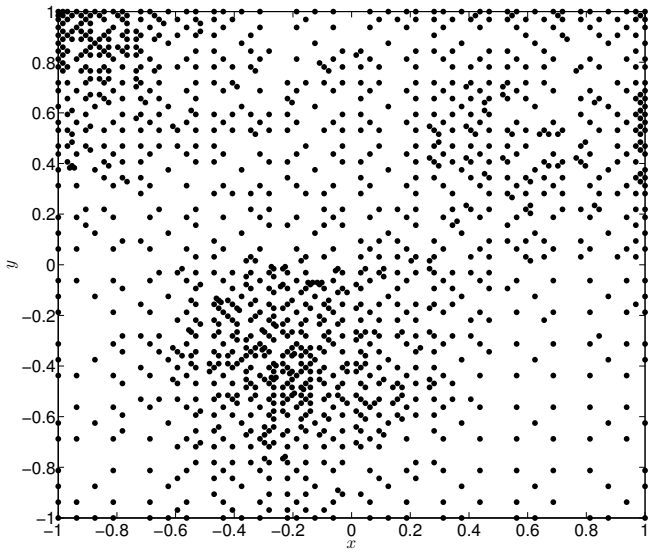

FIGURE 3.13: Centre distribution for adaptive interpolation for the modified Franke function with $\theta_{\text{refine}} = 5.0(-4)$. The number of points in this centre set is 1318.

θ_{refine}	N_{total}	$\lvert\mathbf{X}\rvert$	$e_{\mathbf{X}}(f)$	RMS$(e_{\mathbf{X}}(f))$	$\kappa(A)_{\max}$
1.0(-3)	697	697	2.2(-3)	1.7(-4)	6.1(+8)
7.5(-4)	907	907	1.4(-3)	9.9(-5)	1.1(+7)
5.0(-4)	1319	1318	7.2(-4)	6.3(-5)	2.1(+7)
2.5(-4)	2703	2702	6.3(-4)	3.8(-5)	7.6(+7)
1.0(-4)	6693	6692	2.1(-4)	1.3(-5)	3.4(+8)
7.5(-5)	8823	8820	1.1(-4)	8.4(-6)	5.8(+8)

TABLE 3.12: Adaptive algorithm interpolation results of modified Franke function with different θ_{refine}.

accuracy with notable less evaluations from the target function.

3.5.2.2 The two-dimensional hyperbolic function

The second test function is $f(x, y) = 0.4 \tanh(20xy) + 0.6$ (Figure 3.16) on $[-1, 1]^2$. With $\theta_{\text{refine}} = 5.0(-4)$ the algorithm took 8 iterations to reach the stopping criteria, in Table 3.13, it shows the iterative process. A total of $|\mathbf{X}| = 2106$ centers were used to give an error $e_{\mathbf{X}}(f) = 5.4(-5)$, Figure 3.17 shows the error indicator η has the ability to locate more centers in the rapid variation sub-domain.

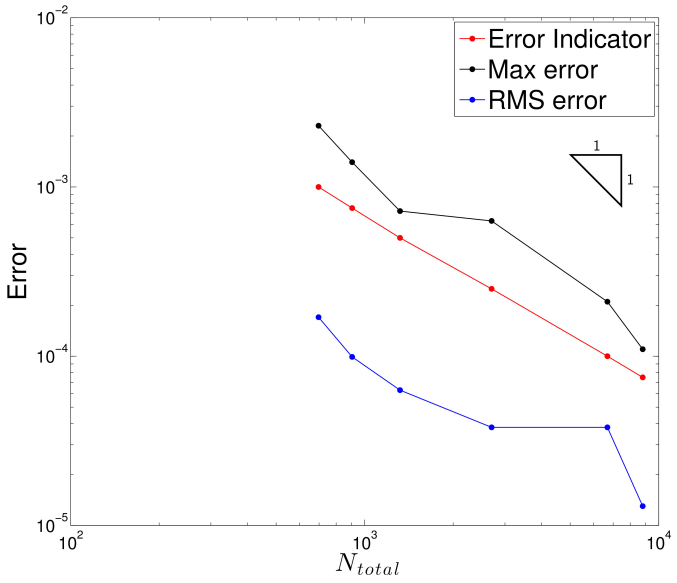

FIGURE 3.14: Convergence of adaptive error indicator interpolation for the modified Franke function.

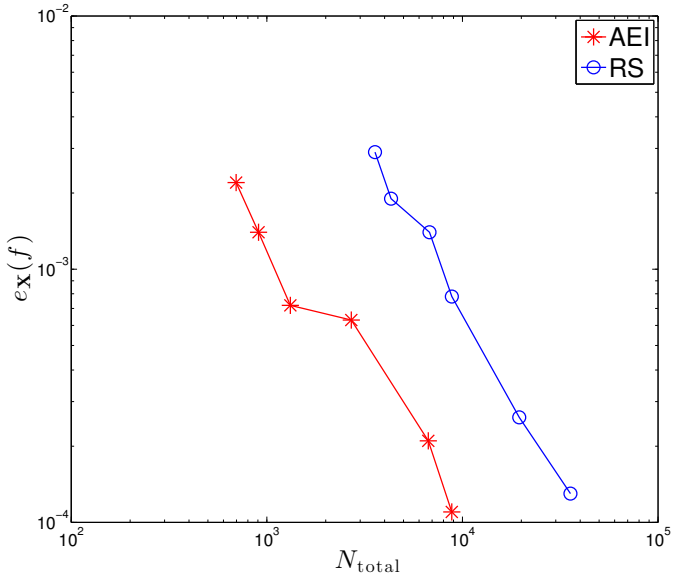

FIGURE 3.15: Comparison between AEI method and RE method for the modified Franke function.

Table 3.14 shows how the number of points needed by the algorithm varies with the choice of θ_{refine}. The higher level of accuracy is required, the larger N_{total} becomes.

Figure 3.18 shows how the maximum and root mean square error decrease with

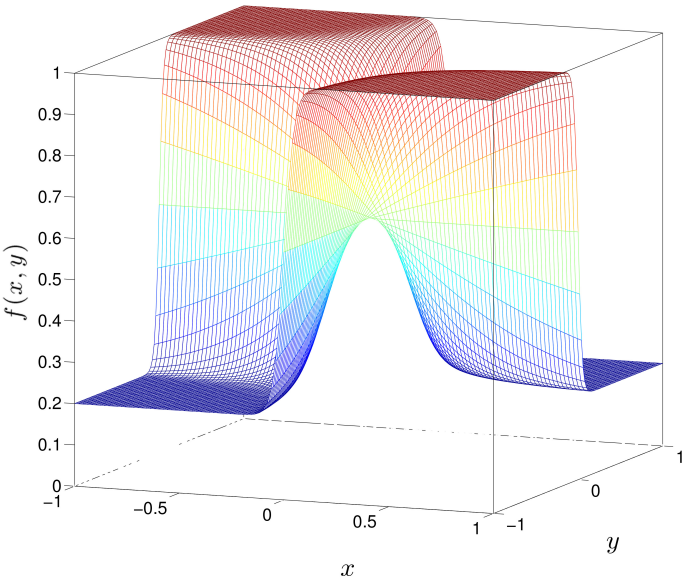

FIGURE 3.16: Interpolation of $f(x,y) = 0.4\tanh(20xy) + 0.6$, with $\theta_{\text{refine}} = 5.0(-4)$.

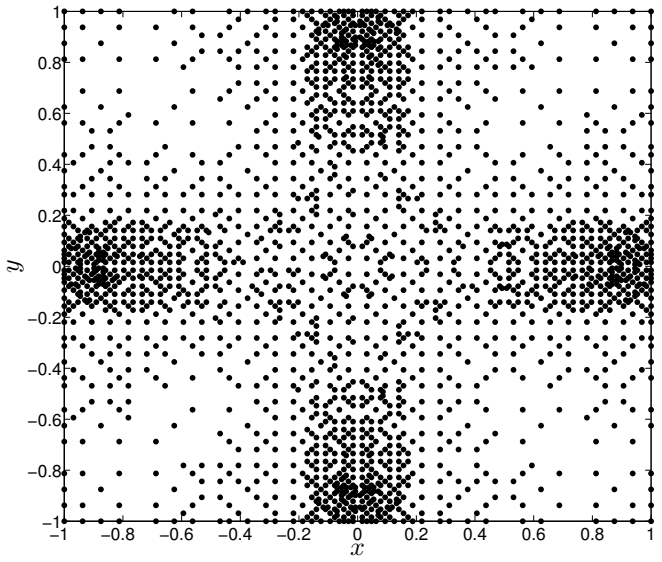

FIGURE 3.17: Final node distribution for approximation of $f(x,y) = 0.4\tanh(20xy) + 0.6$ with $\theta_{\text{refine}} = 5.0(-4)$.

the pre-set threshold, and the number of points required to achieve the given threshold. The red line stands the maximum error, the black line stands for the pre-set threshold θ_{refine} and the blue line stands for the root mean square error. One again sees the same convergence rate $|e_{\mathbf{X}}(f)_{|\mathbf{X}|}| < C \times |\mathbf{X}|^{-1}$.

It	N_{total}	$\lvert\mathbf{X}\rvert$	N_{coarse}	N_{refine}	$e_{\mathbf{X}}(f)$	RMS$(e_{\mathbf{X}}(f))$	$\kappa(A)$
1	100	100	0	304	1.2(-1)	4.1(-2)	1.5(+6)
2	404	404	0	720	4.3(-2)	6.6(-3)	2.8(+6)
3	1124	1124	48	666	7.2(-3)	5.9(-4)	1.4(+7)
4	1790	1742	11	298	1.8(-3)	1.1(-4)	3.6(+7)
5	2088	2029	2	53	1.2(-3)	6.7(-5)	5.1(+7)
6	2141	2080	7	26	5.5(-4)	5.7(-5)	5.4(+7)
7	2167	2099	2	9	5.5(-4)	5.7(-5)	5.4(+7)
8	2176	2106	0	0	5.5(-4)	5.7(-5)	5.5(+7)

TABLE 3.13: Iterative process of AEI interpolation of $f(x, y) = 0.4\tanh(20xy)+0.6$ function, with $\theta_{\text{refine}} = 5.0(-4)$.

θ_{refine}	N_{total}	$\lvert\mathbf{X}\rvert$	$e_{\mathbf{X}}(f)$	RMS$(e_{\mathbf{X}}(f))$	$\kappa(A)_{\max}$
1.0(-3)	1605	1561	4.2(-3)	1.6(-4)	2.9(+7)
7.5(-4)	1810	1776	2.8(-3)	1.1(-4)	3.7(+7)
5.0(-4)	2176	2106	5.4(-4)	5.6(-5)	5.5(+7)
2.5(-4)	3911	3840	2.3(-4)	2.8(-5)	1.4(+8)
1.0(-4)	9168	9080	1.2(-4)	1.1(-5)	7.5(+8)
7.5(-5)	12144	12078	1.4(-4)	9.4(-6)	1.4(+9)

TABLE 3.14: Adaptive algorithm interpolation results of $f(x, y) = -0.4\tanh(20xy) + 0.6$ with different θ_{refine}.

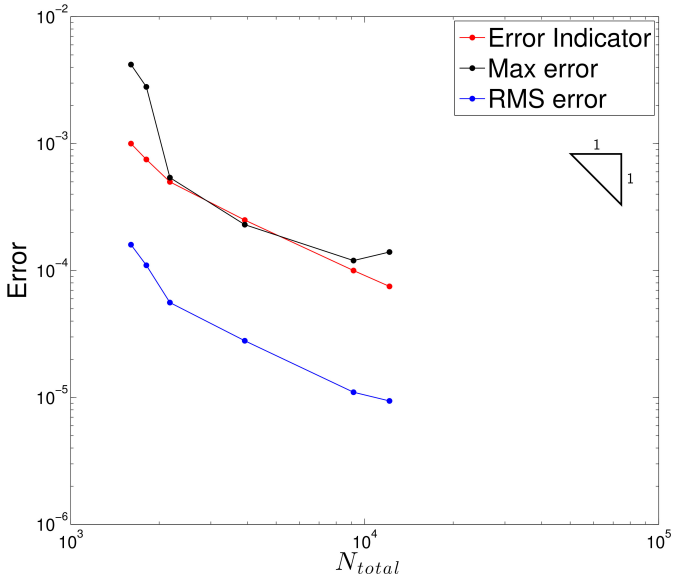

FIGURE 3.18: Convergence of adaptive error indicator interpolation for the $f(x, y) = -0.4\tanh(20xy) + 0.6$ function.

In Figure 3.19, the author compare the adaptive error indicator (AEI) interpolation and residual sub-sampling (RE) method on approximating this target function. One again sees clearly that AEI method could reach the desirable accuracy with notable less evaluations from the target function.

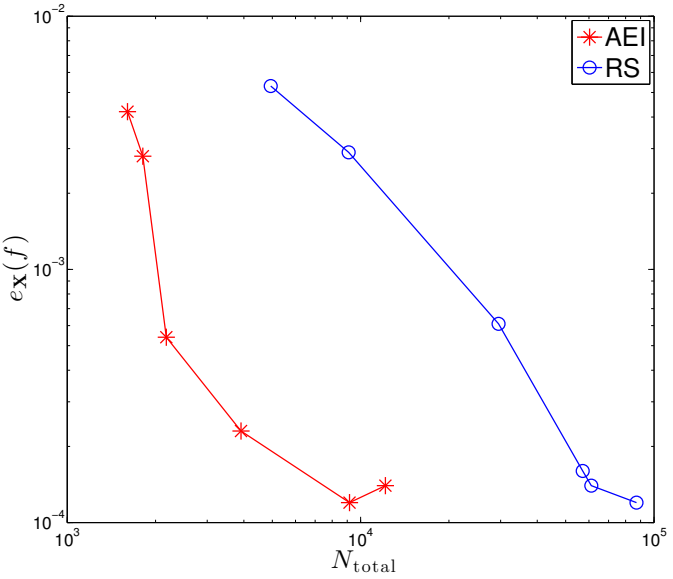

FIGURE 3.19: Comparison between AEI method and RE method for the $f(x, y) = -0.4 \tanh(20xy) + 0.6$ function.

The observant reader will notice that the final error may not decrease with the choice of the error indicator, since having a decrease in θ_{refine} in the last two rows of Table 3.14, but an increase in maximum error. This may happen as the indicator one uses is only that - an indicator. However, one could observe that the trend is decreasing which the mean root square error is always decaying, so that in a global sense a decrease of the threshold results in a decrease in errors.

3.5.2.3 The two-dimensional exponential function

In this example target function is $f(x, y) = \exp(-60((x-0.35)^2+(y-0.25)^2))+0.2$ (Figure 3.20) in $[-1, 1]^2$. Table 3.15 shows the approximation process, and in Figure 3.21 shows how the error indicator puts more centers in the region where the function changes rapidly.

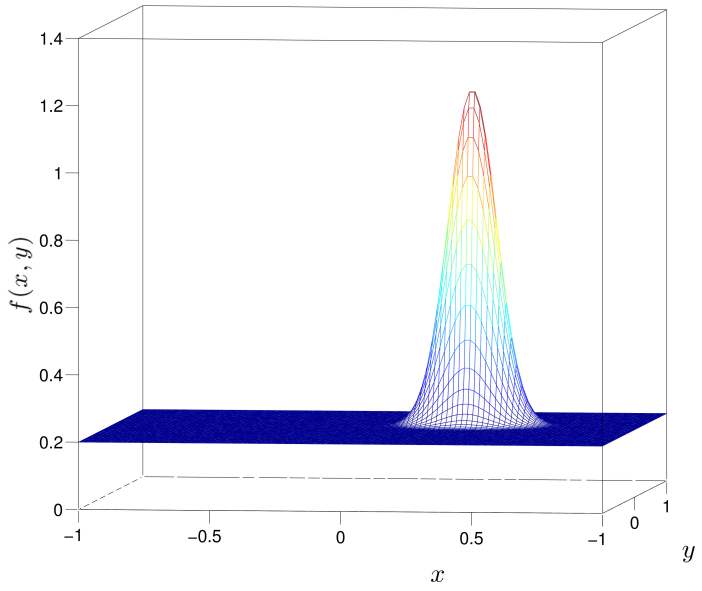

FIGURE 3.20: Interpolation of $f(x,y) = \exp(-60((x-0.35)^2+(y-0.25)^2))+0.2$, with $\theta_{\text{refine}} = 5.0(-4)$.

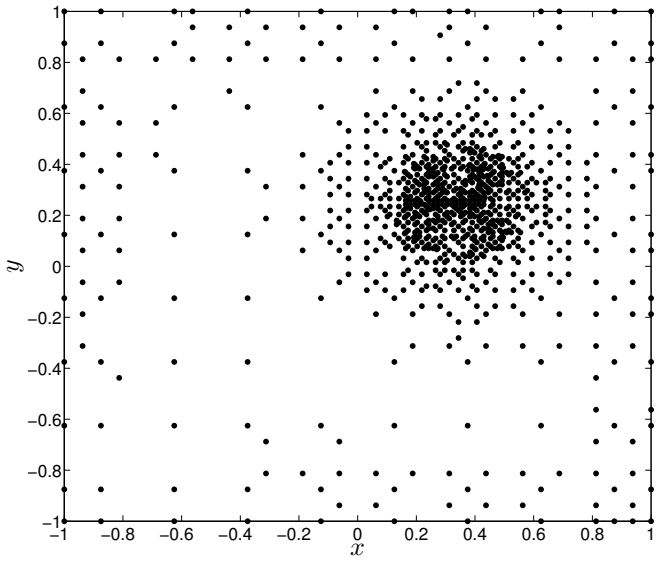

FIGURE 3.21: Final node distribution for approximation of $f(x,y) = \exp(-60((x-0.35)^2+(y-0.25)^2))+0.2$, with $\theta_{\text{refine}} = 5.0(-4)$.

Table 3.16 shows how the error of adaptive interpolation depend on the error indicator. In the two previous examples, there is no big difference in $|\mathbf{X}|$ and N_{total}. In this example, there is notable difference between $|\mathbf{X}|$ and N_{total}. With $\theta_{\text{refine}} = 7.5(-5)$, when using all the available centers to construct $S^{\text{multi}}_{N_{\text{total}}}$ one gets

It	N_{total}	$\lvert\mathbf{X}\rvert$	N_{coarse}	N_{refine}	$e_{\mathbf{X}}(f)$	RMS($e_{\mathbf{X}}(f)$)	$\kappa(A)$
1	100	100	0	224	5.3(-1)	4.1(-2)	1.5(+6)
2	324	324	9	247	8.8(-2)	5.7(-3)	1.9(+6)
3	571	562	122	153	5.7(-4)	4.2(-5)	4.0(+6)
4	724	593	10	68	2.6(-4)	3.3(-5)	5.7(+6)
5	792	651	3	67	2.6(-4)	3.4(-5)	7.5(+6)
6	859	715	0	59	2.6(-4)	3.3(-5)	1.1(+7)
7	918	774	0	66	2.6(-4)	3.3(-5)	1.3(+7)
8	984	840	1	43	2.6(-4)	3.3(-5)	1.6(+7)
9	1027	882	0	18	2.6(-4)	3.3(-5)	1.4(+7)
10	1045	900	0	13	2.6(-4)	3.3(-5)	1.5(+7)
11	1058	913	0	12	2.6(-4)	3.3(-5)	1.5(+7)
12	1070	925	0	4	2.6(-4)	3.3(-5)	1.4(+7)
13	1074	929	0	4	2.6(-4)	3.3(-5)	1.4(+7)
14	1078	933	0	0	2.6(-4)	3.3(-5)	1.4(+7)

TABLE 3.15: Iterative process of AEI approximation of $f(x, y) = \exp(-60((x - 0.35)^2 + (y - 0.25)^2)) + 0.2$ with $\theta_{\text{refine}} = 5.0(-4)$.

better approximation quality with $8.4(-5)$ and $9.8(-6)$ respectively for uniform and root mean square error.

θ_{refine}	N_{total}	$\lvert\mathbf{X}\rvert$	$e_{\mathbf{X}}(f)$	RMS($e_{\mathbf{X}}(f)$)	$\kappa(A)_{\max}$
1.0(-3)	594	476	2.6(-4)	4.5(-5)	8.4(+6)
7.5(-4)	776	650	2.6(-4)	3.8(-5)	1.3(+7)
5.0(-4)	1078	933	2.6(-4)	3.3(-5)	1.4(+7)
2.5(-4)	1660	1511	2.3(-4)	2.6(-5)	1.4(+7)
1.0(-4)	3483	3324	1.6(-4)	1.5(-5)	1.4(+8)
7.5(-5)	4516	4335	9.5(-5)	1.1(-5)	2.2(+8)

TABLE 3.16: Adaptive algorithm interpolation results of $f(x, y) = \exp(-60((x - 0.35)^2 + (y - 0.25)^2)) + 0.2$ with different θ_{refine}.

