PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
MOHAMED EL BACHIR EL IBRAHIMI UNIVERSITY, BORDJ BOU ARRERIDJ

Faculty: Mathematics and Informatics

Department: Mathematics

THESIS

In order to obtain

Doctorate degree in science

SPECIALTY: Mathematics
OPTION: Applied Mathematics
Presented by: TAKOUK Dalila

Theme:

Compactly supported radial basis functions

Board of jury:

Azedine RAHMOUNE	Prof.	University of Bordj Bou Arreridj	President
Belkacem LAKEHALI	M.C.A.	University of M'sila	Supervisor
Rebiha ZEGHDANE	M.C.A.	University of Bordj Bou Arreridj	Co-supervisor
Abdelbaki MEROUANI	Prof.	University of Setif 1	Examiner
Bachir GAGUI	M.C.A.	University of M'sila	Examiner
Rebiha BENTERKI	M.C.A.	University of Bordj Bou Arreridj	Examiner

Academic Year

2022/2023

Acknowledgement

First and foremost, praises and thanks to Allah, the Almighty, for letting me through all the difficulties to complete the research successfully. Prayers and peace be upon our Prophet, Muhammad, his family and all of his companions.

Firstly, I would like to express my deep and sincere gratitude to my thesis advisors Dr. Belkacem LAKEHALI from Mathematics Department, University Mohamed Boudiaf of M'sila, for giving me the opportunity to do research and providing invaluable guidance through this research, and Dr. Rebiha ZEGHDANE from Mathematics Department, University Mohamed El Bachir El Ibrahimi of Bordj Bou Arreridj, El Anasser, for the continuous support and motivation. She has taught me the methodology to carry out the research and to present the research work as clearly as possible. Their wisdom, guidance, enthusiasm and immense knowledge about my PhD thesis and related research, helped me in all the time of research and writing of this thesis.

Secondly, I would like to thank the members of my thesis committee: Pr. Azedine RAHMOUNE from Mathematics Department, University Mohamed El Bachir El Ibrahimi of Bordj Bou Arreridj, El Anasser, Pr. Abdelbaki MEROUANI from Mathematics Department, University Ferhat Abbas of Setif 1, Dr. Rebiha BENTERKI from Mathematics Department, University Mohamed El Bachir El Ibrahimi of Bordj Bou Arreridj, El Anasser, Dr. Bachir GAGUI from Mathematics Department, University Mohamed Boudiaf of M'sila, for their insightful comments, and questions which incented me to widen my research from various perspectives.

Finaly, I am extremly grateful to all my family for their encouragement, love, prayers and sacrifies which helped me in completion of this thesis. My thanks and appreciation also go to my colleages and all members of Mathematics Department, University Mohamed El Bachir El Ibrahimi of Bordj Bou Arreridj.

Dedications

♣ I dedicate my dissertation work and my feeling of gratitude, love and respect to my loving parents and all my family who have been my source of inspiration, encouragement and strength, throughout the process. I will always appreciate all they have done.

♣ I also dedicate this work and give special thanks to my beloved sister salima, and my husband whose continually provide thier support, wisdom and guidance.

ABSTRACT

This thesis deals with the applications of compactly supported radial basis functions for high dimensional reconstruction of surfaces (images) based on irregular samples. These methods without mesh (meshfree) based on the introduction of radial basis functions, contrary to traditional methods, namely finite element (FEM) and finite difference (FDM) methods. We try to introduce the concept of this technique through several applications.

Keywords:

Radial Basis Functions (RBFs), multivariate interpolation, scattered data, numerical solution.

RÉSUMÉ

Cette thèse traite les applications des fonctions de base radiale, à support compact (CSRBF), pour la reconstruction bidimensionnelle de surfaces (images) à partir d'échantillons irréguliers. Ces méthodes sans maillage (meshefree) qui reposent sur l'introduction des fonctions de base radiale, contrairement, aux méthodes traditionnelles, a savoir la méthode des éléments finis (FEM) et la méthode des différences finies (FD). Nous essayons d'introduire le concept de cette méthode à travers plusieurs applications.

Mots clés:

Fonctions de base radiale (RBF), interpolation multivariée, données dispersées, solution numérique.

ملخص:

تتناول هذه الأطروحة تطبيقات وظائف الأساس الشعاعي المدعومة بشكل مضغوط لإعادة بناء الأسطح ذات الأبعاد العالية استنادا إلى عينات غير منتظمة. هذه الطرق بدون شبكة تستند إلى إدخال وظائف الأساس الشعاعي، على عكس الطرق التقليدية، كطريقة الفروق المحدودة وطريقة العناصر المحدودة. نحاول تقديم مفهوم هذه الطريقة من خلال عدة تطبيقات.

كلمات مفتاحية : وظائف الأساس الشعاعية، استيفاء متعدد المتغيرات، بيانات مبعثرة، حل عددي.

Contents

List of Figures

List of Tables

List of Symbols

$\mathbb{R}$:	Set of real numbers.
$\mathbb{R}^s$:	Euclidean space of dimension s
$\mathbb{C}$:	Set of complex numbers
$\mathbb{K}$:	$\mathbb{R}$ or $\mathbb{C}$
$\mathbb{N}$:	Set of natural numbers
$\mathbb{N}_0^d$:	Set of n-dimensional numbers, each of whose entries is a natural number including zero
$C(\Omega)$:	Space of continuous functions on Ω
$C(\Omega \times \Omega)$:	Space of continuous functions on Ω in Ω
$C([0,1])$:	Space of continuous functions on $[0,1]$
$C^k(\Omega)$:	Space of k-times continuously differentiable functions on Ω
$R_N[x]$:	Set of polynomials with coefficients in $\mathbb{R}$ and of degree $\leqslant N$
ϵ:	Shape parameter
$\lceil k \rceil$:	The nearest integers less or equal to k
$\mathcal{N}_\Phi(\Omega)$:	Native space
$\mathcal{H}^s_{\mathbb{R}^s}$:	Sobolev space of order s on $\mathbb{R}^s$
$L^1(\Omega)$:	Lebsegue space of integrable functions on Ω
$L^\infty(\Omega)$:	Lebsegue space of bounded measurable functions on ω
$[a,b]$:	Closed interval of $\mathbb{R}$ with extremities a and b
$]a,b[$:	Open interval of $\mathbb{R}$ with extremities a and b
$[0,\infty)$:	Closed interval containing all real numbers from 0 to ∞ including zero
$(-\infty,0]$:	Closed interval containing all real numbers from $-\infty$ to 0 including zero
$(0,\infty)$:	Open interval containing all real numbers greater than zero
$\pi_m(\mathbb{R}^s)$:	Space of polynomials in s- variables of degree at most m
$\mathbb{P}^d_q$:	Space of d-variate polynomials of order not exceeding than q
P_N:	Orthogonal projection operator.

Introduction

The subject of this thesis concern the applications of radial basis functions for solving linear and nonlinear functional equations in particulary Volterra-Fredholm integral equations in two dimensional space. The proposed algorithm is based on pseudo spectral method using compactly and multiquadric generalized radial basis functions.

Scattered data approximation is a recent fast growing research area [36, 49, 51], it deals with the problem of reconstructing an unknown function from given scattered data. This field has many applications such as, fluid structure interaction, terrain modeling, computer science, different fields as applied mathematics, biology, geology ...etc.

The polynomial interpolation is a powerful tool to approximate given data sites in the univariate setting, since a set of distinct points can be interpolated using a unique polynomial. In higher-dimensional problems, it is not always possible to obtain a unique polynomial interpolation for multivariate data sites justified by the Mairhuber-Curtis theorem [57].

Traditional numerical methods, such as finite difference , finite elements or finite volume methods were motivated mostly by early one or two dimensional simulation of engineering problems via partial differential equations. The discretization of these methods require some sort of underlying computation mesh, which becomes a rather difficult task in high dimensional. To overcome this problem, we must establish a basis which depends on the data locations selected arbitrarily on certain domains. Therefore we can approximate a function without mesh generation on the domain utilizing these basis functions which called radial basis functions (RBFs). Radial basis function methods belong to a category of methods called meshless (meshfree) methods [11, 19, 29], which does not require connectivity of grid/mesh points. This is achieved by composing a univariate basic function with a norm usually Euclidean norm, that makes the problem insensitive to the dimension and makes it virtually one-dimensional. meshless methods have received much attention not only by applied mathematics but also in different fields of science.

RBFs are effective techniques for interpolating an unknown function on a scattered set of points which have been used in the past few decades. These functions involve a single independent variable regardless of the dimension of the problem, so applying them in higher dimensions does not increase the difficulties. Also it should be noted that the RBF approach does not require any domain elements, so it does not depend on the geometry of a domain. Firstly, Hardy [36] has studied RBFs as a multidimensional scattered interpolation method in modeling of the Earth's gravitational field in 1971, by using multiquadrics (MQs), inverse multiquadrics (IMQs) and thin plate splines (TPSs) as a type of free shape parameter RBF. Radial basis functions have been developed by Meinguet [62] and have been investigated for smoothing noisy multidimensional data by Wahba [82]. Franke [34] has published a review paper on the comparison of two-dimensional interpolation methods available in the early 1980. In recent years, the implication of RBFs has been shifted from scattered data interpolation to the numerical solution of partial differential equations (PDEs). A method for the numerical solution of PDEs which utilizes radial basis functions, specially the MQ, as a basis in the collocation method is called Kansa's method created by Kansa in 1990 [49, 52]. Kansa's method has been developed for solving various types of partial differential equations such as the one-dimensional nonlinear Burgers equation with shock wave [39], shallow water equations for tide and currents simulation [40], heat transfer problems [77, 97], parabolic equation with nonlocal boundary conditions [78], financial mathematic problems [41], Klein-Gordon equation [25], and improved Boussinesq equation [75]. Later, Fasshauer has modified Kansa's method to a Hermite-type collocation method for the solvability of the resultant collocation matrix [28].