Figure 3.22shows how the maximum and root mean square error decrease with the pre-set threshold, and the number of points required to achieve the given threshold. It also suggest a converngence rate $|e_{\mathbf{X}}(f)_{|\mathbf{X}|}| < C \times |\mathbf{X}|^{-1}$. In Figure 3.23, the author compares the adaptive error indicator (AEI) interpolation and residual sub-sampling (RE) method on approximating this target function.

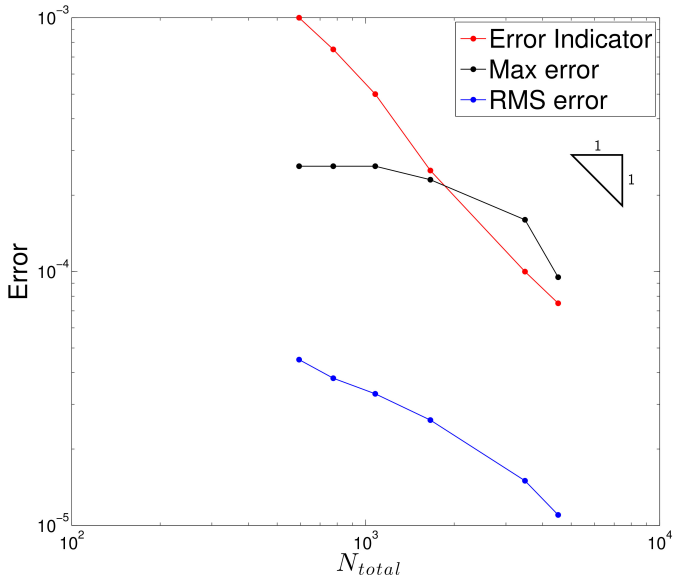

FIGURE 3.22: Convergence of adaptive error indicator interpolation for the $f(x, y) = \exp(-60((x - 0.35)^2 + (y - 0.25)^2)) + 0.2$ function.

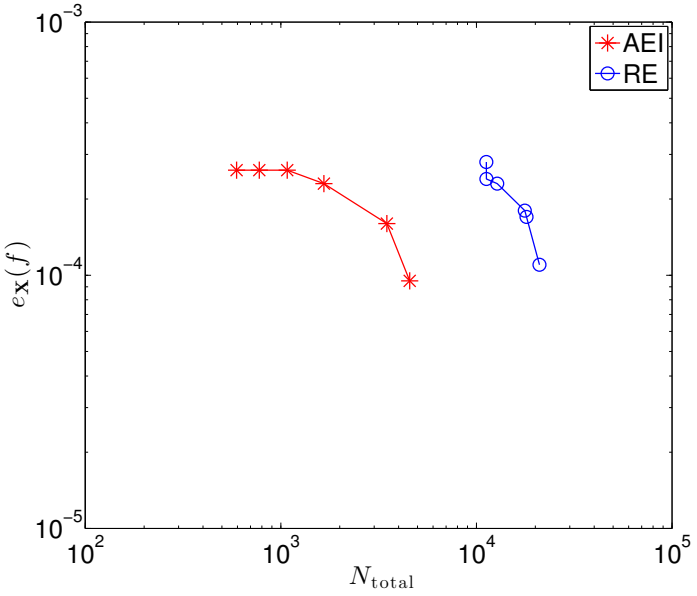

FIGURE 3.23: Comparison between AEI method and RE method for the $f(x, y) = \exp(-60((x - 0.35)^2 + (y - 0.25)^2)) + 0.2$ function.

3.5.2.4 The cone shape function

In one-dimensional cases, the author finds that the shifted absolute value function $f = |x - 0.04|$ is difficult to approximate due to its derivative singularity at $x =$

0.04. In this example the author explores the same singularity in two dimensions $f(x, y) = \sqrt{(x^2 + y^2)} + 0.2$ (Figure 3.24).

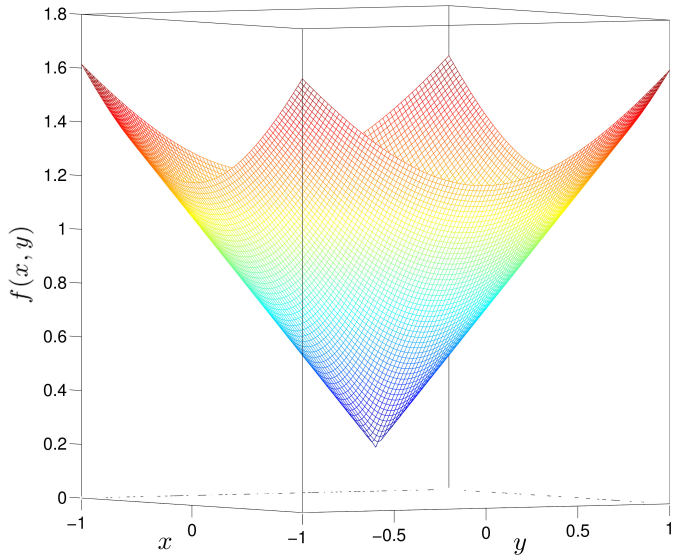

FIGURE 3.24: Interpolation of $f(x, y) = \sqrt{(x^2 + y^2)} + 0.2$, with $\theta_{\text{refine}} = 5.0(-4)$.

With $\theta_{\text{refine}} = 5.0(-4)$ the AEI algorithm took 11 iterations to reach the stopping criteria (see Table 3.17). A total $|\mathbf{X}| = 406$ centers were used to give an error $e_{\mathbf{X}}(f) = 3.1(-3)$ and $\text{RMS}(e_{\mathbf{X}}(f)) = 4.3(-5)$. All the condition number were below $4.4(+6)$.

Figure 3.25 shows the final distribution of centers that generated by AEI algorithm. Comparing 3.25 with Figure 3.21, one might think there are some redundant points that in Figure 3.25. These points that in the domain are necessary because this target function is more complicated than the two-dimensional exponential function. In order words, the curved plane is more complicated than the plain plane, so the more complicated target function would use more points to approximated with the same pre-setted threshold value.

Table 3.18 shows how the number of points need by the AEI algorithm varies with the choices of θ_{refine}. Figure 3.26 shows the process of the AEI algorithm staring with 100 centers with $\theta_{\text{refine}} = 2.5(-5)$. The black line is the maximum error, the

It	N_{total}	$\lvert\mathbf{X}\rvert$	N_{coarse}	N_{refine}	$e_{\mathbf{X}}(f)$	RMS($e_{\mathbf{X}}(f)$)	$\kappa(A)$
1	100	100	0	168	9.9(-2)	4.9(-3)	1.5(+6)
2	268	268	0	62	4.8(-2)	1.1(-3)	1.1(+6)
3	330	330	0	27	2.4(-2)	3.2(-4)	1.5(+6)
4	357	357	0	28	1.3(-2)	1.4(-4)	1.8(+6)
5	385	385	0	10	6.3(-3)	7.1(-5)	2.5(+6)
6	395	395	0	6	3.1(-3)	4.3(-5)	4.3(+6)
7	401	401	0	1	3.1(-3)	4.3(-5)	4.4(+6)
8	402	402	0	1	3.1(-3)	4.3(-5)	4.4(+6)
9	403	403	0	2	3.1(-3)	4.3(-5)	4.4(+6)
10	405	405	0	1	3.1(-3)	4.3(-5)	4.4(+6)
11	406	406	0	0	3.1(-3)	4.3(-5)	4.4(+6)

TABLE 3.17: Iterative process of AEI algorithm for $f(x,y) = \sqrt{(x^2+y^2)} + 0.2$ with $\theta_{\text{refine}} = 5.0(-4)$.

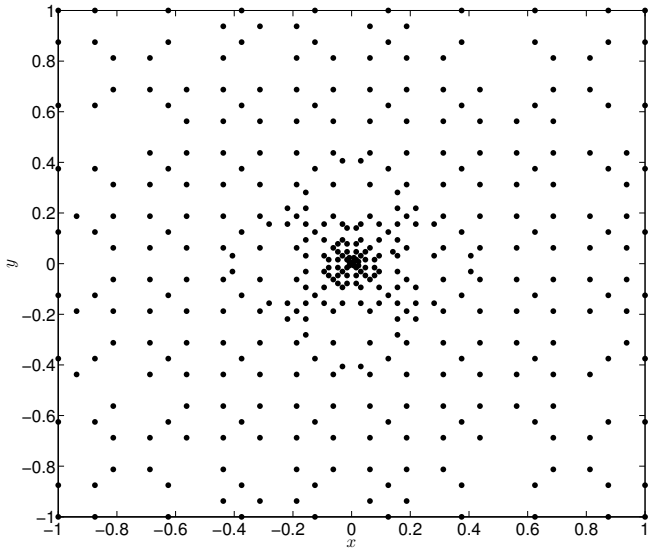

FIGURE 3.25: Final node distribution for approximation of $f(x,y) = \sqrt{(x^2+y^2)} + 0.2$, with $\theta_{\text{refine}} = 5.0(-4)$.

red line is the maximum error indicator value and the blue line is the root mean square error.

One sees that the in the latter parts of the approximation process the error decays much faster than in the beginning, demonstrating an acceleration of the accuracy as the singularity becomes better resolved. Compared to Figure 3.11, there is no bounce of the error and error indicator value, one could say in this case the error

θ_{refine}	N_{total}	$\lvert\mathbf{X}\rvert$	$e_{\mathbf{X}}(f)$	RMS$(e_{\mathbf{X}}(f))$	$\kappa(A)_{\max}$
1.0(-3)	269	269	6.3(-3)	9.6(-5)	1.9(+6)
7.5(-4)	350	350	6.3(-3)	9.0(-5)	2.2(+6)
5.0(-4)	406	406	3.1(-3)	4.3(-5)	4.4(+6)
2.5(-4)	712	712	1.6(-3)	2.1(-5)	1.0(+7)
1.0(-4)	1752	1752	7.8(-4)	9.6(-6)	4.4(+7)
7.5(-5)	2050	2050	7.8(-4)	1.2(-5)	5.6(+7)
5.0(-5)	3267	3267	3.9(-4)	5.4(-6)	1.5(+8)
2.5(-5)	6346	6346	1.9(-4)	2.6(-6)	4.2(+8)

TABLE 3.18: Adaptive algorithm interpolation results of $f(x,y) = \sqrt{(x^2+y^2)}+0.2$ with different θ_{refine}.

indicator could track the error well. Also, one could see that the AEI method is capable to cope with a singularity in the slope of target function.

In Figure 3.27, one sees how the maximum and root mean square error decrease with the pre-set threshold and the number of points required to achieve the given threshold. One sees, as in this two dimensional examples, the AEI method suggests a convergence rate $|e_{\mathbf{X}}(f)_{|\mathbf{X}|}| < C \times |\mathbf{X}|^{-1}$.. Figure 3.28 shows the comparison of AEI and RE method on this cone shape function.

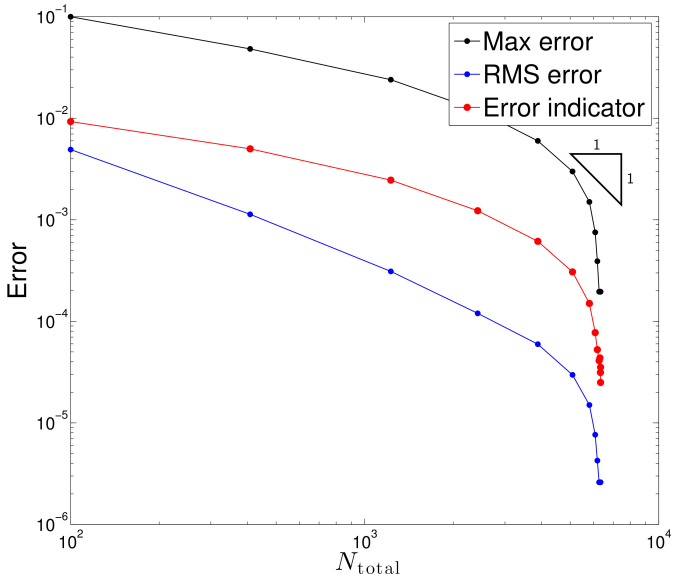

FIGURE 3.26: Error convergence of the AEI algorithm for $f(x,y) = \sqrt{(x^2+y^2)}+0.2$ with $\theta_{\text{refine}} = 2.5(-5)$.

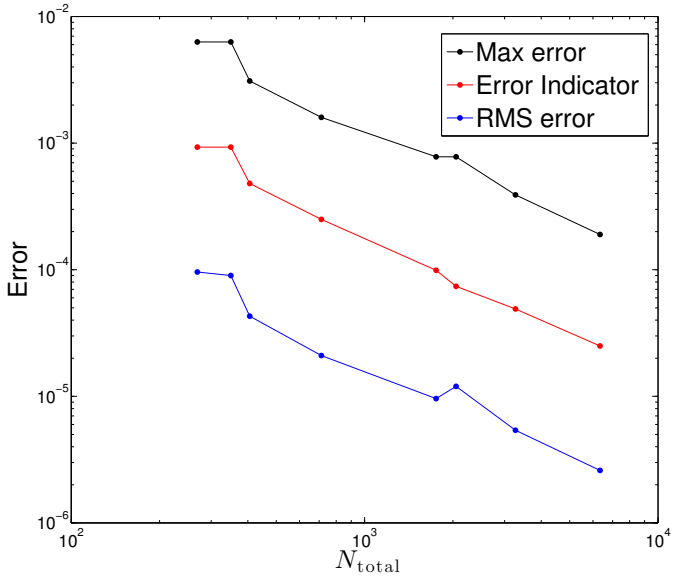

FIGURE 3.27: Error versus number of points for approximation of $f(x,y) = \sqrt{x^2+y^2}+0.2$.

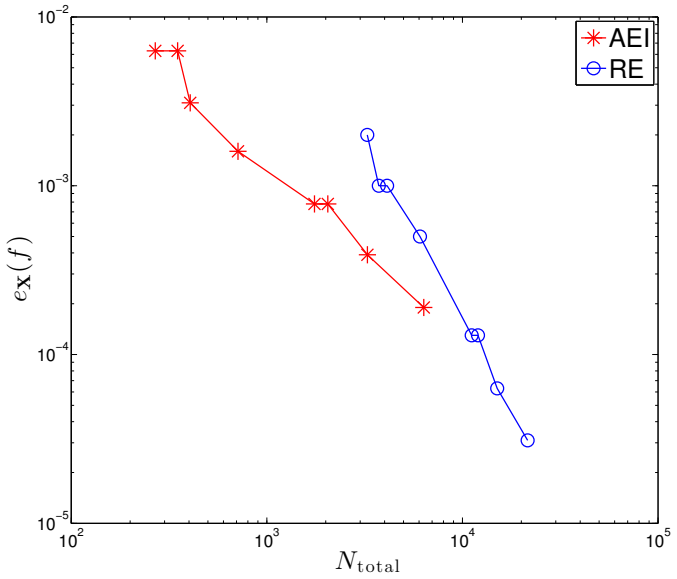

FIGURE 3.28: Comparison between AEI method and RE method for $f(x,y) = \sqrt{x^2+y^2}+0.2$.

3.5.2.5 The Lena picture

In the previous examples, the author has approximated functions with rapid variation or derivative singularities. These are conventional examples in which one has seen that the adaptive error indicator RBF approximation method delivers good

accuracy with the aim of minimising the number of function evaluation used. In this case, the Lena picture (see Figure 3.29 left panel) is used as the target. Here the function is a 128×128 pixel image, so is discrete, i.e. it has discontinuities everywhere. The local RBF approximation is used to compute an approximation to the image between the centers of the pixels (the author terms this an emulator S_P^{multi}); (Figure 3.29 right panel). This emulator is the target function.

Table 3.19 shows three approximation results and Figure 3.30, 3.31,3.32 show the corresponding reconstructed images. In this case, one could see that adaptive error indicator method could deal with this extremely complicated target function.

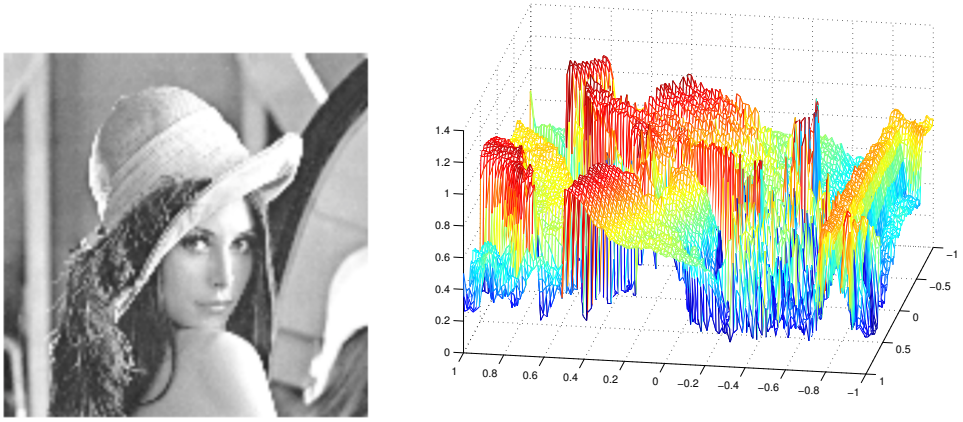

FIGURE 3.29: The Lena picture from [14], $128 \times 128 = 16384$ pixels

.

θ_{refine}	N_{total}	$\lvert\mathbf{X}\rvert$	$e_{\mathbf{X}}(f)$	$\text{RMS}(e_{\mathbf{X}}(f))$	$\kappa(A)_{\max}$
2.0(-2)	3604	3603	6.4(-1)	7.5(-2)	6.6(+6)
1.5(-2)	6703	6702	5.3(-1)	5.3(-2)	1.6(+7)
1.2(-2)	9711	9710	3.1(-1)	3.9(-2)	2.6(+7)

TABLE 3.19: Error in adaptive interpolation of S_P^{multi} with different θ_{refine}.

3.5.3 Three dimensional function approximation

One can extend the two dimensional node adaptation scheme to three dimensions. It begins with the uniformly distributed centers $\mathbf{X}_0$ in $[-1, 1]^3$. The corresponding indication set Ξ_0 for the centers set. It begins with $|\mathbf{X}_0| = 208$ and its indication set Ξ_0 and the neighbourhood set N_ξ parameter M is set to 60.

FIGURE 3.30: The approximation $S_{\mathbf{X}}^{\text{multi}}$ regenerated Lena picture with $N_{\text{total}} = 3604$ and $|\mathbf{X}| = 3603$.

FIGURE 3.31: The approximation $S_{\mathbf{X}}^{\text{multi}}$ regenerated Lena picture with $N_{\text{total}} = 6703$ and $|\mathbf{X}| = 6702$.

FIGURE 3.32: The approximation $S_{\mathbf{X}}^{\text{multi}}$ regenerated Lena picture with $N_{\text{total}} = 9711$ and $|\mathbf{X}| = 9710$.

A test set T containing 25000 Halton nodes is used to test the infinity and root means square errors. The shape parameter c of each center is set to be a constant divided by its distance to the nearest neighbour as in Equation 3.16, $C = 1$, and $\theta_{\text{coarse}} = \theta_{\text{refine}}/1000$. The author compare these results to those in [6].

3.5.3.1 The 3D exponential function

This example is

$$f(x,y,z) = \exp\{-81/16[(x-0.5)^2 + (y-0.5).^2 + (z-0.5)^2]\}/3. \tag{3.20}$$

Table 3.20 shows the results of the adaptive interpolation. Figure 3.33 shows how the maximum error changes with θ_{refine}. The maximum error (black) tracks the pre-set threshold value (red) well. The condition numbers observed in the algorithm remain relatively small.

θ_{refine}	N_{total}	$\lvert\mathbf{X}\rvert$	$e_{\mathbf{X}}(f)$	RMS($e_{\mathbf{X}}(f)$)	$\kappa(A)_{\max}$
5.0(-3)	333	333	3.6(-3)	6.1(-4)	7.4(+1)
1.0(-3)	665	665	1.3(-3)	1.4(-4)	1.5(+4)
5.0(-4)	1419	1419	5.6(-4)	5.8(-5)	5.6(+4)
1.0(-4)	6643	6643	1.7(-4)	1.4(-5)	4.1(+6)

TABLE 3.20: $\exp\{-81/16[(x-0.5)^2 + (y-0.5).^2 + (z-0.5)^2]\}/3$ interpolation results by error indicator.

N_{uniform}	$e_{\mathbf{X}}(f)$
343	2.0(-2)
729	2.8(-3)
1728	1.4(-3)
6859	6.7(-4)

TABLE 3.21: $\exp\{-81/16[(x-0.5)^2 + (y-0.5).^2 + (z-0.5)^2]\}/3$ interpolation results by uniform centers.

Comparing the results in Table 3.20 to those in [6] (summarised in Table 3.22), the AEI method provides much better accuracy with the same number of centers. In Table 3.21 we see the results achieved with uniform centers. Comparing Table 3.20 and Table 3.21, the error indicator adaptive algorithm could still put more centers at the area which it's difficult to approximate, it still could provide much better approximation quality with less centers.

In Figure 3.33 one sees the rate of decay of the error with the number of points. This also suggest a convergence rate $|e_{\mathbf{X}}(f)_{|\mathbf{X}|}| < C \times |\mathbf{X}|^{-1}$.

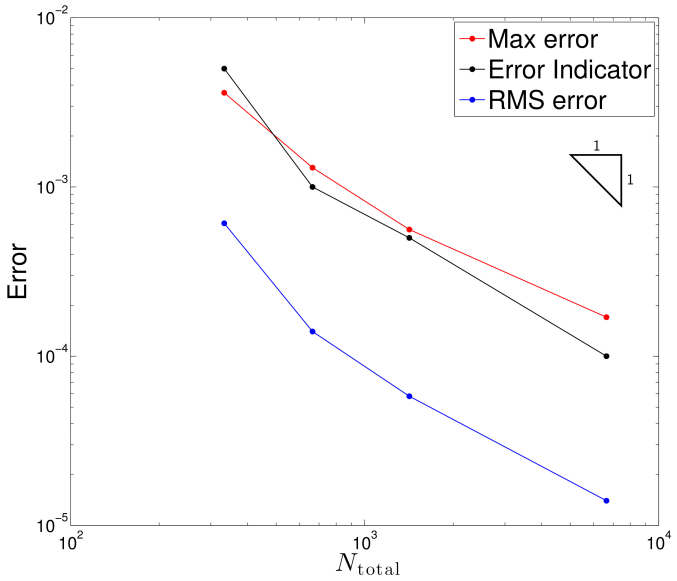

FIGURE 3.33: Error convergence of interpolation results for $f(x,y,z) = \exp\{-81/16[(x-0.5)^2+(y-0.5).^2+(z-0.5)^2]\}/3$.

N_{uniform}	$e_{\mathbf{X}}(f)$	RMS$(e_{\mathbf{X}}(f))$
343	1.0(-2)	8.3(-4)
729	6.7(-3)	4.0(-4)
1728	4.3(-3)	2.0(-4)

TABLE 3.22: $\exp\{-81/16[(x-0.5)^2+(y-0.5).^2+(z-0.5)^2]\}/3$ interpolation results by uniform centers in [6].

3.5.3.2 The European call option

The Black Scholes equation is used to describe the change in price of an option over time. The equation is:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0.$$

The value of a European call option C for a non-dividend-paying underlying stock in terms of the Black Scholes parameters is:

$$C = N(d_1)S - N(d_2)K\exp(-r(T-t)),$$

$$d_1 = \frac{1}{\sigma\sqrt{T-t}}\left[\ln\left(\frac{S}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)(T-t)\right],$$

$$d_1 = \frac{1}{\sigma\sqrt{T-t}}\left[\ln\left(\frac{S}{K}\right) + \left(r - \frac{\sigma^2}{2}\right)(T-t)\right],$$

that:

- $N(.)$ is s the cumulative distribution function of the standard normal distribution.
- $T - t$ is the time to maturity.
- S is the spot price of the underlying asset.
- K is the strike price.
- r he risk free rate (annual rate, expressed in terms of continuous compounding).
- σ is is the volatility of returns of the underlying asset.

The European call option price C could be considered as function of 6 variables $C(S, r, \sigma, K, T, t)$, so the price of a one-year European call option priced at $t = 0$ for a fixed strike price $K = K_1$ is $C(S, r, \sigma, K_1, T = 1, t = 0)$. Table 3.23 shows the interpolation to this function $C(S, r, \sigma, K_1 = 100, T = 1, t = 0)$ in $r = [0.01, 0.05], \sigma = [0.1, 0.3], S = [90, 110]$.

Figure 3.34 shows how the error behaves with respect to the number of points. One sees the error indicator (red line) tracks the maximum error (black line) well. The algorithm appears to give a convergence rate $|e_{\mathbf{X}}(f)_{|\mathbf{X}|}| < C \times |\mathbf{X}|^{-1}$ as in the previous example. The condition number remains of moderate size throughout the algorithm, and the adaptive algorithm is many times better than the results of using uniform data. Specially, Table 3.23 shows an adaptive interpolation using 1910 nodes which the L-infinity error is $7.7(-3)$ while Table 3.24 show an uniform interpolation using 19683 nodes which the L-infinity is $1.1(-2)$.

3.5.3.3 The Perdichizzi turbine case

In the final example, the Perdichizzi turbine case which was described in Chapter 1 is discussed. In short, one needs to estimate the loss coefficient $L(\alpha_1, Po_1, \nu)$ which

θ_{refine}	N_{total}	$\lvert\mathbf{X}\rvert$	$e_{\mathbf{X}}(f)$	RMS$(e_{\mathbf{X}}(f))$	$\kappa(A)_{\max}$
5.0(-3)	1910	1910	7.7(-3)	1.3(-3)	1.5(+6)
2.5(-3)	3595	3595	3.4(-3)	6.2(-4)	4.1(+6)
1.0(-3)	8914	8914	1.2(-3)	2.3(-4)	1.5(+7)
7.5(-4)	11985	11985	1.1(-3)	1.6(-4)	2.3(+7)
5.0(-4)	18283	18283	6.0(-4)	1.4(-5)	3.6(+7)

TABLE 3.23: $C(S, r, \sigma, K_1 = 100, T = 1, t = 0)$ interpolation results.

N_{uniform}	$e_{\mathbf{X}}(f)$
2197	4.9(-2)
4096	3.7(-2)
9261	2.2(-2)
12167	1.8(-2)
19683	1.1(-2)

TABLE 3.24: $C(S, r, \sigma, K_1 = 100, T = 1, t = 0)$ interpolation results by uniform centers.

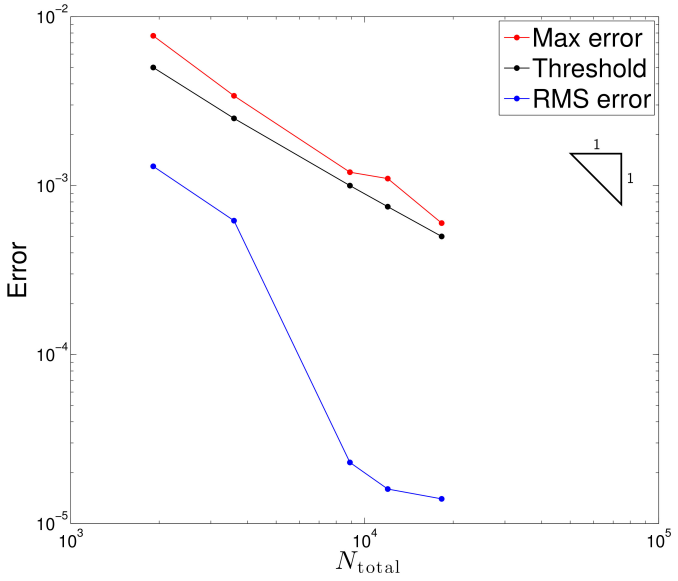

FIGURE 3.34: Error versus points for adaptive interpolation of $C(S, r, \sigma)$.

could used to improve the efficiency of this particular turbine. $L(\alpha_1, Po_1, \nu)$ could be evaluated at arbitrary set of points by experiment and/or CFD. Both methods are expensive and time-consuming. So, the adaptive error indicator (AEI) is used instead to build a surrogate model for the function $L(\alpha_1, Po_1, \nu)$.

However, under the the condition of limited resources, only one uniform scattered

data set is available, which is N_{uniform} of size 21^3 in the $\alpha_1 = [-60, 60]$ degrees, $Po_1 = [1.04 \times 10^5, 2.6 \times 10^5]$ Pa, $\nu = [6.92 \times 10^{-6}, 2.67 \times 10^{-5}]$ $m^2 s^{-1}$. As this interested domain is small in ν when compared to Po_1, then the author scales the uniform set N_{uniform} into the cube $[-1, 1]^3$ to have the scaled uniform set N_{scaled}. The scaling is simply for the easy implementation reason. For the AEI method, the author used this scaled uniform set N_{scaled} to compute an RBF approximation to the function value between the centers, this is an emulator $S^{\text{multi}}_{\text{turbine}}$. This emulator is the target function.

θ_{refine}	N_{total}	$\lvert\mathbf{X}\rvert$	$e_{\mathbf{X}}(f)$	RMS($e_{\mathbf{X}}(f)$)	$\kappa(A)_{\max}$
5.0(-3)	328	328	1.6(-2)	2.6(-3)	6.6(+4)
2.5(-3)	378	378	1.3(-2)	2.4(-3)	8.3(+4)
1.0(-3)	1140	1140	6.7(-3)	1.4(-4)	7.5(+5)
7.5(-4)	1438	1483	6.8(-3)	1.1(-4)	7.5(+5)
5.0(-4)	2397	2397	5.0(-3)	7.1(-4)	1.6(+6)

TABLE 3.25: $S^{\text{multi}}_{\text{turbine}}$ adaptive error indicator interpolation results.

N_{uniform}	$e_{\mathbf{X}}(f)$
343	2.4(-2)
512	2.1(-2)
1331	1.3(-2)
1728	1.1(-2)
2744	6.4(-2)

TABLE 3.26: $S^{\text{multi}}_{\text{turbine}}$ interpolation results by uniform centers.

Table 3.25 shows the adaptive error indicator (AEI) approximation results to this emulator. Table 3.26 shows the results achieved with uniform centers. Figure 3.35 compares the two method on this case. It is shown that adaptive error indicator method delivers much better accuracy while using the same number of samples in this industrial case.

If engineers and developers apply this adaptive error indicator (AEI) method on this Perdichizzi turbine case, the saving of time and expenses which can be achieved by running less experiments and simulations will definitely help to accelerate the design process of this turbine.

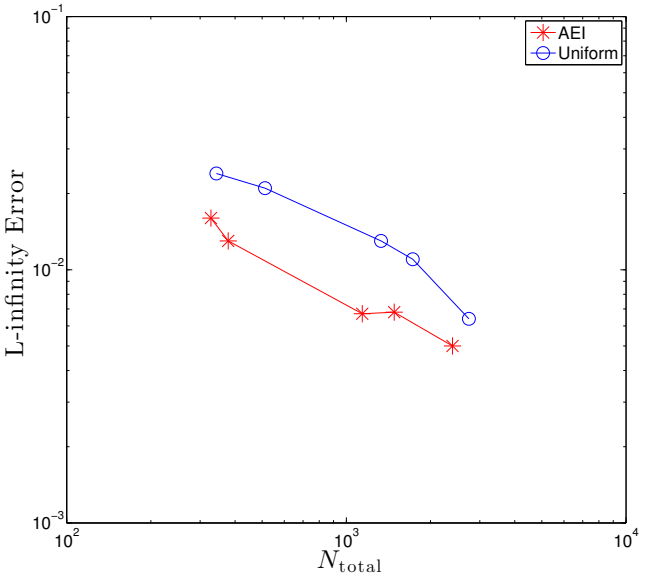

FIGURE 3.35: Comparison between AEI method and uniform interpolation for this turbine case.

For the engineering interest, the author would like to show the error between the approximation $S_{\mathbf{X}}^{\text{multi}}$ and the emulator $S_{\text{turbine}}^{\text{multi}}$ while α_1 is equal to a constant. A test set T of 101×101 uniformly spaced nodes in (Po_1, ν) is used to test the approximation quality.

The error function $E(\alpha_1 = c, Po_1, \nu)$ is defined :

$$E(\alpha_1 = c, Po_1, \nu) = S_{\text{turbine}}^{\text{multi}}(\alpha_1 = c, Po_1, \nu) - S_{\mathbf{X}}^{\text{multi}}(\alpha_1 = c, Po_1, \nu), \quad \text{where} \quad |\mathbf{X}| = 2397.$$

Figure 3.36,3.37,3.38,3.39 and 3.40 show the error functions.

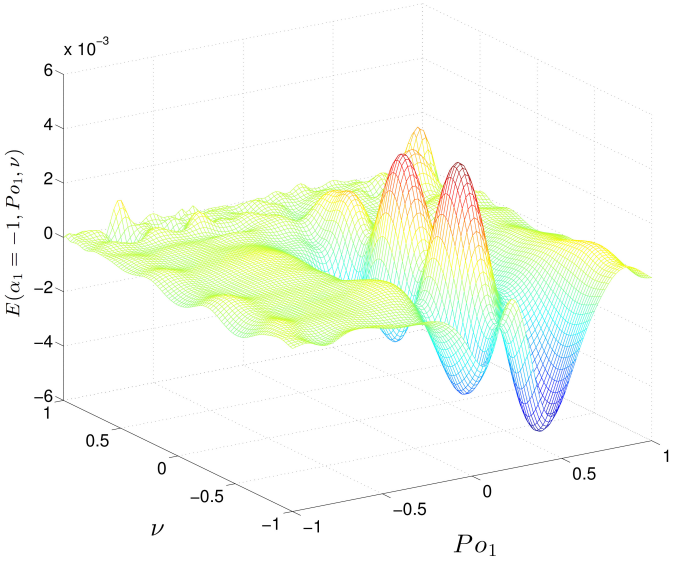

FIGURE 3.36: Error function $E(\alpha_1 = -1, Po_1, \nu)$.

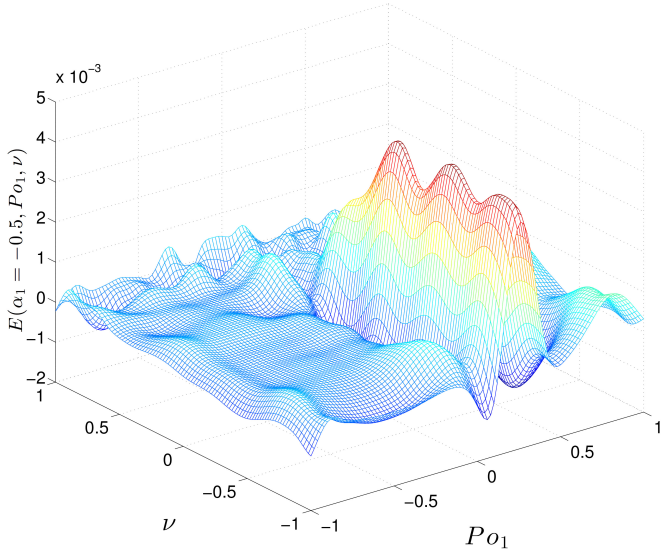

FIGURE 3.37: Error function $E(\alpha_1 = -0.5, Po_1, \nu)$.

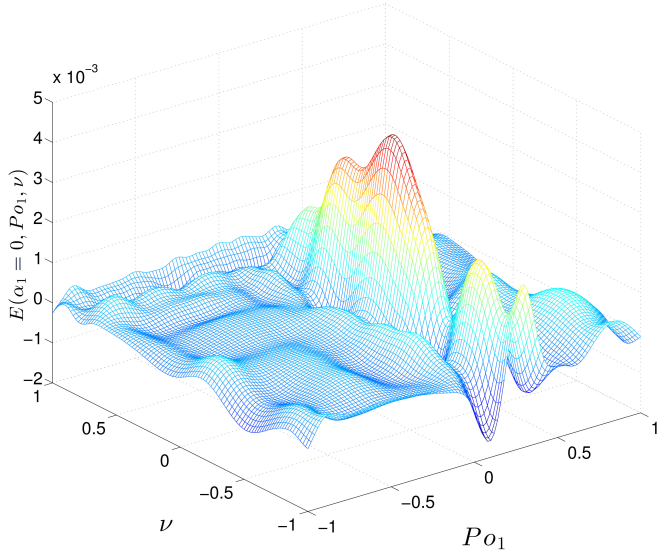

FIGURE 3.38: Error function $E(\alpha_1 = 0, Po_1, \nu)$.

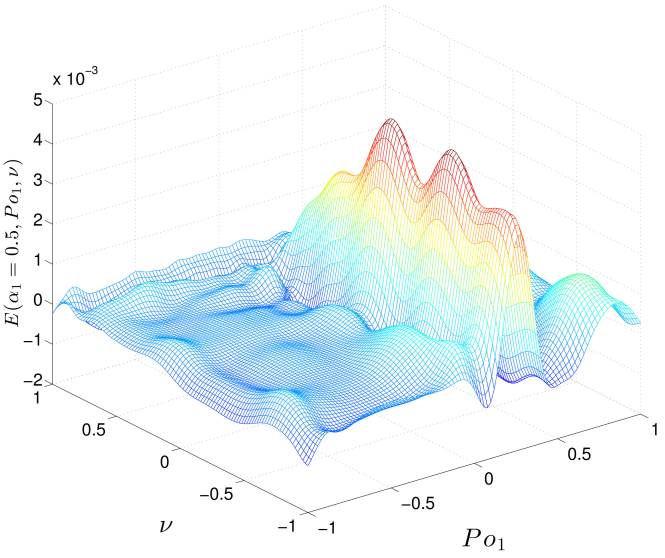

FIGURE 3.39: Error function $E(\alpha_1 = 0.5, Po_1, \nu)$.

3.6 Robustness of the Adaptive Error Indicator Method

This section explores the sensitivity of the AEI algorithm to the parameters that need to be chosen in the algorithm. The user should be reassured that the Parameter Set 1 is proper for most cases.

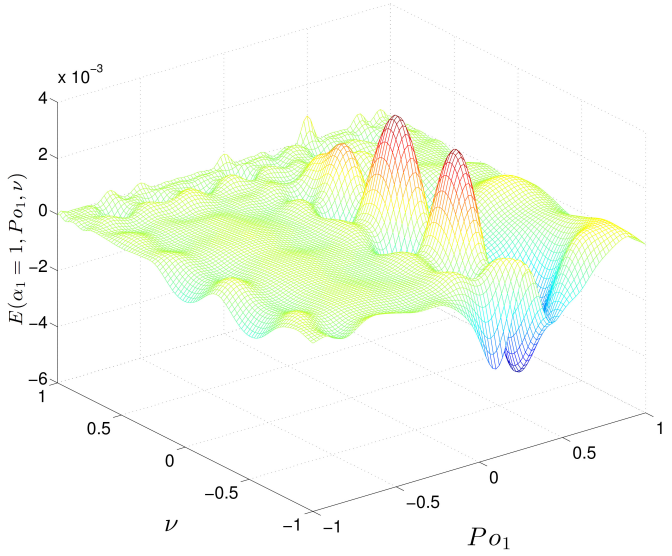

FIGURE 3.40: Error function $E(\alpha_1 = 1, Po_1, \nu)$.

Section 3.1 to 3.3 describe the structure of the adaptive error indicator (AEI), the components of which are:

1. The global interpolant $S^{\text{multi}}_{\mathbf{X}_k}$.
2. The local interpolant $S^{\text{ps}}_{N_\xi}$.
3. The Algorithm 1 which generates the sample set $\mathbf{X}_k$ and corresponding indication set Ξ_k.

In Section 3.5, the AEI method has been applied to different target functions and the approximation results shows that, with the parameter choices made, the algorithm delivers good approximations. The parameters chosen for the mentioned experiments are referred as Parameter Set 1. The values of Parameter Set 1 is summarised below:

1. The shape parameter at ith point, c_i, in the global interpolant $S^{\text{multi}}_{\mathbf{X}_k}$, it is set by Equation 3.16 $c_i = C_1/d_i$, where d_i is the distance to the nearest neighbour. Then $C_1 = 0.75$ for one dimensional cases, $C_1 = 0.5$ for two dimensional cases, $C_1 = 1$ for three dimensional cases.

2. The number of points M in set N_ξ for the local interpolant $S^{\text{ps}}_{N_\xi}$. In one dimensional cases, $M = 4$. In two dimensional cases, $M = 24$. In three dimensional cases, $M = 60$. There is no shape parameter for $S^{\text{ps}}_{N_\xi}$.

Parameter Set 1 is not the optimal choice for parameters, since for different functions f the optimal parameters choices might change. The author will show the robustness of adaptive error indicator method by varying the parameters from Parameter Set 1, and observing that the results do not change too much.

Table 3.27 shows the Runge function $f(x) = (1 + 25x^2)^{-1}$ approximation results with different parameter settings with $\theta_{\text{refine}} = 2.0(-5)$. Table 3.28 shows the approximation results with different parameter settings with $\theta_{\text{refine}} = 2.0(-8)$. The approximation degrades as a result of increasing the shape parameter, at the same time as the condition number decreases. This is what standard theory suggests [79]. One also sees that a change in the indication set size beyond 5 makes little difference. The number of points required is a reflection of the approximation accuracy and this is governed by the choice of the shape parameter. The balance here is between the magnitude of condition number that one is prepared to accept and the accuracy. The adaptive shape parameter which described by Equation 3.16 in the AEI is crucial to keep the condition number small. Slight changes the constant C in Equation 3.16 will not cause the condition number to grow rapidly. The above numerical experiments could show that the AEI method is a stable method.

Parameter settings	N_{total}	$e_{\mathbf{X}}(f)$	RMS$(e_{\mathbf{X}}(f))$	$\kappa(A)_{\max}$
$C_1 = 0.5, M = 3$	140	1.6(-5)	3.4(-6)	2.0(+7)
$C_1 = 0.5, M = 5$	88	1.9(-5)	3.7(-6)	7.4(+6)
$C_1 = 1, M = 6$	79	2.1(-5)	3.3(-6)	4.3(+5)
$C_1 = 1, M = 8$	74	4.0(-5)	5.8(-6)	8.7(+5)

TABLE 3.27: Adaptive error indicator interpolation results of Runge function with different parameter settings, with $\theta_{\text{refine}} = 2.0(-5)$.