Radial basis functions are truly meshless and simple to allow modeling of rather high dimensional problems [18, 48, 74]. These basis functions can be clustered in a specific region to locally increase the accuracy of the method. It was shown that RBF converges to pseudo spectral methods in their flat radial function limit.

Radial basis functions open the door to existence and uniqueness results for interpolating scattered data by radial basis functions in very general settings. Indeed, one of the greatest advantages of this method lies in its applicability in almost any dimension because there are generaly little restrictions on the way the data are prescribed. A further advantage is their high accuracy or fast convergence to the approximated target function in many cases when data become dense.

Radial basis functions have a parameter in their definitions which called the shape parameter, this parameter controls the shape of RBF and the conditioning of linear systems to be solved. Locating an optimal shape parameter is a difficult problem and a topic of current research [17, 69, 76, 81].

The convergence of RBF approaches can be seen in terms of two different types of approximation, stationary and non-stationary. In stationary approximation, the number of centers N is fixed and the shape parameter is refined towards zero. This type is unique to RBF methods which does not exist in polynomial based methods. For non-stationary technique, we fix the value of the shape parameter and N increased.

Radial basis functions methods which rely on global interpolation functions, result large fully-populated matrices and ill-conditioned systems if very smooth radial basis functions are used on large number of points, that can produce poor and unstability in the solution, also proper selection of the shape parameter often involves optimization. To overcome this problem, researchers have developed a scheme based on radial basis functions with compact support (CSRBFs). Using compactly supported radial basis functions produces a sparse interpolation matrix, also the operation of the banded matrix system could reduce the ill-conditioning of the resultant coefficient matrix. The compactly supported basis functions consist of a polynomial which are non-zero on $[0, 1)$ and vanish on $[1, \infty)$. A family of CSRBFs was first introduced by Wu [89] and later expanded by Wendland [84] in the mid 1990s.

In this work, new computational methods based on both globally and compactly supported radial basis functions (CSRBFs) are presented for solving nonlinear integral equations and partial differential equations. These equations reduced to systems of algebraic equations which can be solved via iteration method. Some error estimations are provided and illustrative examples are also included to demonstrate the efficiency and applicability of the proposed methods.

This work is organized in the following way:
The first chapter, is devoted to some elementary concepts of interpolation problems. Also, to

give an introduction to radial basis functions and principle about RBFs inerpolation problems. Chapter 2 gives introduction to CSRBFs with convergence analysis, also it develops some numerical methods for solving different types of equations using RBFs and CSRBFs, such as Burger, Poisson and Schrodinger partial differential equations, also linear and nonlinear Volterra and Fredholm integral equations. The comparaison between stationary and non-stationary approaches is discussed. Also new method based on composite techniques for solving Volterra-Fredholm integral equations is presented.

In chapter 3, we introduce new techniques for finding the approximate solutions of two dimensional nonlinear Volterra-Fredholm integral equations based on generalized global RBFs and CSRBFs, also convergence analysis and error estimates are provided. These techniques can be implemented for solving high dimensional integral equations.

Finally, we summarise the work that has been carried out in this thesis and consider some perspectives of future work.

Chapter 1

Concept of interpolation problem

In chapter 1, some basic results of interpolation problem and an introduction to radial basis functions interpolation problem are given.

Most functions encountred in mathematics can not be evaluated exactly, even though we usually handle them as if they were completely known quantities. The interpolation is the process of finding and evaluating a function whose graph goes through a set of given points, these points may arise as measurements in physical problem, or they may be obtained from a known function.

In many scientific disciplines one faces the following problem. We have a set of data (for example: measurements, and locations at which these measurements were obtained), and we want to find a rule which allows us to deduce information about the process we are studying also at locations different from those at which we obtained our measurements. Thus, we are trying to find a function which is a "good" fit to the given data. There are many ways to decide what we mean by "good", and the only criterion we will consider now is that we want the functions to exactly match the given measurements at the corresponding locations. This approach is called interpolation, and if the locations at which the measurements are taken are not on a uniform or regular grid, then the process is called scattered data interpolation.

1.1 Polynomial interpolation in one dimensional

Given a set of data points $(x_i, y_i) \in X \times Y$, $i = 0 : N$, X, $Y \subset \mathbb{R}$ (or $\mathbb{C}$) are the domains x_i and y_i reside, respectively, and the points x_i are called the interpolation nodes and assumed distinct. Provided with a specific linear subspace V of functions in $C(X)$. Find an interpolating function f in V satisfying the interpolating condition

$$f(x_i) = y_i, \; i = 0, \ldots N.$$

An interpolation function is also called interpolant. The primary purpose of interpolation is to replace a set of data points (x_i, y_i) with a function given analytically, another purpose is to approximate functions with simpler ones, usually polynomials or piecewise polynomials.

We start with the simplest case, when only the values $f_i := f(x_i)$, for $i = 0, \ldots, N$ are given at the pairwise distinct nodes $x_0, \ldots, x_N$. We now seek a unique polynomial $P \in \mathbf{P}_N = R_N[x]$ (a set of polynomials with coefficients in $\mathbb{R}$ and of degree $\leqslant N$),

which interpolates f at the $(N + 1)$ nodes $x_0, \ldots, x_N$, i.e., $P(x_i) = f_i$ for $i = 0, \ldots, N$.

In order actually to compute the interpolating polynomial, we have to choose a basis of the space of polynomials $\mathbf{P}_N$.

There are several basis in which one can represent a polynomial

1- Monomial basis: the monomial basis of a polynomial is of the forme $\{1, x, ..., x^N\}$.

2- Center basis: the center basis of a polynomial is of the form $\{1, (x-c), ..., (x-c)^N\}$ with $c \neq 0$.

3- Lagrange basis: let $(x_i)_{i=0}^N$, be N distinct points, the associated basis is given by

$$L_i(x) = \prod_{j=0, i\neq j}^{N} \frac{x - x_j}{x_i - x_j}.$$

4- Newton basis: let $(x_i)_{i=0}^N$, be N distinct points, the associated basis is given by

$$\{1, (x-x_0), (x-x_0)(x-x_1), .., \prod_{k=0}^{N}(x-x_k)\}.$$

If we write P as above in coefficient representation

$$P_N(x) = \sum_{i=0}^{N} a_i x^i,$$

i.e., with respect to the monomial basis $\{1, x, ..., x^N\}$ of $\mathbf{P}_N$, then the interpolation matrix resulting from interpolation conditions $P_N(x_i) = f(x_i)$ is called Vandermonde matrix. The determinant of Vandermonde matrix is different from zero exactly when the nodes $x_0, \ldots, x_N$ are pairwise distinct. However, the solution of the system requires an excessive amount of computational effort. In addition, the Vandermonde matrices are almost singular in higher dimensions N.

1.1.1 Polynomial interpolation error and Runge's phenomenon

Theorem 1.1. *Let f be a function in $C^{N+1}[a, b]$, and let P_N be a polynomial of degree $\leq N$ that interpolates the function f at $(N+1)$ distinct points $x_0, x_1, \ldots, x_N \in [a, b]$. Then to each $x \in [a, b]$ there exists a point $\xi_x \in [a, b]$ such that*

$$f(x) - P_N(x) = \frac{1}{(N+1)!} f^{(N+1)}(\xi_x) \prod_{i=0}^{N}(x - x_i). \tag{1.1}$$

The Runge phenomenon

Runge phenomenon is a problem of oscillation at the edges of an interval that occurs when using polynomial of high degree over a set of equispaced interpolation points. The discovery was important because it shows that going to higher degree doesn't always improve accuracy.

Interpolation at equidistant points is a natural and well-known approach to construct approximating polynomials. Runge's phenomenon demonstrates, that interpolation can result a divergent approximations.

Let Consider the function

$$f(x) = \frac{1}{1+x^2}. \tag{1.2}$$

Runge found that if this function is interpolated at equidistant points $x_i = -5 : \frac{10}{N} : 5$ between -5 and 5. The resulting interpolation oscillates towards the end of the interval. It can be proven that the interpolation error increases when the degree of the polynomial is increased.

Example 1.1. The comparaison between $f(x)$ and its interpolation polynomial is represented in figure 1.1-1.2.

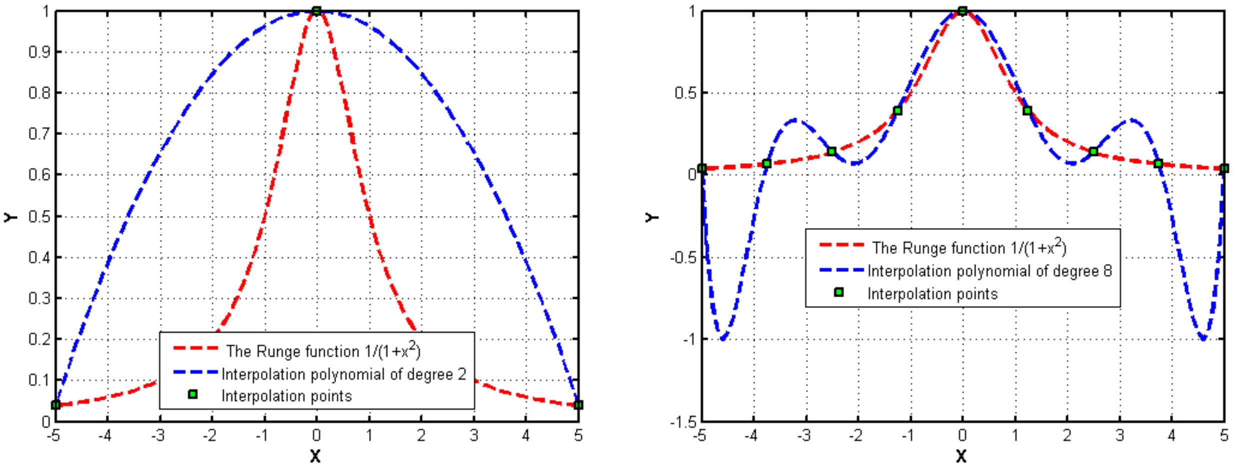

Figure 1.1: Comparaison between Runge's function and its Lagrange interpolation polynomial of second degree (left), eighth degree (right).

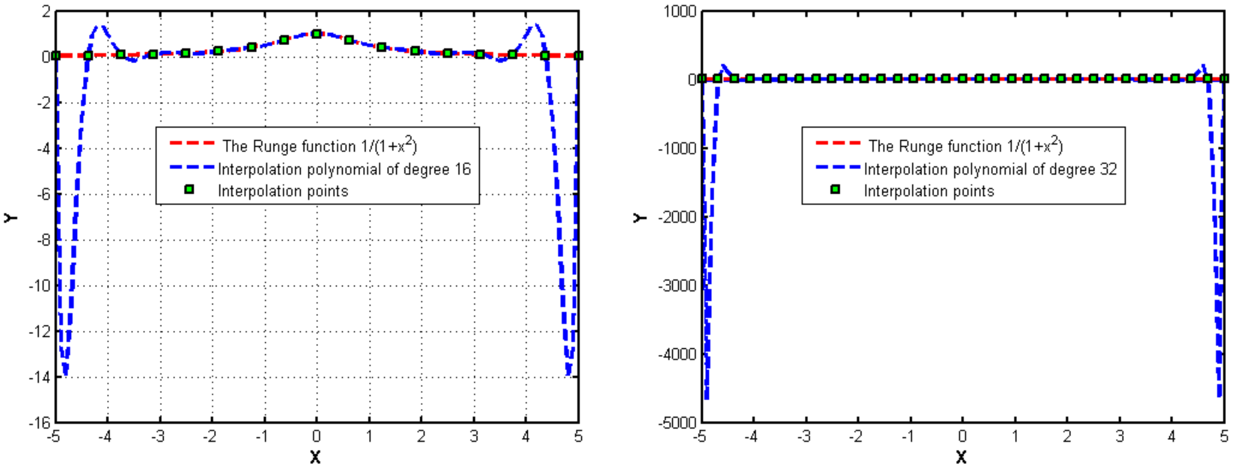

Figure 1.2: Comparaison between Runge's function and its Lagrange interpolation polynomial of twelfth degree (left), sixteenth degree (right).

If we consider the class of functions

$$\mathcal{F} = \left\{ f \in C^{N+1}[a,b], \mid \sup_{\tau\in[a,b]} f^{(N+1)}(\tau) \mid \leq M(N+1)! \right\},$$

for a constant $M > 0$, then the approximation error obviously depends crucially on the choice of the nodes $x_0, \dots, x_N$ via the expression

$$\omega_{N+1}(x) = (x - x_0)\dots(x - x_N). \tag{1.3}$$

Also the equidistance between points leads to Lebesgue constant that increases quickly when N increases. The table 1.1 shows the the error between Runge's function and its interpolation polynomial when N increasing, with $x_{N-1/2} = 5 - \frac{5}{N}$. We remark that

$$\lim_{N\to\infty} \parallel P_N(x_{N-1/2}) - f(x_{N-1/2}) \parallel_\infty = +\infty.$$

The evaluation of $prod(x) = \prod_{i=0}^{N}(x - x_i)$ is illustrated in table 1.2, with: $x_i = -5 : \frac{10}{N} : 5$.

N	$f(x)$	$p_N(x)$	Absolute error
2	0.1379	0.7596	0.6217
10	0.0471	1.5787	1.5317
16	0.0435	-10.1739	10.2174
20	0.0424	-39.9524	39.9949

Table 1.1: The error between Runge's function and its interpolation polynomial when $N \to \infty$.

x	$f(x)$	$p_{20}(x)$	Absolute error	$prod(x)$
0.25	0.9412	0.9425	0.0013	$2.0468e + 006$
1.75	0.2462	0.2384	0.0077	$-6.5587e + 006$
3.75	0.0664	-0.4471	0.5134	$-7.5594e + 008$
4.75	0.0424	-39.9524	39.9949	$-7.2721e + 010$

Table 1.2: The behavior of $prod(x)$.

Runge's phenonmenon can be avoided by

1- **Change of the interpolation points**

The oscillation can be minimized by using nodes that are disturbed more densely towards the edges of the interval, this set of nodes is Chebyshev nodes for which the maximum error in approximating the Runge function is guaranted to diminish with increasing polynomial order.

2- **Use piecewise polynomials**

The problem can be avoided by using spline curves which are piecewise polynomials, when trying to decrease the interpolation error one can increase the number of polynomials pieces which are used to construct the spline instead of increasing the degree of the polynomial.

1.2 Problem of interpolation in higher dimensions

In the univariate setting it is well known that one can interpolate to arbitrary data at $(N + 1)$ distinct data sites using a polynomial of degree N. For the multivariate setting, however, through the work of Alfréd Haar and his description of Haar spaces in 1956 [57], we obtain the negative result that well-posedness is not guaranteed in higher dimensional linear systems with independently chosen basis functions.

Figure 1.3: Alfréd Haar, John Mairhuber, and Phillip Curtis

Theorem 1.2. ***Haar-Mairhuber-Curtis*** *[57].*

Let $\Omega \subset \mathbb{R}^s$, $s \geq 2$, contains an interior point, then there exist no Haar spaces of continuous functions except for one-dimensional ones.

In order to understand this theorem we need

Definition 1.1. [57] Let the linear finite-dimensional function space $\mathcal{B} \subseteq C(\Omega)$ have a basis $\{B_1, \ldots, B_N\}$. Then $\mathcal{B}$ is a Haar space on Ω if

$$det(B_k(x_j)) \neq 0, \ \ k, j = 1, \ldots, N. \tag{1.4}$$

for any set of distinct $x_1, \ldots, x_N$ in Ω.

Proof. To prove Haar-Mairhuber-Curtis theorem , let $s \geq 2$ and suppose $\mathcal{B}$ is a Haar space with basis $\{B_1, \ldots, B_N\}$ with $N \geq 2$. Then, by the definition of a Haar space

$$det(B_k(x_j)) \neq 0, \tag{1.5}$$

for any distinct $x_1, \ldots, x_N$.

Now consider a closed path P in Ω connecting only x_1 and x_2. This is possible since by assumption Ω contains an interior point. We can exchange the positions of x_1 and x_2 by moving them continuously along the path P (without interfering with any of the other x_j). This means, however, that rows 1 and 2 of the determinant (1.5) have been exchanged, and so the determinant has changed sign. Since the determinant is a continuous function of x_1 and x_2 we must have determinant equal to zero at some point along P. This is a contradiction. □

Remark 1.1. 1- Note that existence of a Haar space guarantees invertibility of the interpolation matrix $(B_k(x_j))$, i.e., existence and uniqueness of an interpolant to data specified at $x_1, \ldots, x_N$, from the space $\mathcal{B}$.

2- As mentioned above, univariate polynomials of degree $N-1$ form an N-dimensional Haar space for data given at $x_1, \ldots, x_N$.

3- The Haar-Mairhuber-Curtis theorem implies that in the multivariate setting we can no longer expect this to be the case. It is not possible to perform unique interpolation with (multivariate) polynomials of degree N to data given at arbitrary locations in $\mathbb{R}^2$.

So as a result of this theorem, if we choose our basis functions independently of the data, we are not guaranteed a well-posed problem.
The Haar-Mairhuber-Curtis theorem tells us that if we want to have a well-posed multivariate scattered data interpolation problem, we can't fix in advance the set of basis functions, but the basis should depend on the data location.

1.2.1 Meshfree methods

Originally, the motivation for the basic meshfree approximation methods (radial basis functions) came from applications in geodesy, geophysics, mapping, or meteorology. Later, applications were found in many areas such as in the numerical solution of PDEs, artificial intelligence, learning theory, neural networks, signal processing, statistics (kriging), finance, and optimization. It should be pointed out that meshfree local regression methods have been used (independently) in statistics for more than 100 years . "Standard" multivariate approximation methods (splines or finite elements) require an underlying mesh (e.g. triangulation) for the definition of basis functions or elements. This is very difficult in space dimensions greater than two.