Table 3.29 shows the modified Franke function (two dimensional) approximation results with different parameter settings with $\theta_{\text{refine}} = 1.0(-3)$. Table 3.30 shows the approximation results with different parameter settings with $\theta_{\text{refine}} = 5.0(-4)$.

Parameter settings	N_{total}	$e_{\mathbf{X}}(f)$	RMS($e_{\mathbf{X}}(f)$)	$\kappa(A)_{\max}$
$C_1 = 0.5, M = 3$	1346	2.3(-8)	8.3(-10)	1.8(+11)
$C_1 = 0.5, M = 5$	814	1.6(-8)	9.1(-10)	1.5(+11)
$C_1 = 1, M = 6$	1162	2.5(-8)	5.9(-9)	3.8(+10)
$C_1 = 1, M = 8$	1211	1.9(-8)	5.3(-9)	4.0(+10)

TABLE 3.28: Adaptive error indicator interpolation results of Runge function with different parameter settings, with $\theta_{\text{refine}} = 2.0(-8)$.

Parameter settings	N_{total}	$e_{\mathbf{X}}(f)$	RMS($e_{\mathbf{X}}(f)$)	$\kappa(A)_{\max}$
$C_1 = 2, M = 20$	1128	1.7(-3)	2.7(-4)	3.0(+5)
$C_1 = 2, M = 30$	851	2.2(-3)	3.4(-4)	2.1(+5)
$C_1 = 5, M = 35$	1060	2.4(-3)	4.0(-4)	1.3(+5)
$C_1 = 5, M = 40$	1019	2.4(-3)	4.4(-4)	1.3(+5)

TABLE 3.29: Adaptive error indicator interpolation results of modified Franke function with different parameter settings, with $\theta_{\text{refine}} = 1.0(-3)$.

Parameter settings	N_{total}	$e_{\mathbf{X}}(f)$	RMS($e_{\mathbf{X}}(f)$)	$\kappa(A)_{\max}$
$C_1 = 2, M = 20$	2180	8.3(-4)	1.1(-4)	8.1(+5)
$C_1 = 2, M = 30$	1550	8.9(-4)	1.4(-4)	5.6(+5)
$C_1 = 5, M = 35$	1713	7.3(-4)	1.8(-4)	3.1(+5)
$C_1 = 5, M = 40$	1617	8.1(-4)	1.9(-4)	2.9(+5)

TABLE 3.30: Adaptive error indicator interpolation results of modified Franke function with different parameter settings, with $\theta_{\text{refine}} = 5.0(-4)$.

In these tables there is a strong correlation between the number of points and the error once there are enough points in the indicator set. An increase in the shape parameter leads to an increase in the error, and to a decrease in the condition number, though less extreme than in one dimension.

These results are similar to the results generated by the adaptive error indicator method with Parameter Set 1 and the effect of changes in parameters decrease with increasing dimension.

3.7 Summary

The adaptive error indicator (AEI) RBF interpolation method is motivated by the problem that sampling is expensive. A new error indicator construed by a global interploant and a local interpolant is developed and it is the crucial part of AEI method. The adaptive shape parameter is also applied in this method in order to keep the condition number of interpolation matrix in a moderate scale. By numerical examples up to three dimensional problems, the adaptive error indicator (AEI) RBF interpolation method has verified itself that it delivers accurate approximations with as few as possible evaluations from the target functions.

Chapter 4

Multilevel Adaptive Error Indicator (MAEI) Approximation

4.1 Multilevel RBF interpolation scheme

Multilevel approximation schemes, also known as multistep or multiscaled approximation schemes, are appropriate tools [26, 52, 50, 70, 72] for multivariate scatted data approximation problem, where the sample number N is extremely large and the point in sample set $\mathbf{X}$ are unevenly distributed.

In [50], the multilevel level scheme starts with a decomposition of the data $\mathbf{X}$ into a hierarchy

$$\mathbf{X}_1 \subset \mathbf{X}_2 \subset \cdots \subset \mathbf{X}_{L-1} \subset \mathbf{X}_L = \mathbf{X} \tag{4.1}$$

of nested subsets. The data hierarchy (4.1) is a priori computed in [26] by using a thinning algorithm, a recursive point removal scheme. In (4.1), Delaunay triangulation is used in a thinning algorithm in each coarsening stage and a criterion for removal based on the separation distance and the fill distance which is discussed in more detail in [27]. In this subsequent synthesis of data, a sequence $S_1, ..., S_L$ of approximations to f is then recursively computed by the following multilevel interpolation scheme.

Let $S_0 \equiv 0$. For $j = 1, ..., L$, compute an interpolant $\Delta S_j : \mathbb{R}^d \to \mathbb{R}$ to the residual $f - S_{j-1}$ on $\mathbf{X}_j$. Then let $S_j = S_{j-1} + \Delta S_j$. Altogether, the following L interpolation problems are to be solved one after the other:

$$\begin{aligned} f|_{\mathbf{X}_1} &= \Delta S_1|_{\mathbf{X}_1}; & S_1 &= \Delta S_1; \\ (f - S_1)|_{\mathbf{X}_2} &= \Delta S_2|_{\mathbf{X}_2}; & S_2 &= S_1 + \Delta S_2; \\ &\vdots & &\vdots \\ (f - S_{L-1})|_{\mathbf{X}_L} &= \Delta S_L|_{\mathbf{X}_L}; & S_L &= S_{L-1} + \Delta S_L. \end{aligned} \tag{4.2}$$

That every S_j in Equation 4.2 interpolates f on $\mathbf{X}_j$, that is :

$$S_j|_{\mathbf{X}_j} = f|_{\mathbf{X}_j} \quad \text{for} \quad 1 \leq j \leq L. \tag{4.3}$$

Then the interpolant S_L is the approximation for the target function. In [26, 50], radial basis functions were applied to solve the interpolation problems in Equation 4.2.

In [52], Iske and Levesley proposed another multilevel approximation scheme with a different hierarchy of data sets. This scheme construct a data hierarchy in following form:

$$C_1 \subset C_2 \subset \cdots \subset C_{L-1} \subset C_L, \tag{4.4}$$

where each C_j, $1 \leq j \leq L$, represents a collection of clusters of $\mathbf{X}$. In contrast to [26, 50], the representing set C_j does not have to be subset of $\mathbf{X}$.

This scheme begins with a decomposition of the domain $\Omega \subset \mathbb{R}^d$, the bounding box of $\mathbf{X}$, into a collection of different size cells in each level, each cell contains a cluster of points of $\mathbf{X}$. It is denoted as $\{\omega_n, 1 \leq n \leq N\}_j, 1 \leq j \leq L$, N is number of total points in $\mathbf{X}$. Then ω_n^j denotes the nth cell in level j. For each set $\{\omega_n, 1 \leq n \leq N\}_j$, it generated $x_{\omega_n}^j$, the cell center for each cell ω_n^j. Each cell contains $X_{\omega_n}^j$ points. To each $x_{\omega_n}^j$ a cell average value $S_\omega(x_{\omega_n}^j)$ of f or $f - S_j$ is assigned, where $S_\omega(x_{\omega_n}^j)$ is the polyharmonic spline RBF interpolant which satisfies $S_\omega|_{X_{\omega_n}^j} = f|_{X_{\omega_n}^j}$ or $S_\omega|_{X_{\omega_n}^j} = f - S_j|_{X_{\omega_n}^j}$.

In [26, 50, 52], these multilevel approximation schemes target multivariate scattered data approximation with a certain sample set $\mathbf{X}$. Here, we present a new multilevel approximation scheme with RBF which targets the kind of problem that is discussed in previous chapter, which the sample set $\mathbf{X}$ needs to be decided by the user and sampling from target function f is expensive and time-consuming.

4.2 The error indicator

This scheme generates a collection of sample sets $\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_k$ where the method generates $\mathbf{X}_{k+1}$ from $\{\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_k\}$ via a refinement strategy which will be described below. In contrast with e.g. Iske and Levesley [50, 52], the collection of sample sets is not necessarily a nested sequence. The strategy for sampling more points depends on an error indication, which follows the idea in Chapter 3. In a region where is challenging for the approximation, two different approximation methods will give significantly different results, when compared with regions where the approximation is more straight-forward.

The first approximation method is the current approximation S_L as defined by Equation 4.2. The second approximation method will be a RBF interpolant based on values of the target function on all available sample points. Then a function $\eta(\xi)$ with domain in the current indication set, assigns a positive value ϵ to each indication point. This value indicates the local approximation quality at each indication nodes and serves to determine where the approximate solution S_L requires more accuracy at these specified indication nodes, and requires no extra function evaluation. Below, the author gives the definition of the error indicator which is proposed in this chapter, which is modified version from Chapter 3.

The notation needs to be clarified: $\mathbf{X}_i^j$ denotes the jth set of centers at level i.

Definition 4.1 (Error Indicator 2)**.** For $L \geq 1$, that having the centers set at different level

$$\{\mathbf{X}_1^1, \mathbf{X}_2^1, \cdots, \mathbf{X}_2^{n_2}, \cdots, \mathbf{X}_L^1, \cdots, \mathbf{X}_L^{n_L}\},$$

let the indication set Ξ_L, which corresponds to $\mathbf{M}_L$, be a set of scattered points, at which one wants to know the approximation quality. Then $\eta : \Xi_L \to [0, \infty)$.

The error indicator function $\eta(\xi)$ is in the following form:

$$\eta(\xi) = |S_L(\xi) - S_{\mathbf{M}_L}(\xi)|, \quad \xi \in \Xi_L. \tag{4.5}$$

Here, $|.|$ denotes the absolute value. The function $S_L(\xi)$ is the radial basis function approximation of the target function at ξ as defined in Equation (4.2), which is constructed by $\{\Delta S_{\mathbf{X}_1^1}, \Delta S_{\mathbf{X}_2^1}, \cdots, \Delta S_{\mathbf{X}_2^{n_2}}, \cdots, \Delta S_{\mathbf{X}_L^1}, \cdots, \Delta S_{\mathbf{X}_L^{n_L}}\}$. The function $S_{\mathbf{M}_L}(\xi)$ is the radial basis function reconstruction which matches the target function value at ξ by a scattered point set $\mathbf{M}_L$. $\mathbf{M}_L$ consists of the all the available sample points at current level, that is

$$\mathbf{M}_L = \bigcup_{1 \le i \le L, 1 \le j \le n_i} \mathbf{X}_i^j. \tag{4.6}$$

By definition, the elements in $\mathbf{M}_L$ are unique, and

$$S_{\mathbf{M}_L}(v) = f(v) \quad \textit{for} \quad v \in \mathbf{M}_L. \tag{4.7}$$

The indication set Ξ_L is determined by $\mathbf{M}_L$, the details of the relationship between Ξ_L and $\mathbf{M}_L$ are explained in the following algorithm.

One sees that the error indicator defined above measures the deviation between a multilevel approximation and a global approximation at the point ξ. The intuition inside this method is as simple as previous one, when ξ lies in a smooth region of the function, the two different approximations should give similar results and the error indicator $\eta(\xi)$ is expected to be small, whereas in the region of less regularity for f, or around discontinuities, the error indicator $\eta(\xi)$ is expected to be large.

So, the significance $\eta(\xi), \xi \in \Xi$, are used to flag points $\xi \in \Xi$ as "to be refined" or not according to the definition below.

Definition 4.2. Let θ_{refine} be a tolerance values satisfying $\theta_{\text{refine}} > 0$. One could say a point $\xi \in \Xi$ is classified into refine set N_{refine} when $\eta(\xi) > \theta_{\text{refine}}$.

These parameter θ_{refine} should be specified according to the users' need.

4.3 Adaptive point sets

In mind of the above definitions, the generation of the centers sets $\mathbf{X}_i^j$ for interpolants at different level will be described. That is, tree structure is applied for the centers sets. In Figure 4.1, it shows the sketch of this tree structure of data sets. In $L = 1$ level, it starts with one centers set $\mathbf{X}_1^1$. The following level centers set are generated by the previous level centers sets, that is $\{\mathbf{X}_k^1, \cdots, \mathbf{X}_k^{n_k}\}$ are generated by $\{\mathbf{X}_1^1, \mathbf{X}_2^1, \cdots, \mathbf{X}_2^{n_2}, \cdots, \mathbf{X}_{k-1}^1, \cdots, \mathbf{X}_{k-1}^{n_{k-1}}\}$.

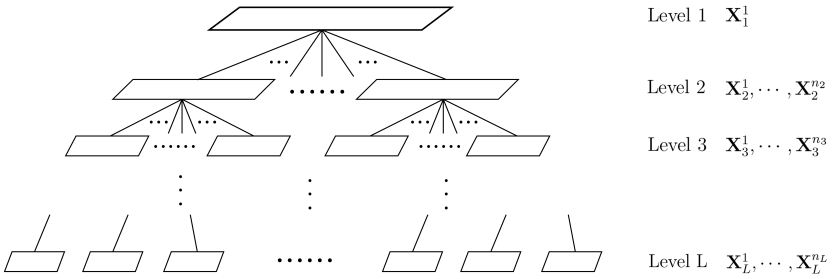

FIGURE 4.1: Sketch of the tree structure of data set $\mathbf{X}_i^j$.

The number of sets in each level $n_k, 2 \leq k \leq L$, is decided by the error indicator, we illustrate the relationship between one certain centers set at level k and its next generation centers sets in level $k + 1$.

In one dimensional cases, the initial sample set $\mathbf{X}_1^1$ is a set of uniformly distributed nodes in the domain Ω, so $\mathbf{M}_1 = \mathbf{X}_1^1$. By above definition, $\mathbf{M}_1 = \{m_1, m_2, \cdots, m_{T_1}\}$, and the indication set Ξ_1 is the middle points of $\mathbf{M}_1$, that is $\Xi_k = \{0.5(m_i + m_{i+1}), i = 1, 2, \cdots, T_1 - 1)\}$. Having $\mathbf{M}_1$ and Ξ_1, we could follow the algorithm below to generate next level centers set $\mathbf{X}_{k+1}^j, 1 \leq j \leq n_{k+1}$.

1. $\mathbf{M}_k$ is specified.

2. $\Xi_k = \{0.5(m_i + m_{i+1}), i = 1, 2, \cdots, T_k - 1\}$.

3. Evaluate the error indicator at current indication set, by having $\eta(\Xi_k)$, N_{refine} is generated,

$$N_{\text{refine}} = \{\xi : \eta(\xi) > \theta_{\text{refine}}\}.$$

4. Classify the points N_{refine} into subsets $\{N_{\text{refine}_1}, N_{\text{refine}_2}, \cdots, N_{\text{refine}_{n_{k+1}}}\}$ according to connectivity, that is points in each subset N_{refine_i} are connected according their index.

5. Determine the interval for each subset N_{refine_i}, that is having the interval $[a_i, b_i]$ for N_{refine_i}. That is

$$a_i = \sup\{m \in \mathbf{M}_k, m < \alpha_i\}, \quad \text{where} \quad \alpha_i = \inf N_{\text{refine}_i},$$

and

$$b_i = \inf\{m \in \mathbf{M}_k, m > \beta_i\}, \quad \text{where} \quad \beta_i = \sup N_{\text{refine}_i}.$$

6. Find the points that

$$I_i = \{m \in \mathbf{M}_k, a_i \leq m \leq b_i\}$$

for $1 \leq i \leq n_{k+1}$.

7. Then $\mathbf{X}_{k+1}^i = N_{\text{refine}_i} \cup I_i$ for $1 \leq i \leq n_{k+1}$. Then $\mathbf{M}_{k+1}$ is updated as in Equation 4.6. That is

$$\mathbf{M}_{k+1} = \cup_{i=1}^{n_{k+1}} \mathbf{X}_{k+1}^i \cup \mathbf{M}_k.$$

Figure 4.2 shows one example of the one dimensional structure. In the first panel of Figure 4.2, one has current all available center point $\mathbf{M}_k$ which are the black dots and the corresponding indication set Ξ_k which are the red crosses. The second panel shows after having the value of error indicator $\eta(\Xi_k)$ and the indication points that have large error (the remaining red crosses) are left. In third panel, we have the set $\mathbf{X}_{k+1}^1$ (in red oval) and $\mathbf{X}_{k+1}^2$ (in blue oval) according to above algorithm.

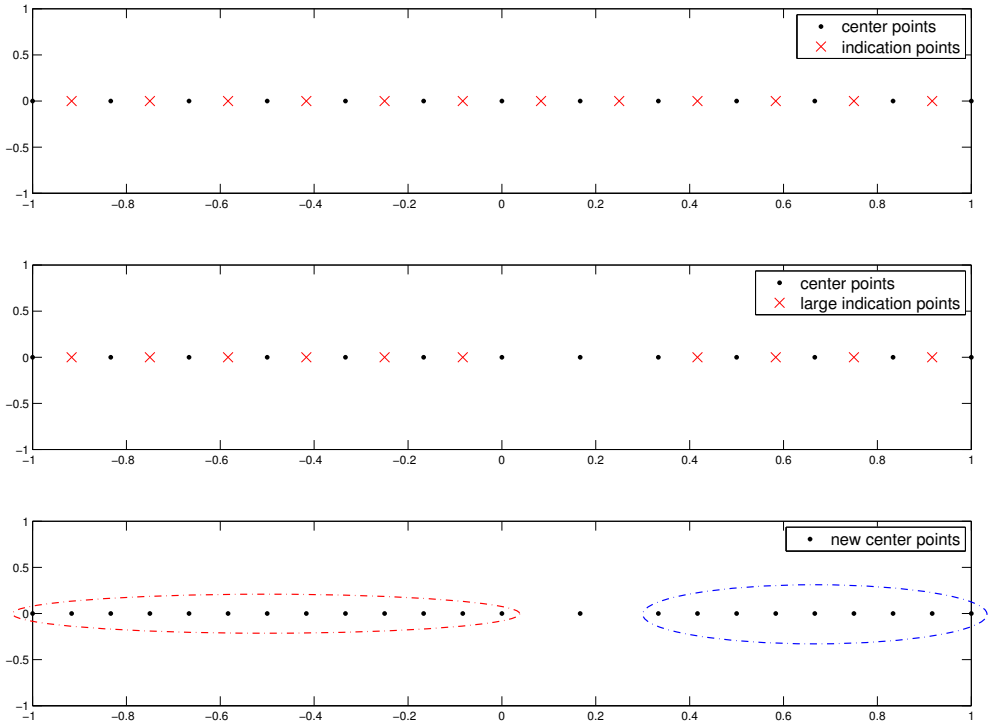

FIGURE 4.2: One example of the data structure in one dimensional case.

In two dimensional cases, there is no natural ordering, so the author applies a modified quadtree data structure to construct sample sets $\mathbf{X}_i^j$ for $1 \leq i \leq L, 1 \leq j \leq n_i$. Quadtree was originally proposed in [25], the author followed the idea and modified it to fit the usage. In the domain $\Omega \in \mathbb{R}^2$, it initially fills the domain Ω with uniformly distributed nodes $\mathbf{X}_1^1 = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ that $N = n \times n, n \geq 3$, that could be considered the domain Ω is divided by $\mathbf{X}_1^1$ into $n-1$ sub domains, which are squares, $\{\square_1, \square_2, \cdots, \square_{n-1}\}$, that in each $\square_i$ consist 4 vertices and 5 indication points, that is

$$\square_i = \{\mathbf{x}_1^i, \cdots, \mathbf{x}_4^i, \xi_1^i, \cdots, \xi_5^i\}. \tag{4.8}$$

One example of this data structure in $[-1, 1]^2$ is shown, see Figure 4.3, the domain is fill by 4 squares $\square$, each square $\square$ has 4 centers (black nodes) and 5 corresponding indication points (red nodes).

When $k = 1$, $\mathbf{M}_1 = \mathbf{X}_1^1$. For each $\mathbf{M}_k$, the data structure has total T_k squares, then the algorithm generates the middle points of each square edge and the center point of each square $\square_j, 1 \leq j \leq T_k$ as the indication set Ξ_k as showed in Figure 4.3, the black nodes set is $\mathbf{X}_1^1$ and the red nodes set is Ξ_1.

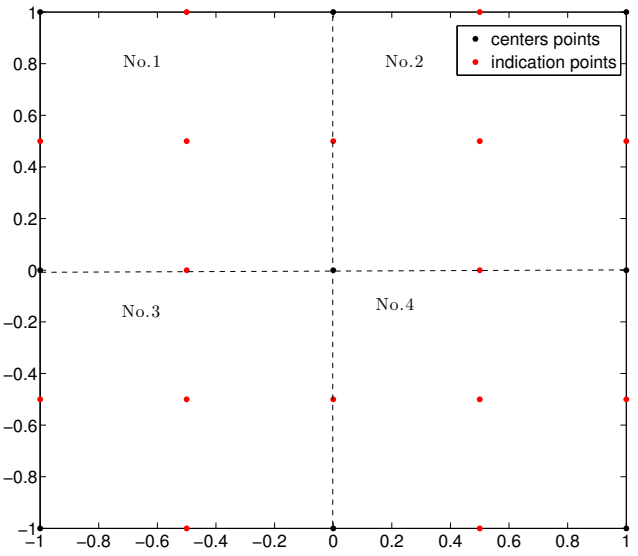

FIGURE 4.3: $n = 3$, exampled 2D initial sample set $\mathbf{X}_1^1$, which $\mathbf{M}_1 = \mathbf{X}_1^1$, and indication set Ξ_1.