Some historical landmarks for meshfree methods in approximation theory

- D. Shepard, Shepard functions, late 1960s (application, surface modelling)
- Rolland Hardy (Iowa State Univ.), multiquadrics (MQs), early 1970s (application, geodesy)
- Jean Meinguet (Université Catholique de Louvain, Louvain, Belgium), surface splines, late 1970s (mathematics)
- Richard Franke (NPG, Montery), in 1982 compared scattered data interpolation methods, and concluded MQs and TPs were best. Franke conjectured interpolation matrix for MQs is invertible.
- Charles Micchelli (IBM), Interpolation of scattered data: Distance matrices and conditionally positive definite functions, 1986.

Advantages of meshfree methods

Meshfree methods have gained much attention in recent years, this is due to the following reasons:

- Many traditional numerical methods (finite differences, finite elements or finite volumes) have trouble with high-dimensional problems.

- Meshfree methods can often handle changes in the geometry of the domain of interest (e.g., free surfaces, moving particles and large deformations) better.
- Independence from a mesh is a great advantage since mesh generation is one of the most time consuming parts of any mesh-based numerical simulation.
- New generation of numerical tools.

Applications

- Original applications were in geodesy, geophysics, mapping, or meteorology.
- Later, many other application areas
  - Numerical solution of PDEs in many engineering applications,
  - Computer graphics,
  - Sampling theory,
  - Artificial intelligence,
  - Machine learning or statistical learning (neural networks or SVMs),
  - Signal and image processing,
  - Statistics (kriging),
  - Finance,
  - Optimization.

1.2.2 Basis functions depending on data

* The basis functions of meshless methods noted by ϕ_i are dependent on the data sites x_i as suggested by Haar-Mairhuber-Curtis.

* The points x_i for which the basic function is shifted to form the basis functions, are usually referred as centers or knots.

* Technically, one could choose these centers different from the data sites. However, usually centers coincide with the data sites. This simplifies the analysis of the method, and is sufficient for many applications. In fact, relatively little is known about the case when centers and data sites differ.

* $\phi_i(x)$ are radially symmetric about their centers, for this reason we call these functions Radial Basis Functions (RBFs).

In 1968 , R.L. Hardy [37] wanted to create a satisfactory function that could represent a topographical curve. While studying this problem, Hardy discovered that, the data could be satisfactory represented by a piecewise linear interpolating function [37]. He proposed that given a set of N distinct scattered data points $\{x_j\}_{j=0}^{N}$ and corresponding measurements $\{f_j\}_{j=0}^{N}$, that the form of

$$\phi_j(x) = |x - x_j|, \ j = 0, \dots, N.$$

Hardy soon recognized that the absolute function had a jump in the first derivative at each source point. Hardy figured out that this problem could be solved by removing the absolute value basis function and replacing it with a function that is continuously differentiable. Hardy's function was $\sqrt{\epsilon^2 + r^2}$, where ϵ is an arbitrary non-zero constant [37]. Hardy applied the interpolation method using this function to more than a one dimensional space. Note that the absolute value of the difference between two points in two dimensional space is the Euclidean distance between the two points; for example, $\mid x - x_j \mid = \sqrt{(x - x_j)^2}$. What Hardy created was an interpolating function based on translates of the Euclidean distance function in two dimensions. Hence, given N distinct scattered data points $\{x_j, y_j\}_{j=0}^{N}$ and corresponding topographic measurements $\{f_j\}_{j=0}^{N}$ for $j = 0, 1, \ldots, N$, Hardy proposed the following basis function

$$\phi_{i,j}(x, y) = \sqrt{(x - x_j)^2 + (y - y_j)^2}. \tag{1.6}$$

To be exact $\phi(r) = \sqrt{x^2 + y^2}$. Also as previously described in one dimension, the vertex of each cone is centered at one of the data points. Again, Hardy ran into the same problem as his one dimensional interpolating function. The problem was that function defined in equation (1.6) suffered from being a piecewise continuous. He was unable to find a simple fix for this problem. Hardy proposed using a linear combination circular hyperboloid basis functions (rotated hyperbola basis functions $\sqrt{\epsilon^2 + x^2}$ translated to be centered at each source point). The new form of equation (1.6) is

$$\phi_{i,j}(x, y) = \sqrt{\epsilon^2 + (x - x_j)^2 + (y - y_j)^2}. \tag{1.7}$$

Hardy discovered that the interpolation method based on the new function was an excellent method for approximating topographical information from sparse data points. Unlike the Fourier series, the new function did not suffer from large oscillations. Also, the function alleviated the problem associated with the polynomial series method (i.e. the polynomial series was unable to account for rapid variations of the topographical surface) [36]. Hardy named this new technique the multiquadric basis function (MQ).

Notice that the multiquadric basis function is also radially symmetric about its center. Because of this radial symmetry, the multiquadric kernel can be described as a Radial Basis Function. In other words, it is a basis function which depends only on the radial distance from its center. Since our basis functions depend only on distance.

Definition 1.2. RBF approximations are usually finite linear combinations of the translation of a radially symmetric basis function. The set of RBFs, ϕ_i is as follows
ϕ_i: $\mathbb{R}^d \longrightarrow \mathbb{R}$, $\phi_i(x) = \phi(\parallel x - x_i \parallel)$, where $\parallel . \parallel$ denote the Euclidean norm and x_i is the center of RBF.

1.3 Radial basis functions interpolation problem

This method was proposed by Edward Kansa in 1990 [49], a proffessor at the university of California. It was used for the first time for polynomial interpolation problems. The method makes it possible to offer a high order accuracy with nodes dispresed on a totally irregular geometry with a particular simple algorithm compared to the classical methods used until this moment. Before Kansa's successful research, Hardy [36, 37] used the multiquadric function to interpolate multidimensional data and reconsile two dimensional geographic surfaces, showed that multiquadric has a physical foundation as a consistent solution to the biharmonic potential problem. Buhmann and Michelli [14] have shown that the MQ interpolation sheme converges faster as the spatial dimension increases, and converges exponentially as the density of the nodes increases. Buhmann and Michelli [15] and Chui et al [23] have shown that MQ and other RBFs were prewavelets. Kansa intervened to solve partial differentiall equations of elliptic, parabolic or even hyperbolic type [51]. This intervention which modified the multiquadric function was very succesful. Finaly, in 1990 Hon et al [42] have improved the MQ method for solving varieties of nonlinear boundary problems, the most common of which is the Burgers equation.

The scattered data approximation problem is as follows, let given a set of N distinct data points $X = \Big(x_1, x_2, ..., x_N\Big)$ in $\mathbb{R}^N$ and a corresponding set of N values $\Big(y_1, y_2, ..., y_N\Big)$ sampled from an unknown function f such that $y_i = f(x_i)$. We can then choose a radial function ϕ and a set of centers $\Big(x_{c1}, x_{c2}, ..., x_{cN}\Big)$ for some $N \in \mathbb{N}$, to obtain a basis $\Big(\phi(||. - x_{c1}||), \phi(||. - x_{c2}||), ..., \phi(||. - x_{cN}||)\Big)$.
This basis can then be used to construct an approximation s of the function f.
One option is to center an RBF on each data site. In that case, the approximation will be constructed from n radial basis functions, and there will be one basis function with $x_c = x_i$ for each $i = 1, 2, ..., N$.

The approximation s is then constructed from a linear combination of those N RBFs, such that

$$s(x) = \sum_{i=1}^{N} c_i \phi(||x - x_i||), \tag{1.8}$$

with: $r_i = ||x - x_i||$. Then,

$$\begin{bmatrix} \phi_1(r_1) & \phi_1(r_2) & \cdots & \phi_1(r_N) \\ \phi_2(r_1) & \phi_2(r_2) & \cdots & \phi_2(r_N) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_N(r_1) & \phi_N(r_2) & \cdots & \phi_N(r_N) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix}.$$

Such that, $c = [c_1, c_2, ..., c_N]$ and $y = [f_1, f_2, ..., f_N]$,

The constants c_i are determined by ensuring that the approximation will exactly match the given data at the data points. This is accomplished by enforcing $s(x_i) = y_i = f_i$, $i = 1, \ldots N$, which produces the system of linear equations

$$Ac = y. \tag{1.9}$$

The solution of the system requires that the matrix A is non-singular. The situation is favorable if we know in advance that the matrix is positive definite. Moroever we would like to characterize the class of functions ϕ for which the matrix is positive definite.

Polynomial Terms

It is sometimes useful to add low order polynomials to our method of radial basis function interpolation. We let π^s_{m-1} be the linear space of polynomials from $\mathbb{R}^s$ to $\mathbb{R}$ of degree at most $m - 1$, and choose p_j, $j = 1, 2, \ldots, M$ as a basis for this space, whose dimension is

$$M = \begin{bmatrix} m - 1 + s \\ m - 1 \end{bmatrix}.$$

This means we let $s(x)$ have the form

$$s(x) = \sum_{i=1}^{N} c_i \phi(||x - x_i||) + \sum_{j=1}^{M} d_j p_j(x), \; x \in \mathbb{R}^s. \tag{1.10}$$

with the additional constraints

$$\sum_{i=1}^{N} c_i p_j(x_i) = 0, \; j = 1, 2, \ldots, M. \tag{1.11}$$

Adding the extra constraints and the polynomial conditions to the interpolant, we find the system of linear equations,

$$\begin{bmatrix} A & P \\ P^T & O \end{bmatrix} \begin{bmatrix} c \\ s \end{bmatrix} = \begin{bmatrix} y \\ 0 \end{bmatrix},$$

where now the matrices and arrays have the following dimensions: A is $N \times N$, P is $N \times M$, O is $M \times M$ of zeros; c, y are $N \times 1$, s and 0 are $M \times 1$.

The addition of polynomials of degree not more than $m-1$ guarantees polynomial precision, meaning that if the data comes from a polynomial of degree less than or equal to $m-1$ they are fitted by that polynomial.

1.3.1 Positive-definite matrices and functions

Definition 1.3. A real symmetric matrix A is called positive semi-definite if its associated quadratic form $c^T Ac \geq 0$, that is

$$\sum_{i=1}^{N} \sum_{j=1}^{N} c_i c_j A_{i,j} \geq 0, \tag{1.12}$$

for $c \in \mathbb{R}^n$. If the quadratic form (1.12) is zero only for $c = 0$ then A is called positive definite.

Hence, if in (1.8) the basis ϕ_i generates a positive definite interpolation matrix that we would always have a well-defined interpolation problem. In order to get such property, we need to introduce the class of positive definite functions.

Definition 1.4. A continuous complex valued function $\phi : \mathbb{R}^s \to \mathbb{C}$ is called positive semi-definite if, for all $N \in \mathbb{N}$, all sets of pairwise distinct points $X = \{x_1, \ldots, x_N\} \subset \mathbb{R}^s$ and $c \in \mathbb{C}^N$ the quadratic form

$$\sum_{i=1}^{N} \sum_{j=1}^{N} c_i \overline{c_j} \phi(x_i - x_j) \geq 0, \tag{1.13}$$

is nonnegative. The function ϕ is then called positive definite if the quadratic form above is positive for $c \in \mathbb{C}^N$, $c \neq 0$.

One of the most celebrated results on positive definite functions is their characterization in terms of Fourier transforms established by Bochner in 1932.

Theorem 1.3. *(Bochner)[6] A (complex-valued) function $\Phi \in C(\mathbb{R}^s)$ is positive definite on $\mathbb{R}^s$ if and only if it is the Fourier transform of a finite non-negative Borel measure μ on $\mathbb{R}^s$, i.e.*

$$\Phi(x) = \frac{1}{\sqrt{(2\pi)^s}} \int_{\mathbb{R}^s} e^{-ixy} d\mu(y), \; x \in \mathbb{R}^s.$$

1.3.2 Completely monotone functions

Definition 1.5. [29](p.47) A function $\phi : [0, \infty) \to \mathbb{R}$ that is $\mathcal{C}[0, \infty) \cap C^\infty(0, \infty)$ and satisfies

$$(-1)^k \phi^{(k)}(r) \geq 0, \;\; r > 0, \;\; k = 0, 1, 2, \dots \tag{1.14}$$

is called completely monotone.

Here we enumerate some of the most important positive definite functions showing that they are completely monotone.

1. The function $\phi(r) = \epsilon$, $\epsilon \geq 0$ is completely monotone on $[0, \infty)$.
2. The function $\phi(r) = e^{-\epsilon r}$, $\epsilon \geq 0$ is completely monotone on $[0, \infty)$ since
$$(-1)^k \phi^{(k)}(r) = \epsilon^k e^{-\epsilon r} \geq 0, \;\; k = 0, 1, 2, \dots .$$
3. The function $\phi(r) = \dfrac{1}{(1+r)^\beta}$, $\beta \geq 0$ is completely monotone on $[0, \infty)$ since
$(-1)^k \phi^{(k)}(r) = (-1)^{2k} \beta(\beta+1) \dots (\beta + k - 1)(1+r)^{-\beta-k} \geq 0, k = 0, 1, 2, \dots$

Theorem 1.4. *(Hausdorff–Bernstein–Widder)[84](p.91) A function $\Phi : [0, \infty) \to \mathbb{R}$ is completely monotone on $[0, \infty)$ if and only if it is the Laplace transform of a nonnegative finite Borel measure ν, i.e. it is of the form*

$$\Phi(r) = \int_0^\infty e^{-rt} d\nu(t).$$

Theorem 1.5. *(Schoenberg)[84](p.93) A function ϕ is completely monotone on $[0, \infty)$ if and only if $\Phi = \phi(\| \, . \, \|_2^2)$ is positive semi-definite on $\mathbb{R}^s$ for all s.*

Multiply monotone functions

This characterization allows to check when a function is positive definite and radial on $\mathbb{R}^d$ for some fixed d.

Definition 1.6. [29](p.49) A function $\phi : (0, \infty) \to \mathbb{R}$ which is $\mathcal{C}^{s-2}(0, \infty)$, $s \geq 2$ and for which $(-1)^k \phi^{(k)}(r) \geq 0$, non-increasing and convex for $k = 0, 1, \dots, s-2$ is called s times monotone on $(0, \infty)$. In case $s = 1$ we only require $\phi \in \mathcal{C}(0, \infty)$ to be non-negative and non-increasing.

Characterizing positive definite functions using more comprehensible approach based on the definition of completely monotone and multiply monotone functions can be found in [11, 29, 84].

1.3.3 Conditionally positive definite functions

Definition 1.7. [84](p.97) A continuous function $\Phi : \mathbb{R}^s \longrightarrow \mathbb{C}$ is said to be conditionally positive semi-definite of order m in $\mathbb{R}^s$, if

$$\sum_{i=1}^N \sum_{j=1}^N c_i \overline{c_j} \phi(x_i - x_j) > 0, \tag{1.15}$$

for any N set $X = \{x_1, \ldots, x_N\} \subset \mathbb{R}^s$ of pairwise distinct points, and $c = (c_1, \ldots, c_N)^T \subset \mathbb{C}^N$ such that

$$\sum_{k=1}^{N} c_k p(x_k) = 0, \tag{1.16}$$

for any complex-valued polynomial p of degree $\leq m-1$. The function Φ is then called conditionally positive definite of order m on $\mathbb{R}^s$ if the quadratic form (1.15) vanishes only when $c \equiv 0$.

The first important fact concerning conditionally positive (semi)-definite functions is their order. To this aim holds the following important result.

Remark 1.2.

- A function which is conditionally positive (semi)-definite of order m is also conditionally positive (semi)-definite of any order $s \geq m$.
- A function that is conditionally positive (semi)-definite of order m in $\mathbb{R}^s$ is also conditionally positive (semi)-definite of order m on $\mathbb{R}^k$ with $k \leq s$.

Theorem 1.6. *[84](p.99) Suppose Φ is conditionally positive definite of order 1 and that $\Phi(0) \leq 0$. Then the matrix $A \in \mathbb{R}^{N \times N}$, i.e. $A_{i,j} = \Phi(x_i - x_j)$, has one negative and $N-1$ positive eigenvalues. In particular it is invertible.*

The most used positive definite RBFs and conditionally positive definite RBFs are given in tables 1.3-1.4 respectively. The representations of multiquadric, gaussian, inverse multiquadric, inverse quadric and polyharmonic splines RBFs are given in figures 1.4-1.5-1.6.

Name	$\phi(r)$
Inverse Multiquadric (IMQ)	$\phi(r) = \dfrac{1}{\sqrt{r^2+\epsilon^2}}$
Gaussian Function (GS)	$\phi(r) = e^{-\epsilon r^2}$

Table 1.3: Positive definite radial basis functions.

Name	$\phi(r)$	Order
Multiquadric (MQ)	$\phi(r) = (r^2+\epsilon^2)^k$, $k > 0$, $k \notin \mathbb{N}$	$\lceil k \rceil + 1$
Inverse Multiquadric (IMQ)	$\phi(r) = (r^2+\epsilon^2)^{-k}$, $k > 0$, $k \notin \mathbb{N}$	0
Polyharmonic spline	$\phi(r) = r^{2k-1}$, $k \in \mathbb{N}$	$\lceil k/2 \rceil + 1$
Polyharmonic spline	$\phi(r) = r^{2k} ln(r)$, $k \in \mathbb{N}$	$\lceil k/2 \rceil + 1$
Thin Plate Spline (TPS)	$\phi(r) = r^2 ln(r)$	2

Table 1.4: Conditionally positive definite radial basis functions, where $\lceil k \rceil$ denotes the nearest integers less than or equal to k, and N the natural number, ϵ a positive constant which is known as the shape parameter .

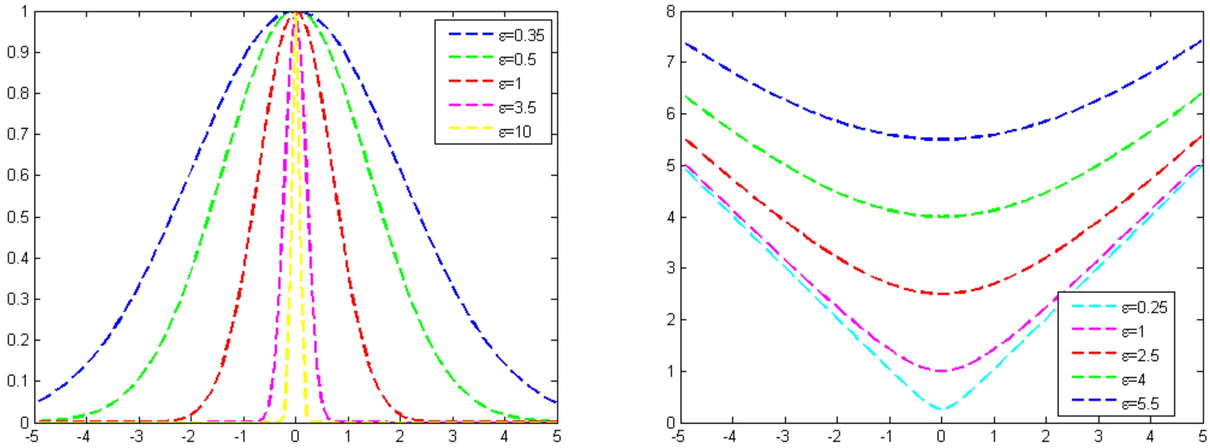

Figure 1.4: Graph of gaussian RBF (left), multiquadric RBF (right), for a center $x = 0$, with different values of shape parameter.