Having above conceptions, we apply below algorithm to generate $\mathbf{M}_{k+1}$ and $\mathbf{X}_{k+1}^j, 1 \leq j \leq n_{k+1}$.

1. $\mathbf{M}_k$ is specified.

2. The middle points of each square edge and the center point of each square $\square_j, 1 \leq j \leq T_k$, are the points in indication set Ξ_k.

3. Evaluate the error indicator at current indication set, by having $\eta(\Xi_k)$, N_{refine} is generated, $N_{\text{refine}} = \{\xi : \eta(\xi) > \theta_{\text{refine}}\}$.

4. For each point in N_{refine}, find its corresponding squares $\square_{\text{refine}}$ that the point belongs to, that is having the candidate squares $\square_{\text{refine}}$ which has at least one indication point in N_{refine}. In order words,

$$\square_{\text{refine}} = \{\square : \square \cap N_{\text{refine}} \neq \emptyset\}.$$

5. Group the squares in $\square_{\text{refine}}$ by connectivity, that is the squares in each group $\square^j_{\text{connected}}$ for $1 \leq j \leq n_{k+1}$ are connected one after another. In other words, $\square^j_{\text{connected}}$ stands for the jth set of connected squares. For each $\square_{\text{connected}}$, it is

$$\square_{\text{connected}} = \{\square : \forall \square \in \square_{\text{refine}} \quad \textit{and} \quad \exists \square_i \in \square_{\text{refine}}, \square \cap \square_i \neq \emptyset\}.$$

   Also, for $\square_i, \square_j, i \neq j, \square_i \in \square_{\text{connected}}, \square_j \in \square_{\text{connected}}$, there exists a path goes from $\square_i$ to $\square_j$ that only passing through the elements inside $\square_{\text{connected}}$.

6. All the available nodes, $\mathbf{M}_k \cup \Xi_k$, that inside the connected squares $\square^j_{\text{connected}}$ make up the sets $\mathbf{X}^j_{k+1}$ for $1 \leq j \leq n_{k+1}$. That is

$$\mathbf{X}^j_{k+1} = \square^j_{\text{connected}}, \quad \textit{for} \quad j = 1, 2, \cdots, n_{k+1}.$$

7. Then $\mathbf{M}_{k+1}$ is updated as in Equation 4.6, that is

$$\mathbf{M}_{k+1} = \cup_{j=1}^{n_{k+1}} \mathbf{X}^j_{k+1} \cup \mathbf{M}_k.$$

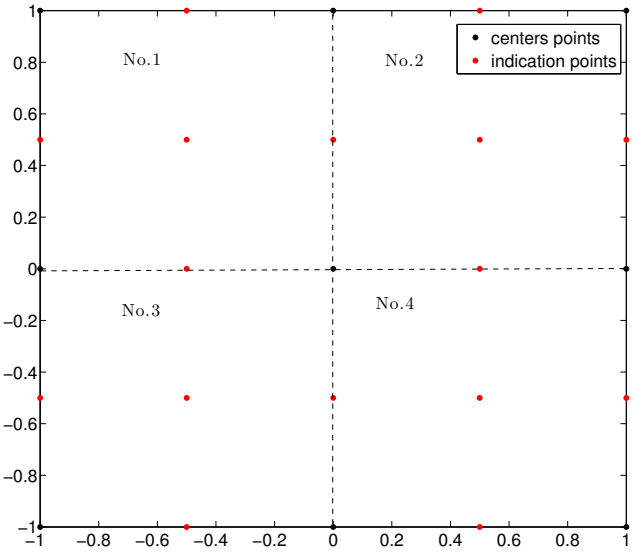

FIGURE 4.4: One example of two dimensional data structure, Panel 1.

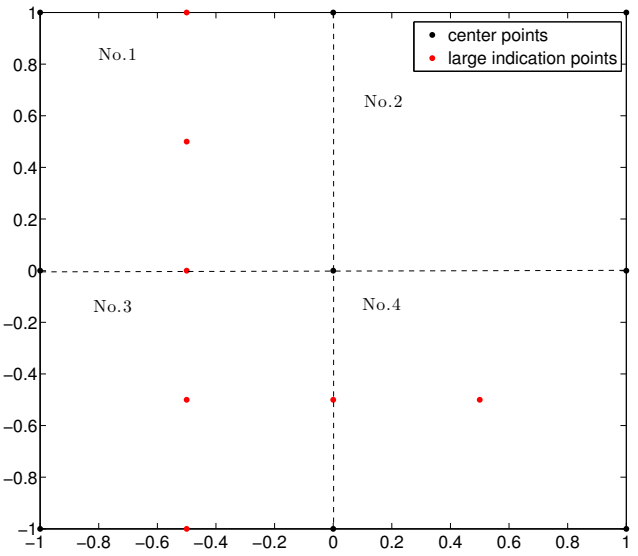

FIGURE 4.5: One example of two dimensional data structure, Panel 2.

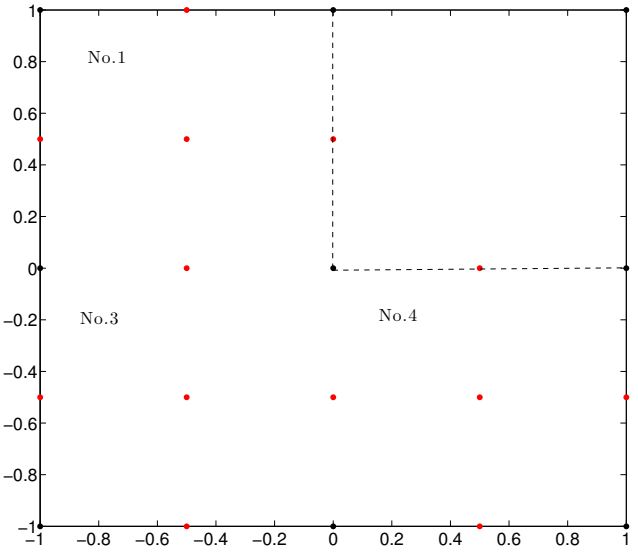

FIGURE 4.6: One example of two dimensional data structure, Panel 3.

Figure 4.4, 4.5, 4.6, 4.7 show one example of two dimensional case. In Figure 4.4, the current all available centers $\mathbf{M}_k$ (the black dots) and its 4 square $\square$ are given, also the corresponding indication set Ξ_k (the red dots) is given.

Figure 4.5 shows the large error points (the red dots) that picked by the error indicator $\eta(\Xi_k)$ and its involved squares $\square_1, \square_3, \square_4$. In Figure 4.6, the connected

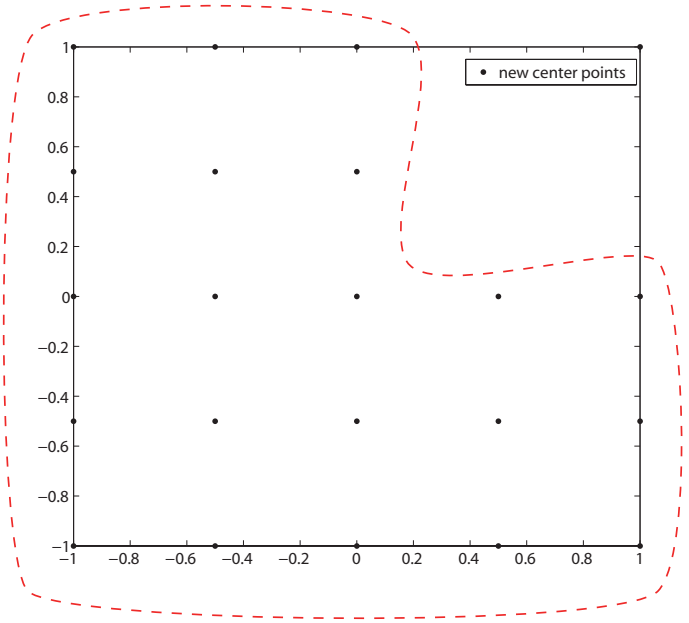

FIGURE 4.7: One example of two dimensional data structure, Panel 4.

squares $\square_1, \square_3, \square_4$ make up the connected set $\square^1_{\text{connected}}$. In Figure 4.7, the points that inside the red line loop constructs the set $\mathbf{X}^1_{k+1}$. Then all the black points in Figure 4.7 make up the set $\mathbf{M}_{k+1}$.

4.4 Multilevel algorithm achieved by using an error indicator

Having above sample set determination algorithm and the error indicator function, one could summarise this multilevel adaptive error indicator (MAEI) approximation scheme for the target function f.

Let $S_0 \equiv 0$. For $\mathbf{X}^j_i \subset \mathbb{R}^d, 1 \leq i \leq L, 1 \leq j \leq n_i$, one could define a sub-domain

$$\Omega_{\mathbf{X}^j_i} = [\min(X^j_i[1]), \max(X^j_i[1])] \times \cdots \times [\min(X^j_i[d]), \max(X^j_i[d])],$$

where $\mathbf{X}_i^j[k]$ denotes the kth dimensional value of $\mathbf{X}_i^j$. Also the interpolant $\Delta S_{\mathbf{X}_i^j}$ is defined as below :

$$\Delta S_{\mathbf{X}_i^j}(\mathbf{x}) = \begin{cases} \Delta S_{\mathbf{X}_i^j}(\mathbf{x}), & \mathbf{x} \in \Omega_{\mathbf{X}_i^j}, \\ 0, & \mathbf{x} \notin \Omega_{\mathbf{X}_i^j}. \end{cases} \tag{4.9}$$

For $1 \leq i \leq L, 1 \leq j \leq n_i$, compute an interpolant $\Delta S_{\mathbf{X}_i^j} : \mathbb{R}^d \to \mathbb{R}$ to the residual $f - S_{i-1}$ on $\mathbf{X}_i^j$, which the sample set $\mathbf{X}_i^j$ is generated by above mentioned algorithms. Then let

$$S_i = S_{i-1} + \sum_{j=1}^{n_i} \Delta S_{\mathbf{X}_i^j}. \tag{4.10}$$

Altogether, the following interpolation problems are to be solved one after the other:

$$\begin{aligned} f|_{\mathbf{X}_1^1} &= \Delta S_{\mathbf{X}_1^1}|_{\mathbf{X}_1^1}; & S_1 &= \Delta S_{\mathbf{X}_1^1}; \\ (f - S_1)|_{\mathbf{X}_2^1} &= \Delta S_{\mathbf{X}_2^1}|_{\mathbf{X}_2^1}; & & \\ &\vdots & & \\ (f - S_1)|_{\mathbf{X}_2^{n_2}} &= \Delta S_{\mathbf{X}_2^{n_2}}|_{\mathbf{X}_2^{n_2}}; & S_2 &= S_1 + \sum_{j=1}^{n_2} \Delta S_{\mathbf{X}_1^j}; \\ &\vdots & &\vdots \\ (f - S_{L-1})|_{\mathbf{X}_L^1} &= \Delta S_{\mathbf{X}_L^1}|_{\mathbf{X}_L^1}; & & \\ &\vdots & & \\ (f - S_{L-1})|_{\mathbf{X}_L^{n_L}} &= \Delta S_{\mathbf{X}_L^{n_L}}|_{\mathbf{X}_L^{n_L}}; & S_L &= S_{L-1} + \sum_{j=1}^{n_L} \Delta S_{\mathbf{X}_L^j}; \end{aligned} \tag{4.11}$$

Note that every interpolant S_i in Equation 4.11 matches f on sample set $\mathbf{X}_i = \bigcup_{j=1}^{n_i} \mathbf{X}_i^j$, i.e.,

$$S_i|_{\mathbf{X}_i} = f|_{\mathbf{X}_i}, \quad \textit{for} \quad 1 \leq i \leq L. \tag{4.12}$$

In 4.11, we use multiquadric RBF, cubic RBF, polyharmonic splines RBF and Gaussian RBF to construct interpolants.

4.5 Numerical Results

The proposed multilevel approximation scheme has been implemented up to two dimensional space. This method shows the effectiveness of this error indicator in locating regions of large interpolation errors and in the multilevel RBF interpolation method to improve the accuracy of the approximation. One dimensional and two dimensional test functions have been tested by this method and compare to results in the papers [17]. In these example cases, the multiquadric radial basis function $\phi(r) = \sqrt{(1 + c^2 r^2)}$ for the first level interpolant S_1. The multiquadric radial basis function has a free parameter c and the choice of this shape parameter will be specified in below cases.

4.5.1 One Dimensional Function Multilevel Approximation

For one dimensional test functions, the author set the initial center set $\mathbf{X}_1^1$ to be the uniformly distributed points in the interval $[-1, 1]$ and the indication set Ξ_k and following centers set $\mathbf{X}$ is generated by above algorithm. A test set T containing 5001 equally spaced nodes is used to test the approximation quality: $e_{S_L}(f) = \max_{t \in T} |f(t) - s_L(t)|$ and the root mean square value in $f(t) - S_L(t), t \in T$, that is $\mathrm{RMS}(e_{S_L}(f))$.

4.5.1.1 The Runge Function

The author first redo a standard approximation problem in Section 3.5.1.1, the Runge function $f(x) = (1 + 25x^2)^{-1}$ on $[-1, 1]$. The shape parameter c in the multiquadric radial basis function for s_1 is set to be

$$0.75 \times 2/(|\mathbf{X}_1| - 1)$$

as described in Chapter 3. In the following examples, the interpolant S_1 follows this setting. In the error indicator function, the interpolant that used all the available sample points is $S^{\mathrm{ps}}_{\mathbf{M}_L}$, the polyharmonic splines RBF is applied. As it is

defined, that is

$$S^{\text{ps}}_{M_L}(x) = \sum_{i=1}^{N_{M_L}} \alpha_i (||x - x_i||_2)^5 + \beta_1 + \beta_2 x + \beta_3 x^2. \tag{4.13}$$

For the interpolants $\Delta S^{\text{gau}}_{\mathbf{X}_i^j}, 2 \leq i \leq L, 1 \leq j \leq n_i$, the Gaussian RBF is used. The Gaussian RBF also has a shape parameter c, the author sets c_i in $\Delta S^{\text{gau}}_{\mathbf{X}_i^j}$ in below form:

$$c_v = \frac{C}{h_v}, \tag{4.14}$$

where h_v is the nearest distance between x_v and other centers in set $\mathbf{X}_i^j = \{x_w, wd = 1, 2, \cdots, V_i^j\}$, in this case we set $C = 20$.

Figure 4.8 shows the final approximation of the Runge function by this multilevel approximation scheme, with $|\mathbf{X}_1^1| = 13$, refinement threshold $\theta_{\text{refine}} = 2.0(-5)$. Again, one observes that sample points cluster near the boundaries where approximation is more challenging due to the one-sided nature of the information, and at the origin, where the target function changes more rapidly. Note that the final maximum error is $1.8(-5)$ which is below θ_{refine} suggesting that the error indicator is working well. The largest condition number of interpolation matrix in this case is $3.3(+3)$.

Table 4.1 shows present the progress of the multilevel approximation process.In Table 4.1, L stands for the level number, m_L stands for the accumulated sample set number and $|\mathbf{M}_L|$ is the accumulated total sample point number. It has a total of 4 levels and 7 sample sets $\mathbf{X}$. The final approximation S_L has total 43 sample points, that is the whole process computed a total of 43 evaluations of the target function. $\kappa(A)_{\max}$ stands for the largest condition number at each level of approximation.

Figure 4.9 shows the total centers set that used in S_L in different level, the y-axis shows the level number L. In this case, the error indicator and the above algorithm together generated these 7 centers set in 4 levels.

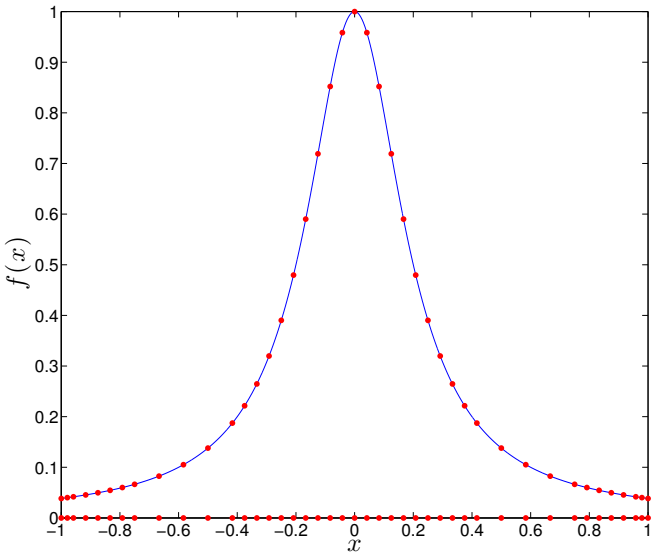

FIGURE 4.8: Runge function multilevel adaptive error indicator approximation with final sample points distribution, initial $|\mathbf{X}_1^1| = 13$ with $\theta_{\text{refine}} = 2.0(-5)$.

L	m_L	$\lvert\mathbf{M}_L\rvert$	$e_{S_L}(f)$	RMS$(e_{S_L}(f))$	$\kappa(A)_{max}$
1	1	13	1.3(-2)	5.1(-3)	3.3(+3)
2	2	25	3.0(-4)	9.5(-5)	0.9(+1)
3	5	41	3.3(-5)	6.9(-6)	1.6(+2)
4	7	43	1.8(-5)	5.3(-6)	4.9(+1)

TABLE 4.1: Multilevel approximation process for Runge function with $\theta_{\text{refine}} = 2.0(-5)$.

Figure 4.10 shows how the error decreases with the number of points in the set $\mathbf{M}_L$, staring at 13, for $\theta_{\text{refine}} = 2.0(-8)$, and finish with 143 points, the total number of points that sampled from target function. The final approximation S_L used 143 centers, with $e_{S_L}(f) = 1.9(-8)$ and RMS$(e_{S_L}(f)) = 4.8(-9)$. The red nodes in Figure 4.10 are the maximum values of the error indicator function at each level in absolute value, so one can see that the error indicator is a good measure of approximation error because the measured error (red line) tracks the approximation error (black line). One sees a convergence rate $|e_{\mathbf{M}_L}(f)_{|\mathbf{M}_L|}| < C \times |\mathbf{M}_L|^{-3.5}$.

In Chapter 3, the author used the adaptive error indicator (AEI) interpolation

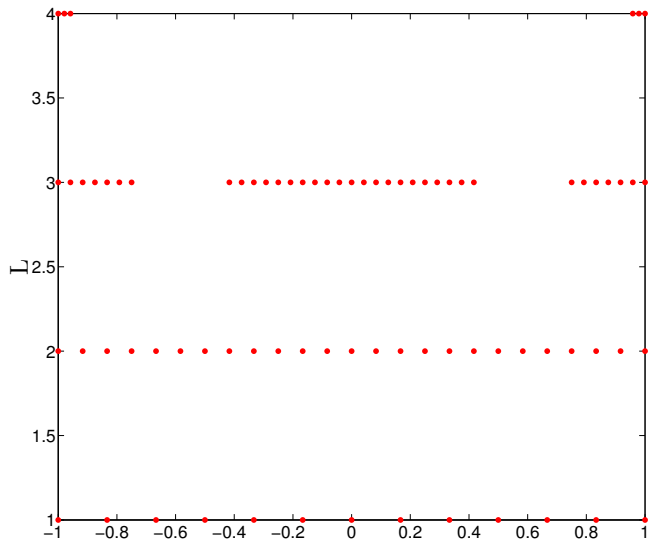

FIGURE 4.9: The centers sets used in Runge function multilevel error indicator approximation, with $\theta_{\text{refine}} = 2.0(-5)$.

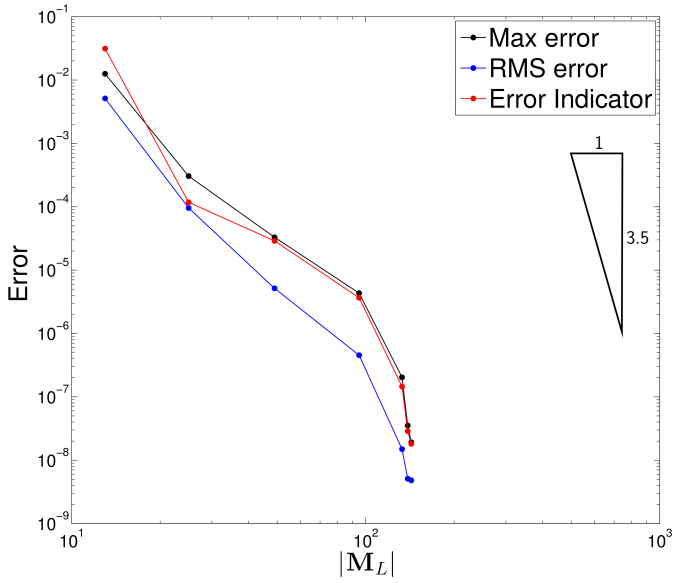

FIGURE 4.10: Runge function approximation error for each level L, $|\mathbf{M}_L|$ is the total number of samples of the target function. $\theta_{\text{refine}} = 2.0(-8)$.

method on the same example. The number of centers that used in the final interpolant and the total numbers of function samples computed from target function were recorded. In [17], Driscoll and Heryudono use the residual sub-sampling method also on this example, they record the number of centers used in the final interpolant, but the total numbers of function samples computed from the target

function is not reported.