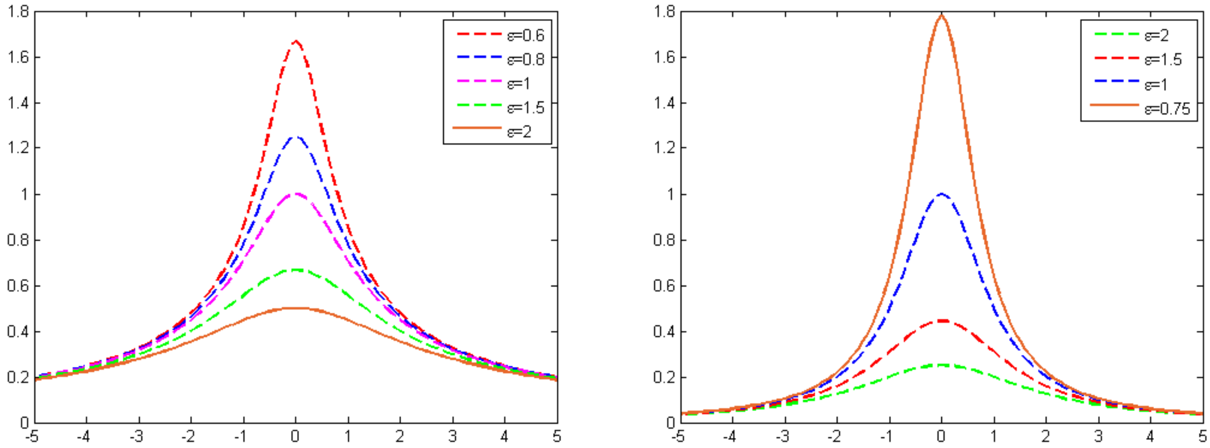

Figure 1.5: Graph of inverse multiquadric RBF (left), inverse quadric RBF (right), for a center $x = 0$, with different values of shape parameter.

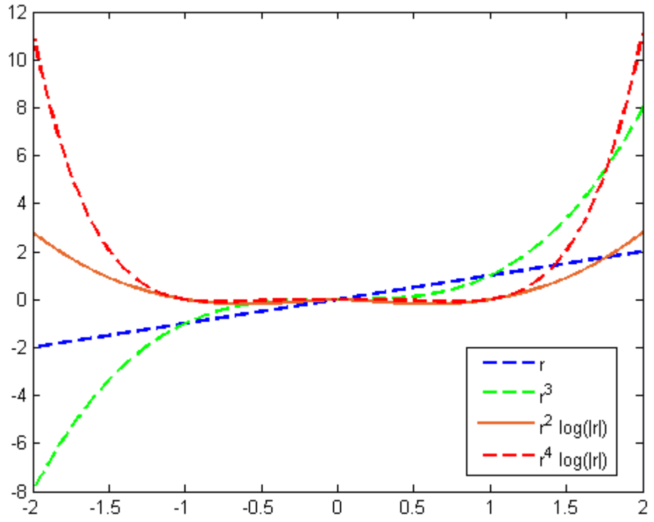

Figure 1.6: Graph of polyharmonic splines.

1.3.4 Stability

As observed by Schaback [68] there is a trade-off between the accuracy of the interpolation and the condition number of the matrix defined in equation (1.9) in the 2-norm is defined as follows,

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

where σ_{min} is the smallest singular value and σ_{max} the largest singular value of A. The shape parameter affects both the conditioning of the system matrix and the accuracy of the RBFs method as shown in figure 1.7.

In order to improve the accuracy of the interpolant we can change the shape parameter ϵ to use ‘flat’ basis functions , which leads to a more ill-conditioned problem since the condition number of the matrix A increases. This trade-off has been called the uncertainty principle (Phrase used to describe the fact that a RBF approximant can not at the same time be accurate and well conditioned). Alternatively we can increase the number of interpolation points to increase the accuracy, but this also leads to a larger condition number of A.

increasingly flat basis functions provide more accurate interpolants, but are unusable because of the numerical instabilities from ill-conditioning. This problem is addressed in [33], where algorithms for the stable computation of these interpolants are provided.

1.3.5 Shape parameter

Many RBFs, including all of the ones named here, have a variable ϵ in their definitions. This variable is called the shape parameter. Finding the shape parameter that will produce the most accurate approximation is a topic of current research.

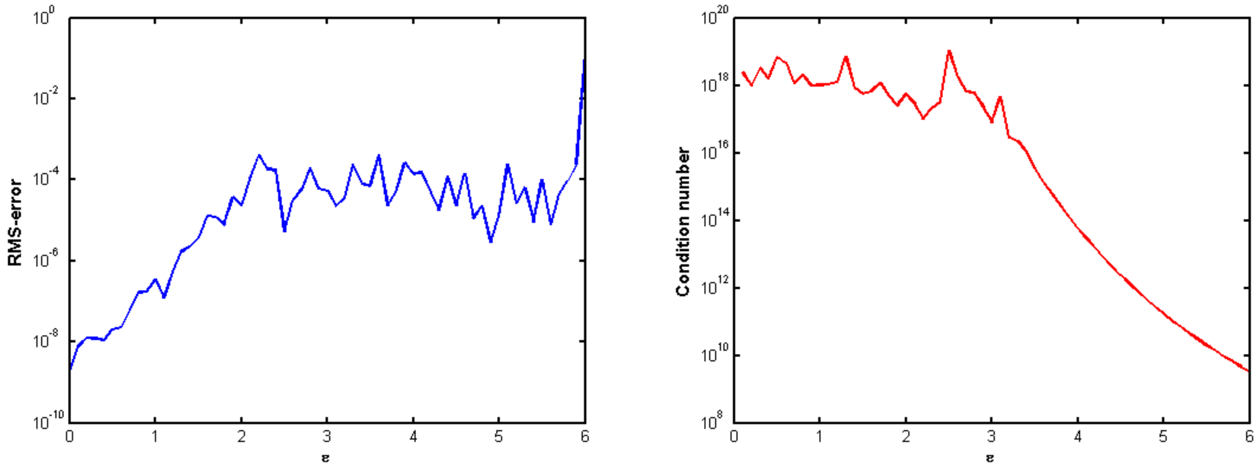

Figure 1.7: Gaussian RBF ($\phi(r) = e^{-(\epsilon r)^2}$) interpolation of the function, $f(x) = e^{-x^2} sin(x)$, using N=20 of Chebyshev nodes. RMS-error (RMS-error=$\frac{1}{\sqrt{N}}||s - f||_2$) versus the shape parameter (Left), condition number versus the shape parameter (right).

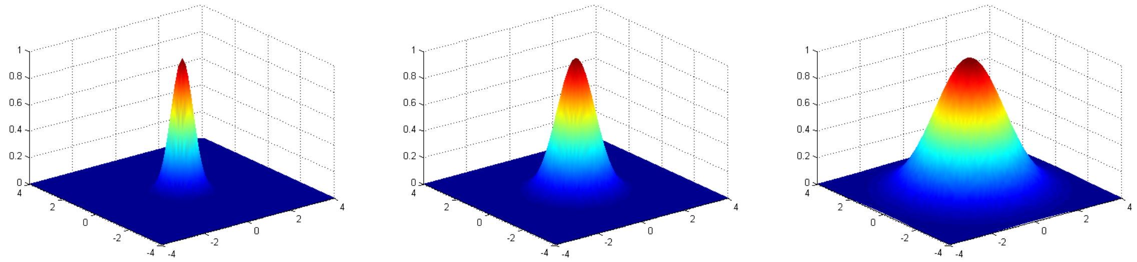

Figure 1.8: Graph of Gaussian RBF centered at the origin, with a shape parameter $\epsilon = 3$ (left),with a shape parameter $\epsilon = 1$ (center), with a shape parameter $\epsilon = 0.3$ (right) .

Choosing optimal shape parameter

In recent years, a lot of continued efforts of many authors to establish the theory of evaluating the optimal shape parameter ϵ in the MQ radial basis function interpolation. However, such an explicit formula is only available in special cases. Consequently, numerically determining the optimal ϵ proves to be essential. Numerical experiments find that the best ϵ, via a numerical scheme, may not be theoretically optimal. A large shape parameter results in a well conditioned system matrix; however, the approximation using the RBF is poor. If one chooses to use a small shape parameter, this results in a very accurate RBF approximation, but the system matrix is ill-conditioned. Many strategies for selecting an optimal value of ϵ were suggested.

Constant shape parameter

Many scientists and mathematicians use the constant shape parameter for interpolation of data [34, 36, 44].

Definition 1.8. [36] Hardy's ϵ is given by

$$\epsilon = 0.815d \text{ with } d = \frac{1}{N}\sum_{i=1}^{N} d_i,$$

where d_i is the distance from the i^{th} center to the nearest neighbor and N is the number of centers.

Definition 1.9. [34] Franke's ϵ is given by

$$\epsilon = \frac{1.25D}{\sqrt{N}},$$

where D is the diameter of the smallest circle encompassing all the center locations and N is the number of centers.

Many authors have tried to construct a satisfactory formula for the shape parameter in MQ interpolation as noted in [32, 79]. When creating an optimal shape parameter, one must combat the uncertainty principle. The goal of finding a shape parameter formula for the interpolant is to provide good accuracy with not too high of a condition number.

Variable shape parameter

When the theory is established for radial basis functions, a constant shape parameter was used. If one uses a variable shape parameter, the complexity of theory becomes extremely difficult to explain. In [7] there are somewhat restrictive sufficient conditions that show the system matrix A is non-singular with a variable shape parameter. One positive aspect of a variable shape parameter is that it creates distinct entries in the RBF matrices which can lead to lower condition numbers [80]. The downside to a variable shape parameter is that it caused A to be nonsymmetric. Recall that if one uses a constant shape parameter, the system matrix A will be symmetric. There are additional papers that used the variable shape parameter see- [7, 49, 50].

Variable linear shape parameter

The variable linear shape is a $1 \times N$ matrix that contains shape parameters generated by the formula,

$$\epsilon_j = \epsilon_{min} + (\frac{\epsilon_{max} - \epsilon_{min}}{N-1})j, \ j = 0, 1, \ldots N-1,$$

where ϵ_{min} and ϵ_{max} are the maximum and the minimum values for the variable shape ϵ respectively. The representation of variable linear shape parameter is given in figure 1.9.

Exponentially varying shape parameter

The exponentially varying shape is a $1 \times N$ matrix that contains shape parameters generated by the formula,

$$\epsilon_j = \left[\epsilon_{min}^2 \Big(\frac{\epsilon_{max}^2}{\epsilon_{min}^2}\Big)^{\frac{j-1}{N-1}}\right]^{\frac{1}{2}}, \text{ for} j = 1, 2, \ldots, N,$$

where ϵ_{min} and ϵ_{max} are the maximum and the minimum values for the variable shape ϵ respectively. The representation of exponentially varying shape parameter is given in figure 1.10.

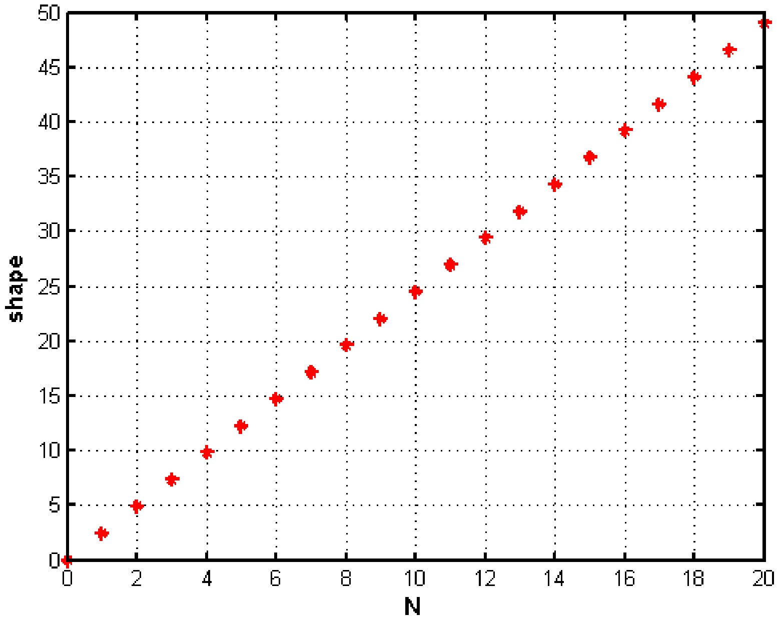

Figure 1.9: Plot of variable linear shape parameter with $\epsilon_{min} = 1$, $\epsilon_{max} = 50$, and $N = 20$.

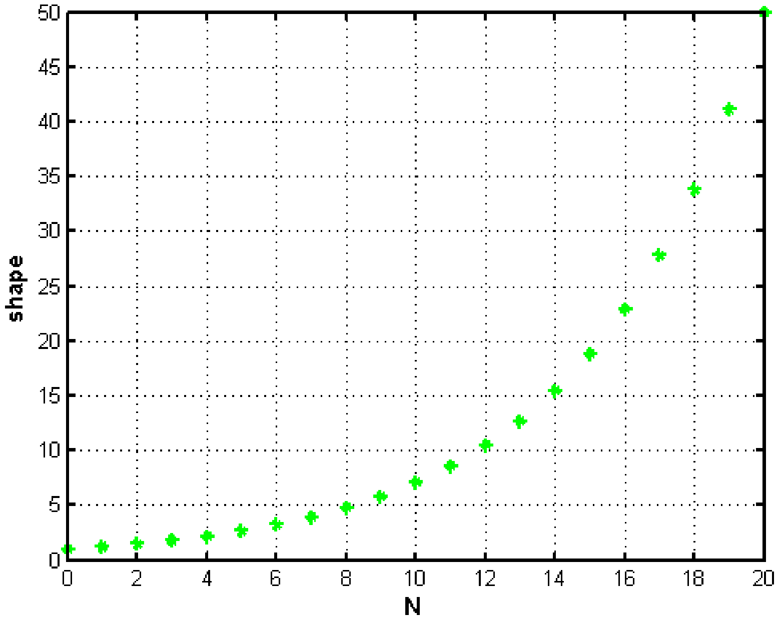

Figure 1.10: Plot of exponentially variable shape parameter with $\epsilon_{min} = 1$, $\epsilon_{max} = 50$, and $N = 20$.

Variable random shape parameter

The variable random shape is a $1 \times N$ matrix that contains shape parameters generated by the formula,

$$\epsilon_j = \epsilon_{min} + (\epsilon_{max} - \epsilon_{min}) rand(1, N),$$

where ϵ_{min}, ϵ_{max} are the maximum and the minimum values for the variable shape ϵ respectively. The representation of variable random shape parameter is given in figure 1.11. The representation of absolute errors using linear variable, exponentially varying and random variable shape parameter is presented in figure 1.12.

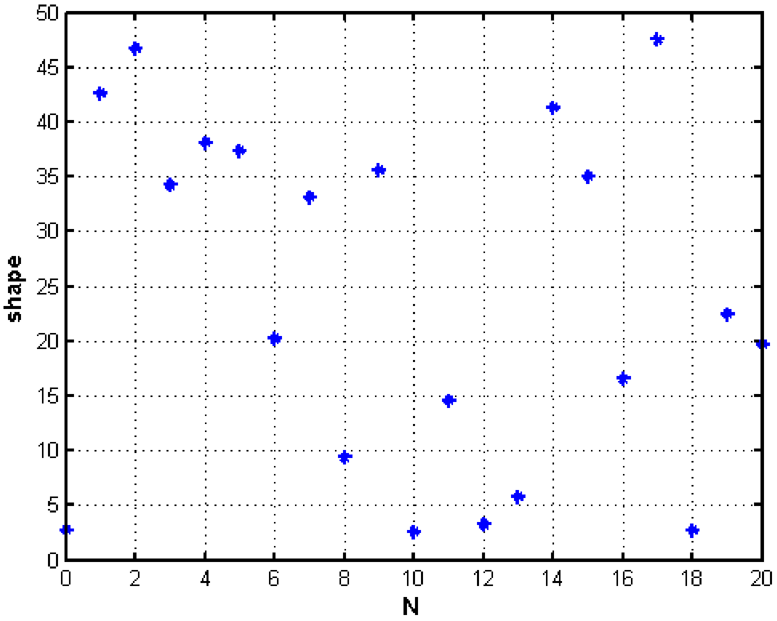

Figure 1.11: Plot of random variable shape parameter with $\epsilon_{min} = 1$, $\epsilon_{max} = 50$, and $N = 20$.

The power function

The error of interpolation is given by

$| f(x) - s(x) | \leq P_{\phi,\chi}(x) || f ||_{\mathcal{N}_\phi(\Omega)}$, where $P_{\phi,\chi}(x)$ denotes the power function. This estimate decouples the interpolation error into a component independent of the data function f and one depending on f . Once we have decided on a basic function f and a data set χ, we can use the power function based on scaled versions of ϕ to optimize the error component that is independent of f. The power function can be computed via

$P_{\phi,\chi}(x) = \sqrt{\phi(x,x) - (b(x))^T A^{-1} b(x)}$, where A is the interpolation matrix,

and $b = [\phi(., x_1), \ldots, \phi(., x_n)]^T$.