Table 4.2 compares the results and the resources needed, $e_{\mathbf{X}}(f)$ stands for the maximum error of the approximation, the N_{total} stands the total evaluation numbers from the target function and N_{total} for sub-sampling method is implemented by the author of this thesis. In Table 4.2, RS stands for residual sub-sampling method, AEI stands for adaptive error indicator approximation and MAEI stands for above described multilevel error indicator approximation scheme. In this case for Runge function, we see the multilevel error indicator approximation outperforms the adaptive error indicator method and the sub-sampling method by a great amount. The MAEI method provides the same level of accuracy with less total evaluations from the target function than the AEI method and the RS method.

Method	$e_{\mathbf{X}}(f)$	N_{total}
RS	1.3(-5)	285
AEI	1.4(-5)	85
MAEI	1.8(-5)	43
RS	1.7(-7)	2491
AEI	1.8(-8)	710
MAEI	1.9(-8)	143

TABLE 4.2: Comparison of methods for Runge function

4.5.1.2 The hyperbolic tan function

This example considers $f(x) = \tanh(60x - 0.1)$. In the error indicator, the interpolant that used all the available sample points is $S^{\text{cub}}_{\mathbf{M}_L}$, the cubic RBF is applied. For the interpolants $\Delta S^{\text{multi}}_{\mathbf{X}_i^j}, 2 \leq i \leq L, 1 \leq j \leq n_i$, the multiquadric RBF is used. For the multiquadric shape parameter c_v for $\Delta S^{\text{multi}}_{\mathbf{X}_i^j}, 2 \leq i \leq L, 1 \leq j \leq n_i$, is setted as described in Equation 4.14. In this case, $C = 1$.

Table 4.3 shows the process of multilevel adaptive error indicator (MAEI) approximation with threshold $\theta_{\text{refine}} = 1.0(-5)$. In this case, the multilevel approximation converges at 8 levels with 12 sample sets which have total 139 evaluations from the target function. The final approximation S_L has error $e_{S_L}(f) = 9.2(-6)$ and

L	m_L	$\lvert\mathbf{M}_L\rvert$	$e_{S_{m_L}}(f)$	RMS$(e_{S_{m_L}}(f))$	$\kappa(A)_{max}$	$e_{S_{m_L}}(f)\kappa(A)_{max}$
1	1	13	7.2(-1)	1.7(-1)	3.3(+3)	2.4(+3)
2	2	25	5.2(-1)	9.6(-2)	8.6(+4)	4.5(+4)
3	3	49	2.6(-1)	3.7(-2)	3.3(+5)	8.6(+4)
4	6	81	4.9(-2)	5.6(-3)	5.2(+5)	2.5(+4)
5	7	109	1.4(-3)	1.3(-4)	4.5(+5)	6.3(+2)
6	8	127	9.2(-6)	1.8(-6)	1.9(+5)	1.7
7	11	138	9.2(-6)	1.8(-6)	2.2(+4)	2.0(-1)
8	12	139	9.2(-6)	1.8(-6)	1.4(+2)	1.3(-3)

TABLE 4.3: Approximation process of MAEI scheme of $\tanh(60x - 0.1)$, with $\theta_{\text{refine}} = 1.0(-5)$.

RMS$(e_{S_L}(f)) = 1.8(-6)$. Figure 4.11 shows how the error indicator distributes centers around the steepest part of f.

One observes that the even the max condition number $\kappa(A)_{max}$ grows in the beginning, the $e_{S_{m_L}}(f)\kappa(A)_{max}$ is not growing as fast as the max condition number. The $e_{S_{m_L}}(f)\kappa(A)_{max}$ could be considered a upper bound for the approximation error, one would like to see it is in moderate scale.

Figure 4.12 shows the total centers set that used in S_L in different level. In this case, the error indicator and the above algorithm together generated these 12 centers set in 8 levels. Level 7, has 3 centers sets, but these points are very close and are not readily distinguishable from the figure.

Figure 4.13 shows the process with $\theta_{\text{refine}} = 1(-8)$, staring with 13 sample points, that is $|\mathbf{M}_1| = |\mathbf{X}_1^1| = 13$. The algorithm stops with total 50 sample sets and total 476 sample points in 12 level. The final approximation S_L has $e_{S_L}(f) = 1.0(-8)$ and RMS$(e_{S_L}(f)) = 2.7(-9)$. The largest condition number of interpolation matrix is 9.6(+6). One sees a convergence rate $|e_{\mathbf{M}_L}(f)_{|\mathbf{M}_L|}| < C \times |\mathbf{M}_L|^{-2.5}$ for this case.

Table 4.4 compares the approximation accuracy and the total number of function evaluation needed by the adaptive error indication (AEI) algorithm, the residual sub-sampling (RS) method and the multilevel adaptive error indicator (MAEI) approximation . In this example, the multilevel scheme achieves a better result with notably less function evaluations at the the lower $e_{\mathbf{X}}(f)$ of 1.0(−8).

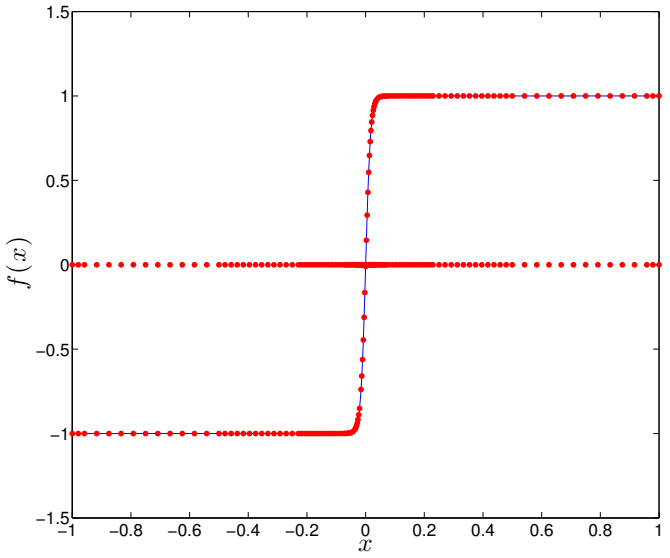

FIGURE 4.11: Hyperbolic tan function approximation with final sample points distribution, initial $|\mathbf{X}_1^1| = 13$ with $\theta_{\text{refine}} = 1.0(-5)$.

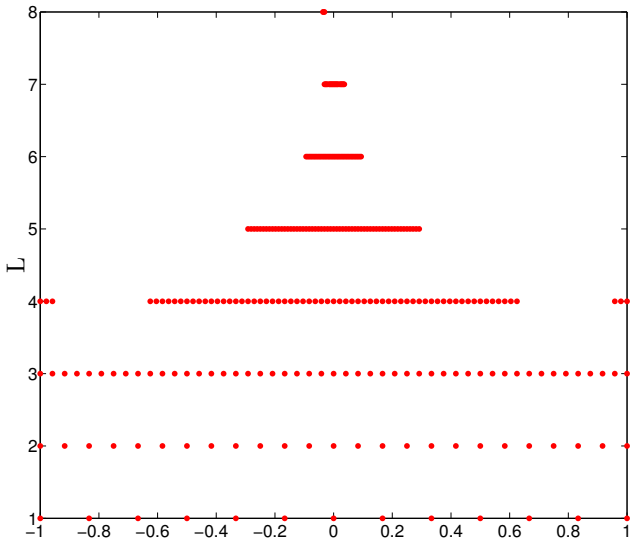

FIGURE 4.12: The centers sets used in $\tanh(60x-0.1)$ multilevel error indicator approximation, with $\theta_{\text{refine}} = 1.0(-5)$.

4.5.1.3 The shifted absolute value function

The final univariate example is $f(x) = |x - 0.04|$. In the error indicator function, the interpolant that used all the available sample points is $S^{\text{multi}}_{\mathbf{M}_L}$, the multiquadric RBF is applied, the shape parameter c_i of each center $\mathbf{x}_i$ is set be a constant

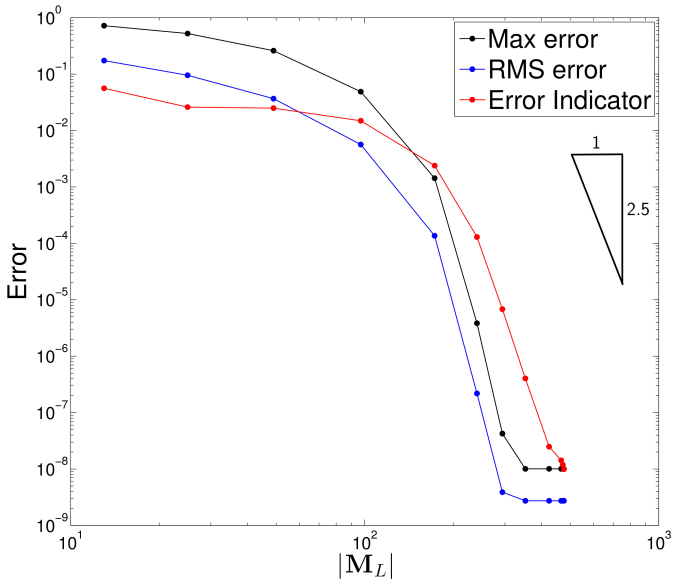

FIGURE 4.13: Hyperbolic tan function approximation error for each level L, $|\mathbf{M}_L|$ is the total number of samples of the target function. $\theta_{\text{refine}} = 1.0(-8)$.

Method	$e_{\mathbf{X}}(f)$	N_{total}
RS	2.5(-5)	441
AEI	1.1(-5)	141
MAEI	9.2(-6)	139
RS	1.7(-7)	5671
AEI	1.7(-8)	726
MAEI	1.0(-8)	476

TABLE 4.4: Comparison of methods for $f(x) = \tanh(60x - 0.1)$.

divided by the distance to nearest neighbour h_i, that is

$$c_i = 3/h_i.$$

For the interpolants $\Delta S^{\text{multi}}_{\mathbf{X}_i^j}, 2 \leq i \leq L, 1 \leq j \leq n_i$, the multiquadric RBF is used, and the shape parameter c_v is set as in Equation 4.14 and the $C = 1.5$ for this case.

Figure 4.14 shows the sample points clustered around the derivative discontinuity of $|x - 0.04|$ as expected. The final approximation representation S_L used total 15 sample sets, Table 4.5 shows the approximation S_L staring with 13 uniformly

distributed sample points, and ending with 101 points. The final approximation S_L has L-infinity and root mean square errors $3.6(-5)$ and $6.9(-6)$ respectively.

Figure 4.15 shows the total centers set that are used in S_L in different levels. In this case, the error indicator and the above algorithm together generated these 15 centers set in 12 levels.

Figure 4.16 shows the progress of the multilevel approximation scheme staring with $|\mathbf{X}_1^1| = 13$, and $\theta_{\text{refine}} = 2.0(-8)$. The process terminates when the level L reaches 22, with total 58 sample sets. The final approximation S_L with total 426 sample points has maximum error $e_{S_L}(f) = 3.6(-8)$ and $\text{RMS}(e_{S_L}(f)) = 5.5(-9)$. One again sees a convergence rate $|e_{\mathbf{M}_L}(f)_{|\mathbf{M}_L|}| < C \times |\mathbf{M}_L|^{-2.5}$ for this case.

Table 4.5 compares the results and function evaluations required for residual subsampling method, the adaptive error indicator algorithm and this multilevel approximation scheme. In this example, the multilevel approximation scheme still could deliver slightly better results with slightly less total evaluations from the target function. The author would like to emphasis that the applications include cases where function evaluation is expensive.

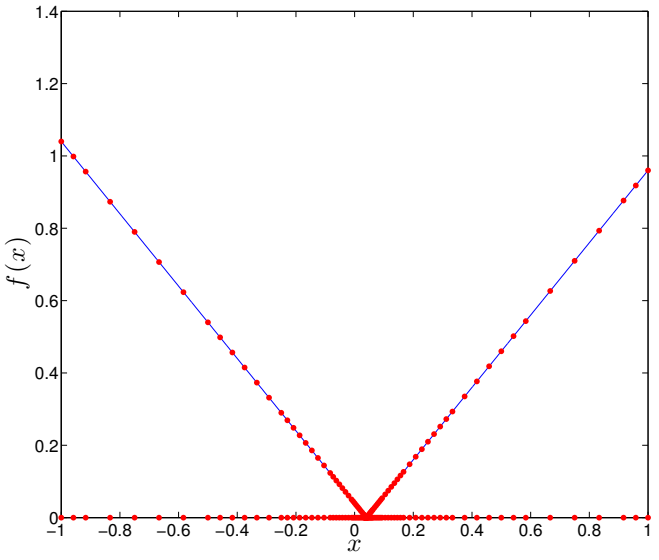

FIGURE 4.14: $f(x) = |x - 0.04|$ multilevel approximation with final sample points distribution, initial $|\mathbf{X}_1^1| = 13$ with $\theta_{\text{refine}} = 2.0(-5)$.

L	m_L	$\lvert\mathbf{M}_L\rvert$	$e_{S_{m_L}}(f)$	RMS$(e_{S_{m_L}}(f))$	$\kappa(A)_{max}$
1	1	13	3.7(-2)	6.4(-3)	3.3(+3)
2	2	25	2.7(-2)	3.0(-3)	1.2(+4)
3	5	40	7.2(-3)	9.8(-4)	1.4(+4)
4	6	54	3.4(-3)	3.0(-4)	1.7(+4)
5	7	66	1.6(-3)	9.3(-5)	1.2(+4)
6	8	76	1.4(-3)	4.2(-5)	8.8(+3)
7	9	82	7.7(-4)	1.7(-5)	3.3(+3)
8	10	87	3.3(-4)	8.9(-6)	2.3(+3)
9	11	92	2.1(-4)	7.5(-6)	2.3(+3)
10	12	93	3.8(-5)	7.0(-6)	4.6(+1)
11	13	97	4.9(-5)	7.0(-6)	6.7(+3)
12	15	101	3.6(-5)	6.9(-6)	3.2(+2)

TABLE 4.5: Approximation process of multilevel scheme for $|x - 0.04|$, with $\theta_{\text{refine}} = 2.0(-5)$.

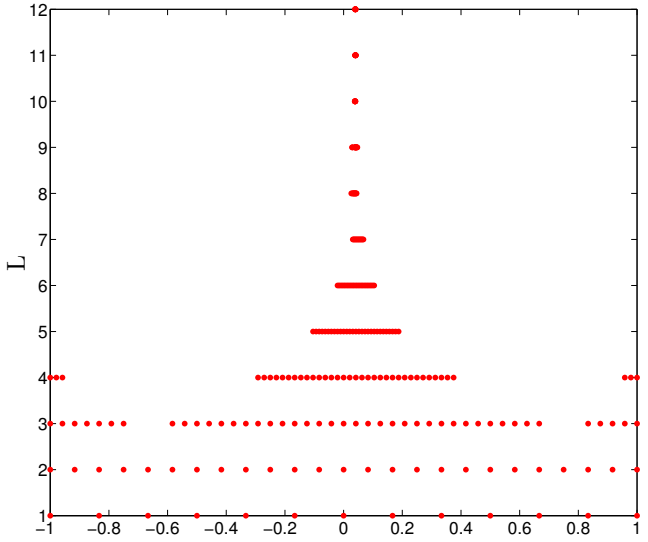

FIGURE 4.15: The centers sets used in $f(x) = |x-0.04|$ multilevel error indicator approximation, with $\theta_{\text{refine}} = 2.0(-5)$.

4.5.2 Two Dimensional Function Multilevel Approximation

The author now considers two two-dimensional examples, where the node adding scheme explained above is applied. The author used uniformly distributed nodes as $\mathbf{X}_1^1$ and $|\mathbf{X}_1^1| = 121$. A test set T of 101×101 uniformly spaced nodes on $[-1, 1]^2$ is used to test the approximation quality: $e_{S_L}(f) = \max_{t \in T} |f(t) - S_L(t)|$ and the

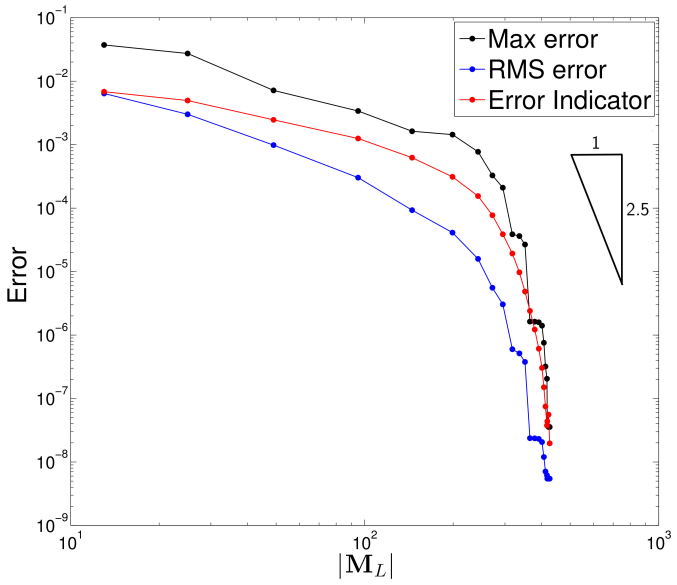

FIGURE 4.16: $f(x) = |x - 0.04|$ multilevel approximation error for each level L, $|\mathbf{M}_L|$ is the total number of samples of the target function. $\theta_{\text{refine}} = 2.0(-8)$.

Method	$e_{\mathbf{X}}(f)$	N_{total}
RS	3.6(-5)	674
AEI	3.8(-5)	121
MAEI	3.6(-5)	101
RS	1.8(-7)	5638
AEI	3.8(-8)	459
MAEI	3.6(-8)	426

TABLE 4.6: Comparison of methods for $f(x) = |x - 0.04|$.

root mean square value in $f(t) - S_L(t), t \in T$, that is $\text{RMS}(e_{S_L}(f))$.

In the following examples, the interpolant S_1 is set to be the multiquadric interpolant $S^{\text{multi}}_{\mathbf{X}_1^1}$, and the shape parameter c is set to be a constant and equal to 1 for each node in this interpolant.

4.5.2.1 The modified Franke Function

The modified Franke function of Equation 3.19 is now considered. For the interpolants $\Delta S^{\text{gau}}_{\mathbf{X}_i^j}, 2 \leq i \leq L, 1 \leq j \leq n_i$, the Gaussian RBF is applied. The shape

parameter c_v for $\Delta S^{\text{gau}}_{\mathbf{X}_i^j}$ is set as

$$c_v = \frac{C}{h_v}, \tag{4.15}$$

where h_v is the nearest distance between $\mathbf{x}_v$ and other centers in set $\mathbf{X}_i^j = \{\mathbf{x}_v, w = 1, 2 \cdots, V_i^j\}$. In this case, $C = 200$. In the error indicator, the interpolant that used all available sample points is $S^{\text{cub}}_{\mathbf{M}_L}$, the cubic RBF is applied.

With $\theta_{\text{refine}} = 5.0(-5)$ only 5 iterations are needed to reach the stopping criteria. Figure 4.17 shows the final nodes distribution and demonstrates that error indicator locates more points in regions of rapid variation. In this case, we have $|\mathbf{M}_L| = 1964$ nodes with $e_{S_L}(f) = 8.8(-5)$. Table 4.7 shows results corresponding to different values of θ_{refine}.

Figure 4.18 compares the results of approximation methods for modified Franke function, the y-axis is the total evaluation number from the target function and the x-axis is the maximum error of the approximation. The red line stands for multilevel error indicator approximation, the blue line stands for the adaptive error indicator approximation and black line stands for residual sub-sampling method. One could see clearly that the multilevel RBF approximation outperforms largely. One sees that error decays approximately like $|e_{\mathbf{M}_L}(f)_{|\mathbf{M}_L|}| < C \times |\mathbf{M}_L|^{-1.5}$ for this case, which $|\mathbf{M}_L|$ is the total number of nodes that used in this approximation.

θ_{refine}	$\lvert\mathbf{M}_L\rvert$	$e_{S_L}(f)$	RMS$(e_{S_L}(f))$
5.0(-4)	990	4.8(-4)	8.1(-5)
2.0(-4)	1231	2.5(-4)	3.5(-5)
1.0(-4)	1554	2.4(-4)	1.7(-5)
5.0(-5)	1964	8.8(-5)	7.5(-6)

TABLE 4.7: Multilevel level approximation results of the modified Franke function with different θ_{refine}.

4.5.2.2 The two-dimension exponential function

In this example, target function is $f(x, y) = \exp(-60((x-0.35)^2+(y-0.25)^2))+0.2$ (Figure 4.19) in $[-1, 1]^2$. For the interpolants $\Delta S^{\text{multi}}_{\mathbf{X}_i^j}, 2 \leq i \leq L, 1 \leq j \leq n_i$, the

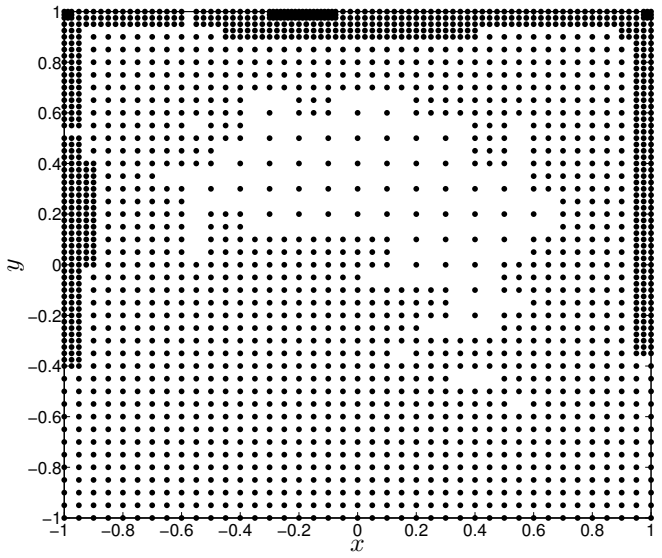

FIGURE 4.17: Final nodes distribution for multilevel approximation for the modified Franke function with $\theta_{\text{refine}} = 5.0(-5)$. The number of total sampled points is 1964.

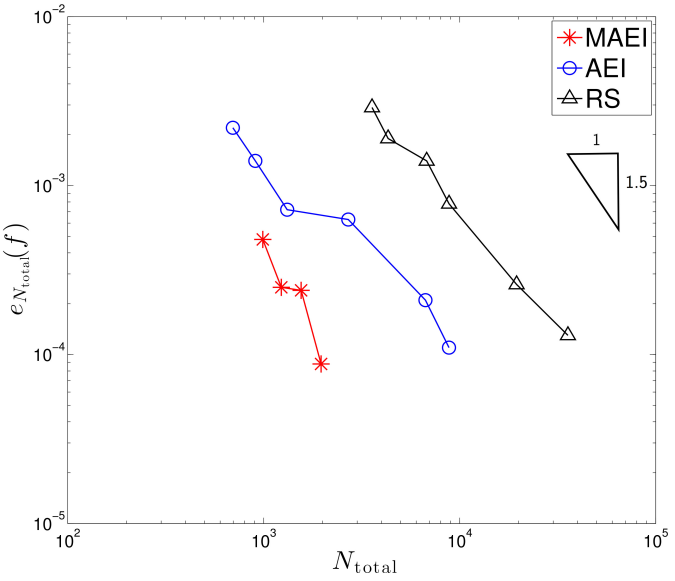

FIGURE 4.18: Comparison of methods for the modified Franke function

MQ RBF is applied. The shape parameter c_v for $\Delta S^{\text{multi}}_{\mathbf{X}_i^j}$ is set as Equation 4.15, which $C = 50$. In the error indicator, the interpolant that used all available sample points is $S^{\text{cub}}_{\mathbf{M}_L}$, the cubic RBF is applied.

Table 4.8 shows how the multilevel approximation depend on the error indicator.

Figure 4.20 shows how the error indicator puts more nodes in the region where the function changes rapidly. Figure 4.21 compares the results of approximation methods for this function. We could see clearly that the multilevel adaptive error indicator RBF approximation outperforms the residual sub-sampling method largely and when higher accuracy level is needed the multilevel RBF approximation shows its ability. One again sees that error decays approximately like $|e_{\mathbf{M}_L}(f)_{|\mathbf{M}_L|}| < C \times |\mathbf{M}_L|^{-1.5}$.

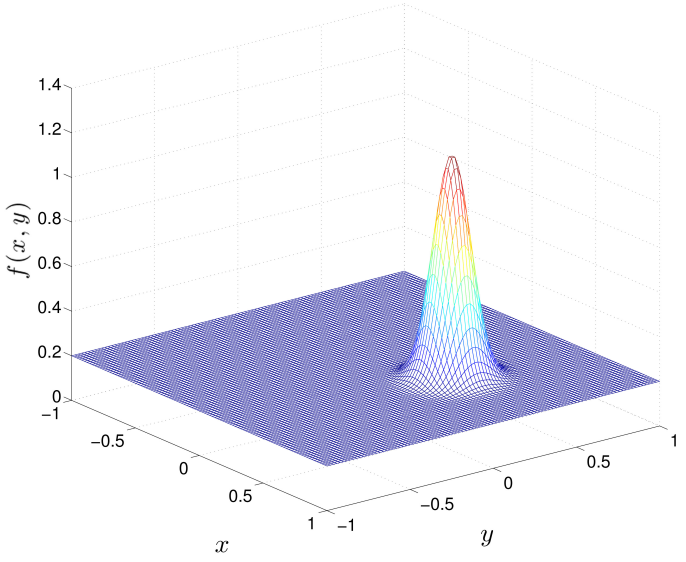

FIGURE 4.19: The two-dimension exponential function multilevel adaptive error indicator (MAEI) approximation result.

θ_{refine}	$\lvert\mathbf{M}_L\rvert$	$e_{S_L}(f)$	$\text{RMS}(e_{S_L})(f)$
5.0(-4)	1261	5.2(-4)	8.7(-5)
2.0(-4)	1543	1.8(-4)	2.2(-5)
1.0(-4)	1625	9.1(-5)	1.2(-5)
5.0(-5)	1796	8.1(-5)	6.0(-6)

TABLE 4.8: Multilevel level approximation results of this two-dimension exponential function with different θ_{refine}.

4.5.2.3 The modified Rosenbrock function

The third test function is the modified Rosenbrock function $f(x,y) = (1-x)^2 + 5(y-x^2)^2$ on $[-1,1]^2$, see Figure 4.22. For the interpolants $\Delta S^{\text{gau}}_{\mathbf{X}_i^j}, 2 \leq i \leq L, 1 \leq$

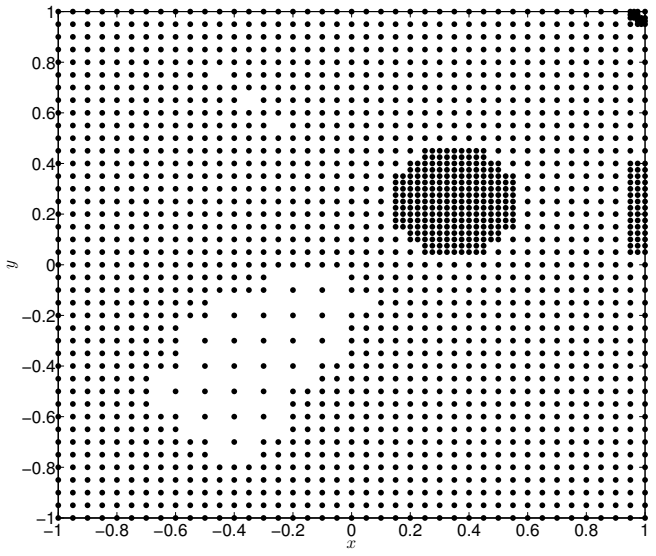

FIGURE 4.20: Final nodes distribution for multilevel approximation for the his two-dimension exponential function with $\theta_{\text{refine}} = 5.0(-5)$. The number of total sampled points is 1796.

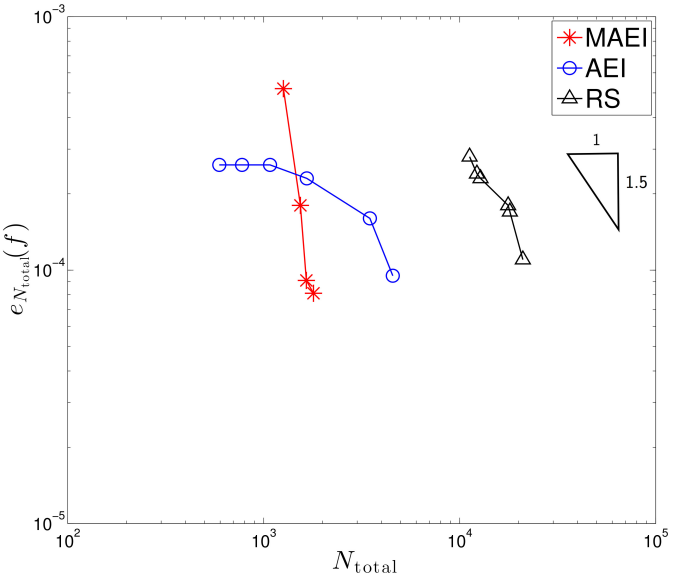

FIGURE 4.21: Comparison of methods for this two-dimension exponential function.

$j \leq n_i$, the Gaussian RBF is applied. The shape parameter c_v for $\Delta S^{\text{gau}}_{\mathbf{X}_i^j}$ is set as Equation 4.15, which $C = 300$. In the error indicator, the interpolant that used all available sample points is $S^{\text{multi}}_{\mathbf{M}_L}$, the MQ RBF is applied. The shape parameter c for $S^{\text{multi}}_{\mathbf{M}_L}$ is simply set to constant 1.

With $\theta_{\text{refine}} = 5.0(-5)$ the algorithm took 4 levels to reach the stopping criteria. A total of $|\mathbf{M}_{L=4}| = 1717$ nodes were used to give an error $e_{S_L}(f) = 4.9(-5)$. Figure 4.23 shows how the error indicator puts more points near the boundaries where the function value rapidly increases. Table 4.9 shows how the number of points needed by the multilevel scheme varies with the choice of θ_{refine}. Figure 4.24 shows the comparison between RE, AEI and MAEI. This modified Rosenbrock function is difficult to approximate due to its rapid change in the corner of the domain. One could see multilevel level approximation could deliver better accuracy with less total nodes applied. One still sees a convergence rate $|e_{\mathbf{M}_L}(f)_{|\mathbf{M}_L|}| < C \times |\mathbf{M}_L|^{-1.5}$ for this case.

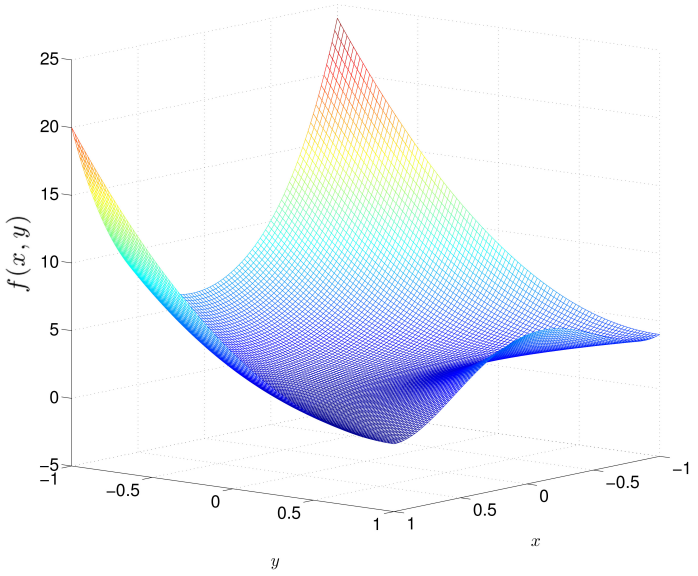

FIGURE 4.22: The modified Rosenbrock function multilevel adaptive error indicator (MAEI) approximation result.

θ_{refine}	$\lvert\mathbf{M}_L\rvert$	$e_{S_L}(f)$	RMS$(e_{S_L})(f)$
5.0(-4)	942	6.2(-4)	1.5(-4)
2.0(-4)	1305	1.8(-4)	4.0(-5)
1.0(-4)	1521	9.8(-5)	1.6(-5)
5.0(-5)	1717	4.9(-5)	5.3(-6)

TABLE 4.9: Multilevel level approximation results of this modified Rosenbrock function with different θ_{refine}.

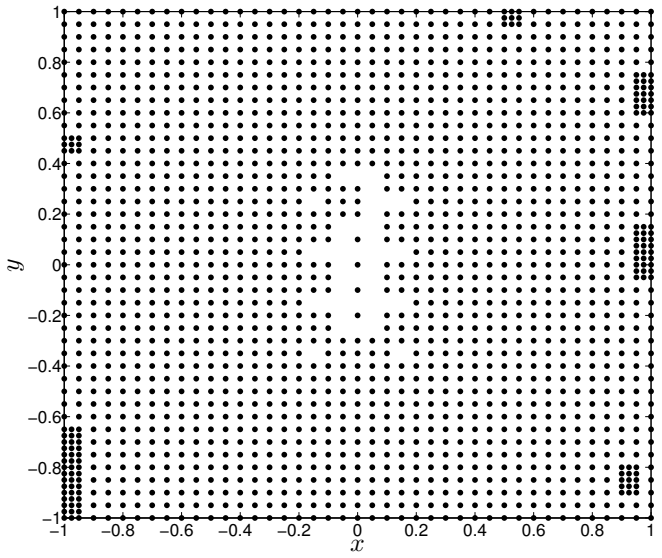

FIGURE 4.23: Final nodes distribution for multilevel approximation for the modified Rosenbrock function with $\theta_{\text{refine}} = 5.0(-5)$. The number of total sampled points is 1717.

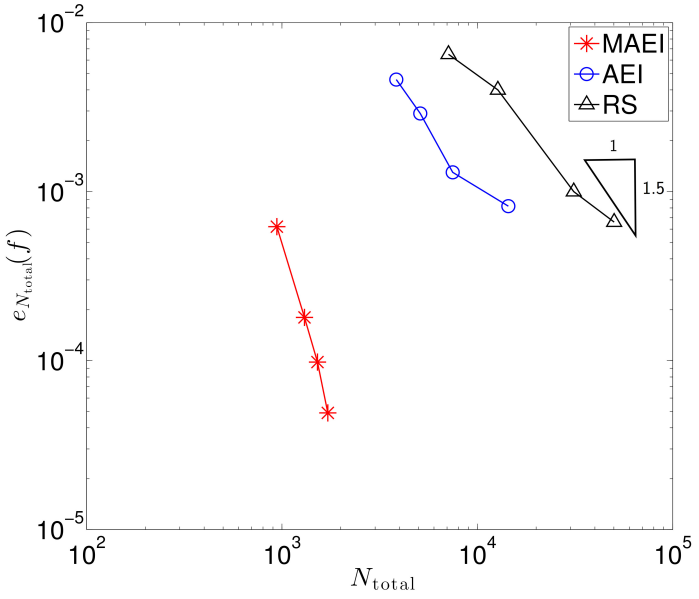

FIGURE 4.24: Comparison of methods for modified Rosenbrock function.

In Section 4.1 4.2 4.3 and 4.4, the author has described the structure of the multilevel adaptive error indicator (MAEI) approximation scheme, the components are:

1. The first level global interpolant $S^{\text{multi}}_{\mathbf{X}_1^1}$.

2. The interpolant $\Delta S_{\mathbf{X}_i^j}$ for following levels.

3. Then interpolant S_{M_L} which use all the available center.

4. Then algorithms that generating $\mathbf{X}_i^j$, M_k and corresponding indication set Ξ_k.

In MAEI method, when compared to AEI, there is more parameters to be user defined. The user defined parameters in MAEI are:

1. The shape parameter c for $\Delta S_1^{\text{multi}}$.

2. The basis function type and its shape parameter for $\Delta S_{\mathbf{X}_i^j}, i = 2, \cdots, L, j = 1, \cdots, n_i$.

3. The basis function type and its shape parameter for S_{M_L}.

In this thesis, the author could not propose a parameter setting suited for all the test functions. The multilevel adaptive error indicator (MAEI) approximation scheme could deliver desirable accuracy with much less sources used, however, it is not as robust as adaptive adaptive error indicator (AEI) method.

4.6 Summary

Still motivated by the kind of problem mention in Chapter 1, the author combines the adaptive error indicator (AEI) RBF interpolation method and the multilevel approximation scheme using a domain decomposition approach to have a new method called multilevel adaptive error indicator (MAEI) RBF approximation scheme. Presented preliminary numerical results have revealed that the fundamental idea of MAEI is useful and MAEI is more efficient than AEI.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

The author has constructed a new adaptive algorithm via error indicator functions for the radial basis functions (RBFs) method applied to interpolation problems. Nodes can be added and removed based on residuals evaluated at a finer point set. The author has proposed the use of an error indicator based the idea that different approximation methods should give different results in regions where it is difficult to approximate. The author compares the global RBF interpolant with a local RBF interpolant to provide a quantitative measure of the approximation error. The error indicator assigns a value to the current global interpolation and this value describes the approximation quality. According to this error indicator value, an area with poor approximation quality and an area with good quality are determined. This detection process requires no unnecessary sampling form the target function, so it provides a considerable saving in time especially where the target function is costly to evaluate.

Applying this error indicator gives an adaptive interpolation algorithm, with the following steps: "approximation - detect (achieved by error indicator) - refine/-coarse - approximation". On the examples the author has presented in one, two and three dimensions, this adaptive algorithm provides an accuracy level which may be pre-set by the user. One could observe in the examples how effective the

indicator is in automatically clustering centers in regions of high variation of the target function.

In order to deliver reliable approximation, the condition number of interpolation matrix $\kappa(A)$ should be kept to a moderate scale. In this thesis, an adaptive shape parameter for the multiquadric RBF has been used and it was observed that, as theoretically predicted, this is effective in keeping $\kappa(A)$ to a reasonable size.

In the monograph of Buhmann [8], he mentioned there is still an open question on existence and uniqueness of interpolant which every centers has different shape parameters. In order words, there is yet not theoretical proof for the invertibility of the interpolation matrix A where it is fitted with center-dependent shape parameters. The application of adaptive shape parameter, also known as centers-dependent shape parameters, regrettably jeopardizes the symmetry of the interpolation matrix A. It breaks the proof of its non-singularity. From the practical prospective, nevertheless, one has observed a substantial benefit from using the adapting the shaper parameters. Consequently, one sees the importance of devoting more efforts into the direction of understanding this issue.

The numerical experiments have shown that the adaptive error indicator (AEI) approximation provides a similar global accuracy to the RS method with fewer evaluations of the target function. This is a desirable property, especially when the target function is expensive to evaluate once. One might worry that the AEI method could not reach the stop criteria for some cases, however, in previous examples, the AEI method could always reach the stop criteria and deliver good accuracy. In future work, the author would like to explore the adaptive θ_{refine} and θ_{coarse} thresholds.

The author has shown that the choice of parameters in the approximation model do not have a significant effect on the quality of the results, suggesting that the method described is robust.

In Chapter 3, the author proposed the parameter setting for the AEI method which is named Parameter Set 1. The adaptive error indicator RBF interpolation method with Parameter Set 1 delivers good approximation ability and certain

robustness for different approximation problems. In the future, one would like to develop a technique for selecting more optimal parameters for the AEI while maintaining an appropriate robustness.

The adaptive error indicator (AEI) RBF interpolation method discussed in Chapter 3 is still a global method in the sense that all the selected centers are applied to build the global interpolant $S^{\text{multi}}_{\mathbf{X}_k}$. The varying shape parameter based on the nearest distance between adjacent neighbours is decisive. However, in some extreme cases, such as with functions with very steep slopes, that global interpolant with center-dependent shape parameters is not always reliable. Namely, the condition number of interpolation matrix $\kappa(A)$ is still large, which ruins the global interpolation.

In addition to the ill-conditioning problem, the size of interpolation matrix A grows with the number of centers. Consequently, it is impractical to solve this large linear system with direct methods. One possible mitigation for this is to apply indirect methods to solve this linear system.

The author also investigated the multilevel level approximation schemes and proposed a new RBF multilevel adaptive approximation scheme via error indicators. This multilevel adaptive error indicator (MAEI) method is a hybridization between adaptive error indicator (AEI) method and multilevel interpolation method via a domain decomposition technique. This error indicator is a modified version of the error indicator in Chapter 3. This method compares the current RBF approximation with a global RBF interpolant to provide a quantitative measure of approximation error. The current RBF approximation is achieved by a multilevel adaptive scheme, which consist of local approximations of residuals at certain subdomains. This error indicator also follows the idea that the deviation between global and local approximations could indicate the error effectively.

The author also showed some preliminary approximation results achieved by this multilevel adaptive approximation scheme. The author compared these results with the residual sub-sampling (RE) method. It turns out that the multilevel adaptive error indicator (MAEI) approximation provides the same accuracy with

fewer evaluations from the target function compared with the adaptive error indicator (AEI) interpolation.

The current MAEI method applies a MQ basis function in the first level interpolation. In the rest levels, another basis function has been applied. Different basis functions have been applied to different target functions based on the author's experiments and experiences. In order to improve the completeness of the MAEI method, the automatic selection of basis function for different levels and different target functions should be explored in the future.

5.2 Future Work

In the future, the author would like to merge local RBF interpolants with error indicator to form the localised adaptive error indicator RBF interpolation. The author believes the localised adaptive error indicator method will be highly promising solution to the approximation problem which target functions are with very steep slopes. Figure 5.1 shows an example in two dimensional case of the local RBF interpolant. Given a set of scattered data points and corresponding function value, one would like to know the function value at one certain point $\mathbf{x}$. The main difference from global RBF interpolation is to use subset of total centers to build a local RBF interpolant. The usage of relative small amount of centers to form the local interpolant avoids a very large interpolation matrix. The local interpolant is created by only a subset of RBF centers (the solid dots). As long as the local interpolant is established, the function value at $\mathbf{x}$ can be compute by it.

The data structure for this localised adaptive error indicator RBF interpolation could be inherited from [45], see Figure 5.2 for a example. In Figure 5.2, the solid dot stands for the centers $\mathbf{X}$ and the red cross stands for the indication points Ξ. The current approximated value at indication points are computed by the local interpolant whose centers are inside the circle. The centers inside the circle could be called as N_{circle}. The error indicator still follows the idea that two different kind of RBF interpolant give different approximations at the points where is difficult

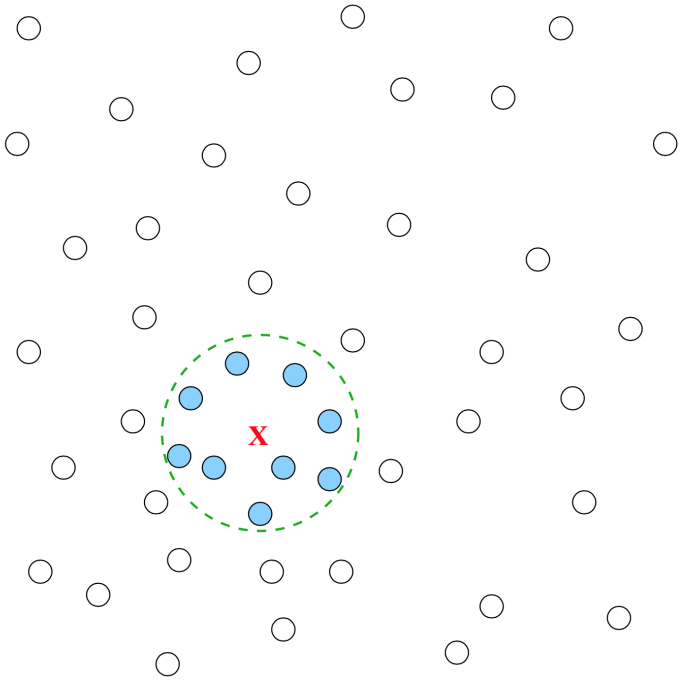

FIGURE 5.1: Example of localized RBF interpolant.

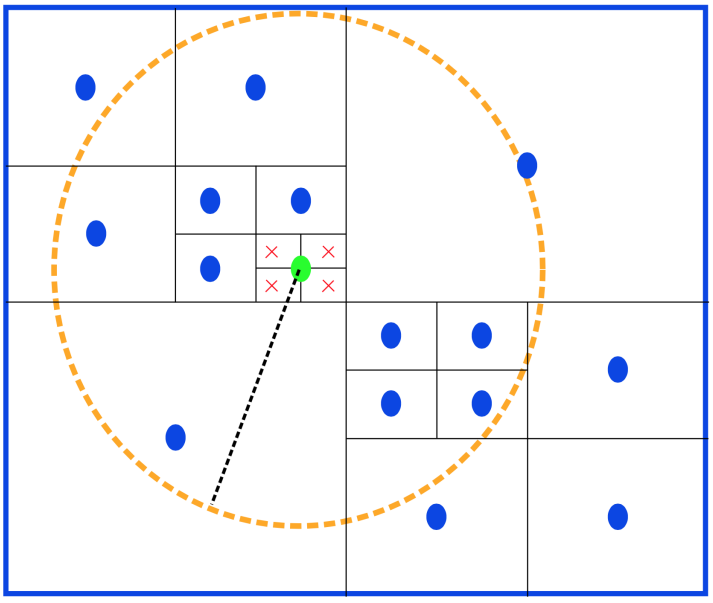

FIGURE 5.2: Example of possible data structure for localized adaptive error indicator RBF interpolation.

to approximate. In order to decide the indication point ξ should turn into center

or not, one computes the error indicator function at ξ, that is

$$\eta(\xi) = |S^{\text{multi}}_{N_{\text{circle}}}(\xi) - S^{\text{ps}}_{N_{\text{circle}}}(\xi)|.$$

The author believes that this error indicator could measure the actual error and guided us to have a series of local interpolants that could represent the unknown target function well.

In multilevel adaptive error indicator (MAEI) approximation scheme, the author would like to apply the error indicator that defined in Definition 3.1. In Chapter 3, it has showed the robustness of this error indicator.

Considering $f - \Delta S_i$ as target function, then apply AEI method to generate approximation ΔS_{i+1} for it. Repeat this step could generate $\Delta S_{\mathbf{X}_1}, \Delta S_{\mathbf{X}_2}, \cdots, \Delta S_{\mathbf{X}_L}$, which $\Delta S_{\mathbf{X}_i}$ is defined as Equation 4.9 and the sub-domain Ω_i is decided by the error indicator in AEI method. Then

$$s = \sum_{i=1}^{L} \Delta S_{\mathbf{X}_i},$$

is the multilevel adaptive level approximation for target function f.

The author conjectures that above mentioned points should be studied in the future in order to improve the adaptive error indicator (AEI) method and multilevel adaptive error indicator (MAEI) method.

Bibliography

[1] J. P. Agnantiaris, D. Polyzos, and D. E. Beskos. Some studies on dual reciprocity BEM for elastodynamic analysis. *Computational Mechanics*, 17:270-277, 1996.

[2] I. Barrodale, M. Berkley and D. Skea. Warping digital images using thin plate splines. presented at the *Sixth Texas International Symposium on Approximation Theory* Austin, January 1992.

[3] J. Behrens and A. Iske. Grid-free adaptive semi-lagrangian advection using radial basis functions. *Computers and Mathematics with Applications*, 43(3-5):319–327, 2002.

[4] M. Bozzini, L. Lenarduzzi, and R. Schaback. Adaptive interpolation by scaled multiquadrics. *Advances in Computational Mathematics*, 16:375–387, 2002.

[5] M. Bozzini, L. Lenarduzzi, M. Rossini, and R. Schaback. Interpolation with variably scaled kernels. *IMA Journal of Numerical Analysis*, 35(1):199-219, 2015.

[6] M. Bozzini and M. Rossini. Testing Methods for 3D Scattered Data Interpolation *Monografia de la Academia de Ciencias de Zaragoza*, 20:111-135, 2002.

[7] M.D. Buhmann. Spectral convergence of multiquadric interpolation. *Proceeding of the Edinburgh Mathematical Society*, 36:319–333, 1993.

[8] M.D. Buhmann. *Radial Basis Functions: Theory and Implementations.* Cambridge University Press, Cambridge, 2003.

[9] R. E. Carlson and E. J. Kansa. Improved accuracy of multiquadric interpolation using variable shape parameters. *Computers and Mathematics with Applications*, 24(12):99–120, 1992.

[10] Y.A. Cengel and M.A Boles Thermodynamics : An Engineering Approach 8th Edition *McGraw-Hill Education*, 2014

[11] T. Cecil, J. Qian, and S. Osher. Numerical method for high dimensional hamilton-jacobi equation using radial basis functions. *Journal of Computational Physics*, 196:327–347, 2004.

[12] C. S. Chen, M. A. Golberg, and S. Karur. Improved multiquadric approximation for partial differential equations. *Engineering Analysis with Boundary Elements*, 18(1):9–17, 1996.

[13] C. de Boor. Multivariate approximation. *The State of Art in Numerical Analysis*, Oxford University Press, Oxford, 87-110, 1982.

[14] A. M. del Rey, G. R. Sanchez and A. de la Villa Cuenca A Protocol to Cipher Digital Images Based on Cat Maps and Cellular Automata. *Pattern Recognition and Image Analysis*, Third Iberian Conference, IbPRIA 2007, Girona, Spain, June 6-8, 2007, Proceedings, Part I, 571-578, 2007.

[15] Z. Dong, E.H. Georgoulis, J. Levesley, and F. Usta. Fast multilevel sparse Gaussian kernels for high-dimensional approximation and integration. *Research Report*, University of Leicester, 2015.

[16] T. A. Driscoll, and B. Fornberg. Interpolation in the limit of increasingly flat radial basis functions. *Computers and Mathematics with Applications*, 43(3):413–422, 2002.

[17] T. A. Driscoll and A. R. H. Heryudono. Adaptive residual subsampling methods for radial basis function interpolation and collocation problems. *Computers and Mathematics with Applications*, 53(6):927–939, 2007.

[18] T. A. Driscoll and R. B. Platte. Polynomials and potential theory for Gaussian radial basis function interpolation. *SIMA Journal of Numerical Analysis*, 43(2):750–766, 2005.

[19] M. R. Dubal, R. A. Matzner, and S. R. Olvera. *In approaches to numerical relativity.* Cambridge University Press, Cambridge, 1993.

[20] G. E. Fasshauer and M. J. McCourt. Stable Evaluation of Gaussian Radial Basis Function Interpolants. *SIAM Journal on Scientific Computing*, 34(2):A737-A762, 2012.

[21] G. E. Fasshauer. *Meshfree approximation methods with MATLAB.* volume 6 of *Interdisciplinary Mathematical Science.* World Scientific, 2007.

[22] G. E. Fasshauer, A. Q. M. Khaliq, and D. A. Voss. Using meshfree approximation for multi-asset American option problems. *Chinese Institute Engineers*, 24:563–571, 2004.

[23] G. E. Fasshauer and J. G. Zhang. On choosing optimal shape parameters for RBF approximation. *Numerical Algorithms*, 45(1-4):345–368, 2007.

[24] A. I. Fedoseyev, M. J. Friedman, and E. J. Kansa. Improved multiquadric method for elliptic partial differential equations via PDE collocation on the boundary. *Computers and Mathematics with Applications*, 43(3-5):439-455, 2002.

[25] R. A. Finkel and J. L. Bentley. Quad Trees: A Data Structure for Retrieval on Composite Keys. *Acta Informatica*, 4:1-9, 1974.

[26] M. S. Floater and A. Iske. Multistep scattered data interpolation using compactly supported radial basis functions. *Journal of Computational and Applied Mathematics*, 73:65-78, 1996.

[27] M. S. Floater and A. Iske. Thinning algorithms for scattered data interpolation. *BIT Numerical Mathematics*, 38(4):705-720, 1998.

[28] M. S. Floater and A. Iske. Thinning, inserting, and swapping scattered data. *Surface Fitting and Multi-Resolution Methods*, 139-144. Vanderbilt University Press, Nashville, 1996.

[29] N. Flyer and B. Fornberg. Accuracy of radial basis function interpolation and derivative approximations on 1-D infinite grids. *Advances in Computational Mathematics*, 23:5-20, 2005.

[30] N. Flyer, B. Fornberg, and E. Larsson. Stable computation with Gaussian radial basis functions. *SIAM Journal on Scientific Computing*, 33(2):869–892, 2011.

[31] B. Fornberg and E. Larsson. A numerical study of some radial basis function based solution methods for elliptic PDEs. *Computers and Mathematics with Applications*, 46(5-6):891–902, 2003.

[32] B. Fornberg and C. Piret. Stable computation of multiquadric interpolant for all value of shape parameter. *SIAM Journal on Scientific Computing*, 30(1):60–80, 2007.

[33] B. Fornberg and G. Wright. Stable computation of multiquadric interpolants for all values of the shape parameter. *Computers and Mathematics with Applications*, 48(5-6):853-867, 2004.

[34] R. Franke. A critical comparison of some methods for the interpolation of scattered data. *Technical Report NPS-53-79-03*, Naval Postgraduate School, 1979.

[35] C. Franke and R. Schaback. Convergence order estimates of meshless collocation methods using radial basis functions. *Advances in Computational Mathematics*, 8:381–399, 1998.

[36] R. Franke. Scattered data interpolation: test of some methods. *Mathematics of Computation*, 38(157):181–200, 1982.

[37] R. Franke. Recent advances in the approximation of surfaces from scattered data. *Topics in Multivariate Approximation* Academic Press, New York, 79-98, 1987.

[38] E. H. Georgoulis, J. Levesley, and F. Subhan. Multilevel sparse kernel-based interpolation. *SIAM Journal on Scientific Computing*, 35(2):A815–A831, 2013.

[39] M. Golomb and H. F. Weinberger. Optimal approximation and error bounds. *Numerical Approximation*, 117–190, 1959.

[40] D. Gong, C. Wei, L. Wang, L. Feng, and L.D. Wang. Adaptive Methods for Center Choosing of Radial Basis Function Interpolation: A Review. *Lecture Notes in Computer Science*, 6733: 573-580, 2010.

[41] D. Gong, J. Chang, and C. Wei. An adaptive method for choosing center sets of RBF interpolation. *Journal of Computers*, 6(10): 2112-2229, 2011.

[42] R. L. Hardy. Multiquadric equations of topography and other irregular surfaces. *Journal of Geophysical Research*, 76(8):1905-1915, 1971.

[43] R. L. Hardy. Theory and applications of the multiquadric-biharmonic method. *Computers and Mathematics with Applications*, 19(8/9):163-208, 1990.

[44] J. G. Hayes. Fitting surfaces to data. *The Mathematics of Surfaces*, Oxford University Press, Oxford, 17-38.

[45] A. R. H. Heryudono. Adaptive radial basis function methods for the numerical solution of partial differential equations, with application to the simulation of human tear film. *Ph.D. thesis of University of Delaware*, 2008.

[46] Y. C. Hon and R. Schaback. On unsymmetric collocation by radial basis functions. Applied Mathematics and Computation, 119(2-3):177-186, 2001.

[47] Y. C. Hon. A quasi-radial basis functions methods for American option pricing. *Computers and Mathematics with Applications*, 43:513–524, 2002.

[48] Y. C. Hon and X. Z. Mao. A radial basis function method for solving option pricing model. *Financial Engineering*, 8:1–24, 1999.

[49] Y. C. Hon and Z. Wu. Convergence error estimate in solving free boundary diffusion problem by radial basis functions method. *Engineering Analysis with Boundary Elements*, 27:73–79, 2003.

[50] A. Iske. Hierarchical scattered data filtering for multilevel interpolation schemes. *Mathematical Methods for Curves and Surfaces*, 211-220. Vanderbilt University Press, Nashville, 2001.

[51] A. Iske. Optimal distribution of centers for radial basis function methods. *Technical Report TUM-M0004*, Technische Universitat Maunchen, 2000

[52] A. Iske and J. Levesley. Multilevel scattered data approximation by adaptive domain decomposition. *Numerical Algorithms* , 32 (1):13-25, 2003.

[53] E. J Kansa. Multiquadrics - a scattered data approximation scheme with applications to computational fluid-dynamics - I. *Computers and Mathematics with Applications*, 19(8-9):127-145, 1990.

[54] E. J. Kansa. Multiquadrics - a scattered data approximation scheme with applications to computational fluid-dynamics - II. *Computers and Mathematics with Applications*, 19(8-9):147-161, 1990.

[55] E. Larsson, U. Pettersson, J. Presson, and G. Marcusson. Improved radial basis function methods for multi-dimensional option pricing. *Computers and Mathematics with Applications*, 222(1):82-93, 2008.

[56] J. Levesley and Z. Luo. Error estimates and convergence rates for variational hermite interpolation. *Journal of Approximation Theory*, 95:264-279, 1998.

[57] S. P. Lloyd. Least Squares Quantization in PCM *IEEE Transactions on Information Theory*, 28:129-137, 1982.

[58] W. A. Light and H. Wayne. Some remarks on power functions and error estimates for radial basis function approximation. *Journal of Approximation Theory*, 92:245-266, 1998.

[59] M. W. Lu, Y. C. Hon, X. Z. Mao, and Y. M Zhu. Multiquadric method for the numerical solution of a biphasic mixture model. *Applied Mathematics Computation*, 88(2-3):153–176, 1997.

[60] L. T. Luh On Wu and Schaback's error bound. *International Journal of Numerical Methods and Applications*, 1(2):155-174, 2009.

[61] L. T. Luh The shape parameter in the Gaussian function. *Computers and Mathematics with Applications*, 63(3):687-694, 2012.

[62] L. T. Luh The shape parameter in the Gaussian function II. *Engineering Analysis with Boundary Elements*, 37(6):988-993, 2013.

[63] W. R. Madych and S. A. Nelson. Multivariate interpolation and conditionally positive definite functions. *Approximation Theory and its Applications*, 4:77-89, 1988.

[64] W. R. Madych and S. A. Nelson. Multivariate interpolation and conditionally positive definite functions. II. *Mathematics of Computation*, 54(189):211-230, 1990.

[65] W. R. Madych and S. A. Nelson. Bounds on multivariate polynomials and exponential error estimates for multiquadric interpolation. *Journal of Approximation Theory*, 70(1):94-114, 1992.

[66] S. D. Marchi. On optimal center locations for radial basis function interpolation: computational aspects. *Rend. Splines Radial Basis Functions and Applications*, 61(3):343-358, 2003.

[67] S. D. Marchi, R. Schaback, and H. Wendland. Near-optimal data-independent point locations for radial basis function interpolation. *Advances in Computational Mathematics*, 23:317-330, 2005.

[68] C. D. Meyer Matrix analysis and applied linear algebra. *SIAM*, Philadelphia, 2000.

[69] C. A. Micchelli. Interpolation of scattered data: distance matrix and conditionally positive definite function. *Constructive Approximation*, 2:11-22, 1986.

[70] F. J. Narcowich, R. Schaback, and J. D. Ward. Multilevel interpolation and approximation. *Applied and Computational Harmonic Analysis*, 7:243-261, 1999.

[71] S. L Naqvi. *Adaptive radial basis function interpolation for time-dependent partial differential equations.* PhD thesis, University of Leicester, June 2013.

[72] Y. Ohtake, A. Belyaev, and H. P. Seidel. 3D scattered data interpolation and approximation with multilevel compactly supported RBFs. *Graphical Models*, 67:150-165, 2005.

[73] A. Pena. Option pricing with radial basis functions: a tutorial. Technical report, Wilmott Magazine, 2005.

[74] A. Perdichizzi and V. Dossena. Incidence angle and pitch-chord effects on secondary flows downstream of a turbine cascade. *Journal of Turbomachinery*, 115:383-391, 1993.

[75] R. B. Platte. How fast do radial basis function interpolants of analytic functions converge. *SIMA Journal of Numerical Analysis*, 31(4), 2011.

[76] M. J. D. Powell. Radial basis functions for multi-variable interpolation: a review, numerical analysis. *Longman Scientific and Technical*, 223–241, 1987.

[77] M. J. D. Powell. The uniform convergence of thin plate spline interpolation in two dimensions. *Numerische Mathematik*, 68(1):107-128, 1994.

[78] S. Rippa. An algorithm for selecting a good value for the parameter c in radial basis function interpolation. *Advances in Computational Mathematics*, 11:193-210, 1999.

[79] S. A. Sarra and E. J. Kansa. Multiquadric Radial Basis Function Approximation Methods for the Numerical Solution of Partial Differential Equations. *Tech Science Press*, 2009.

[80] R. Schaback. Error estimates and condition numbers for radial basis function interpolation. *Advances in Computational Mathematics*, 3:251–264, 1995.

[81] R. Schaback and Z. Wu. Local error estimates for radial basis function interpolation of scattered data. *SIMA Journal of Numerical Analysis*, 13:13–27, 1993.

[82] J. Stewart. Positive definite functions and generalizations, an historical survey. *Rocky Mountain Journal of Mathematics*, 6(3): 409-434, 1976.

[83] L. Sydow, L. J. Hook, E. Larsson, E. Lindstrom, S. Milovanovic, J. Persson, V. Shcherbakov, Y. Shpolyanskiy, S. Siren, J. Toivanen, J. Walden, M. Wiktorsson, J. Levesley, J. Li, C. W. Oosterlee, M. J. Ruijter, A. Toropov, and Y. Zhao. BENCHOP - The BENCHmarking project in Option Pricing. *International Journal of Computer Mathematics*, 92:2361-2379, 2015.

[84] H. Wendland. Meshless galerkin approximation using radial basis functions. *Mathematics of Computation*, 68(228):1521–1531, 1999.

[85] H. Wendland. *Scattered data approximation.* Cambridge University Press, Cambridge, 2005.

[86] Z. Wu. Solving PDE with radial basis function and the error estimation. *Advances in computational mathematics. Lecture notes on pure and applied mathematics*, 202, 1998.

[87] Z. Wu. Dynamically knots setting in meshless method for solving time dependent propagations equation. *Computer Methods in Applied Mechanics and Engineering*, 193:1221-1229, 2004.