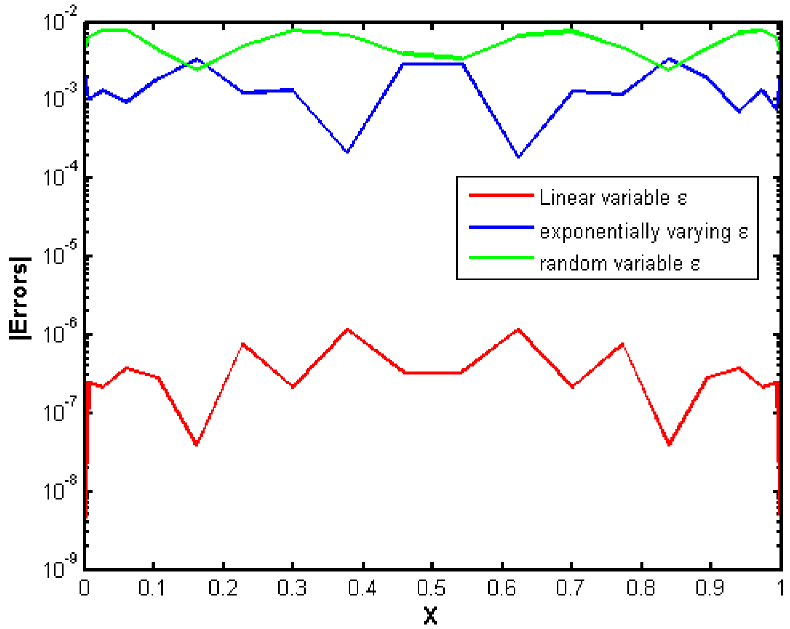

Figure 1.12: Plot of absolute errors for interpolating $sinc(x + 1)$ using linear variable, exponentially varying and random variable shape parameter with $\epsilon_{min} = 2.1$, $\epsilon_{max} = 7.6$, and $N = 20$, using $M = 9$ Chebyshev points.

* Advantage: objective and does not depend on any knowledge of the data function.

* Disadvantage: will not be an optimal one since the second component of the error bound also depends on the basic function via the native space norm (which changes when A is scaled).

The power function strategy for one and two dimensional interpolation using gaussian RBF is given in figure 1.13.

Trial and error strategy

It is the simplest approach. It consists in performing various interpolation experiments with different values of the shape parameter. The best parameter, say ϵ will be the one that minimize the interpolation error. In figure 1.14, we plot the interpolation max-error varying ϵ for different data points, using the Gaussian kernel in the univariate case. The minimum of every curve gives the "optimal" value.

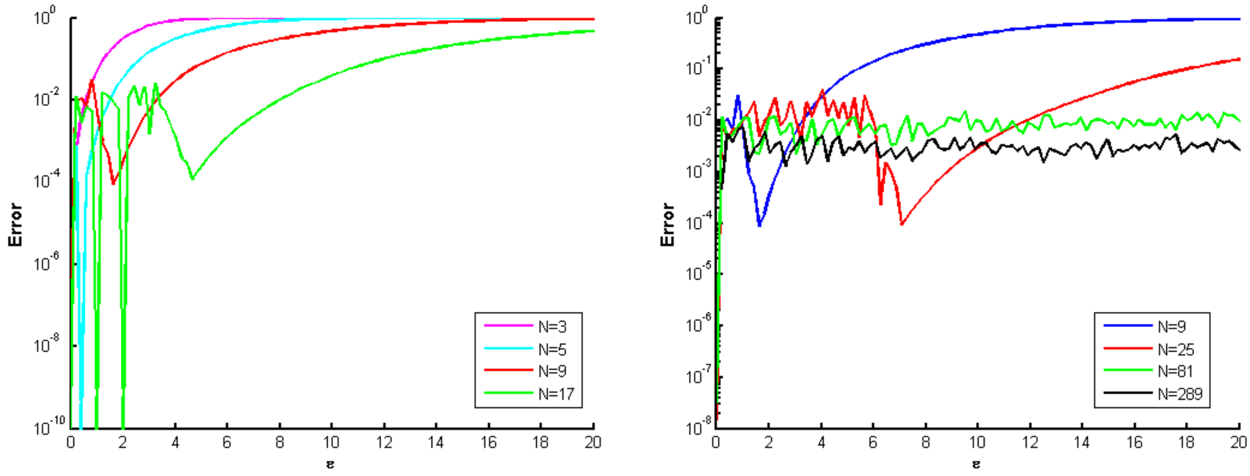

Figure 1.13: The power function strategy by gaussian RBF in one dimensional (left), and two dimensional (right) for $\epsilon \in [0, 20]$, taking 100 values of ϵ and for different equispaced data points.

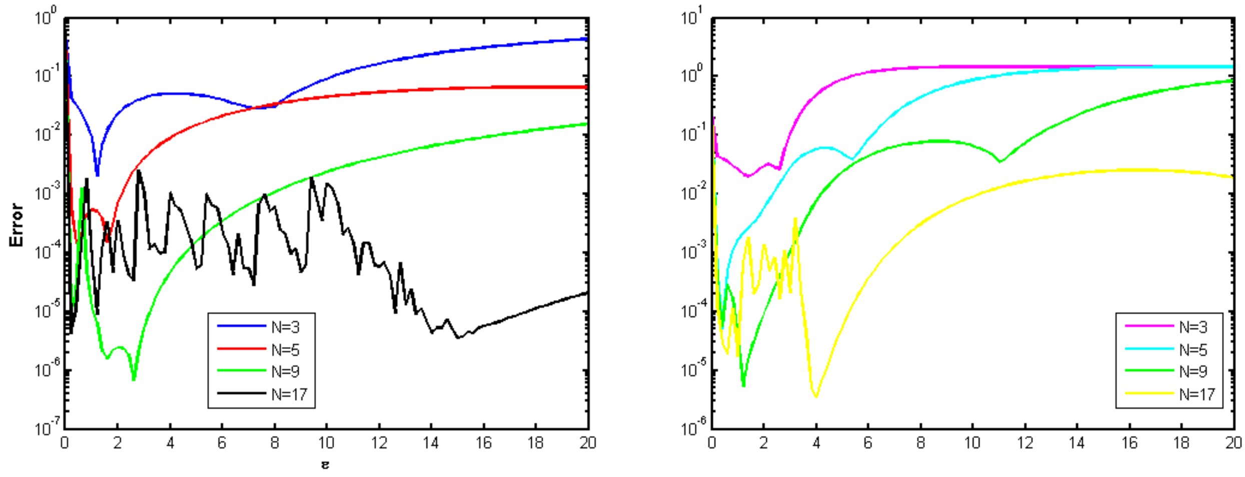

Figure 1.14: Trial and error strategy for the interpolation of the $sin(x)$ function (left) and Franke's test function $f(x) = exp(-x) + sin(2x)$ (right) by the gaussian for $\epsilon \in [0, 20]$, taking 100 values of ϵ and for different equispaced data points.

Cross Validation method

This method is popular in the statistics literature, known in the case the 2-norm is used as PRESS (Predictive Residual Sum of Squares). The optimal value ϵ is obtained minimizing the (least-squares) error for a fit of the data values based on an interpolant for which one of the centers is left out.

1- **Leave one out cross validation** $(LOOCV)$
Cross validation attempts to test the accuracy of a method by separating the available data into two or more parts. One part of the data set is used to construct an approximation, and error is measured using a different part of the data set. Leave one out cross validation $(LOOCV)$ uses $(N-1)$ points from the data set to construct an approximation, then checks that approximation's error at the remaining data site. The procedure is repeated leaving out each data site once, and the resulting set of errors are used to estimate the method's relative accuracy .

$$LOOCV(\epsilon) = \sum_{i=1}^{N} |s_i(x_i) - y_i|^2,$$

where s_i is the interpolant when excluding the points x_i and using functions with shape parameter ϵ.

Shmuel Rippa [76] showed that $s_i(x_i) - y_i = \dfrac{\lambda_i}{A_{ii}^{-1}}$ where λ_i is the i-th coefficient of the interpolant f constructed using the full data set and A_{ii}^{-1} is the i-th diagonal element of A^{-1}. The vector λ can be found by solving $A\lambda = y$, and the diagonal elements of A^{-1} can be computed using a matrix factorization. As a result, computing the $LOOCV$ using Rippa's formula has computational complexity $O(N^3)$. For the following experiments, $LOOCV$ will be computed using the formula

$$LOOCV(\epsilon) = \sum_{i=1}^{N} \left[\frac{\lambda_i}{A_{ii}^{-1}}\right]^2.$$

2- **Generalized cross validation** (GCV)
Generalized cross validation (GCV) is a variation of leave one out cross validation which replaces the diagonal elements of A^{-1} with their average. GCV is similar to $LOOCV$, but has some invariance properties which LOOCV lacks. As with $LOOCV$, the computational complexity of GCV is $O(N^3)$. In the following experiments, GCV is calculated by using the formula

$$GCV(\epsilon) = \frac{\displaystyle\sum_{i=1}^{N} \lambda_i^2}{\left[\frac{1}{N}\displaystyle\sum_{i=1}^{N} A_{ii}^{-1}\right]^2}.$$

3- **Maximum Likelihood Estimator** (MLE)
Another method for predicting which shape parameter ϵ will minimize the error of an RBF interpolation is to use a maximum likelihood estimator (MLE). Assuming that f is a Gaussian process, maximizing the likelihood function is equivalent to minimizing $y^t A^{-1} y.[det(A)]^{1/N}$. This formula rapidly approaches 0 as N increases, causing numerical error. The resulting numerical error can be prevented by taking the logarithm of the

function in its computation and applying the identity $det(A) = \prod_{i=1}^{N} \lambda_i(A)$, where $\lambda_i(A)$ is the i-th eigenvalue of A. Additionally, computing A^{-1} can be avoided because $A^{-1}\lambda = c$. MLE is calculated by using the formula $MLE(\epsilon) = log(y^t c) + \frac{1}{N}\sum_{i=1}^{N} log(\lambda_i(A))$. Computing the coefficients λ_i and computing the eigenvalues of a matrix each require $O(N^3)$ operations. So the value of this function for a single shape parameter ϵ has computational complexity $O(N^3)$. The LOOCV strategy for the interpolation of the *sinc* function by gaussian RBF in one and two dimensional is shown in figure 1.15.

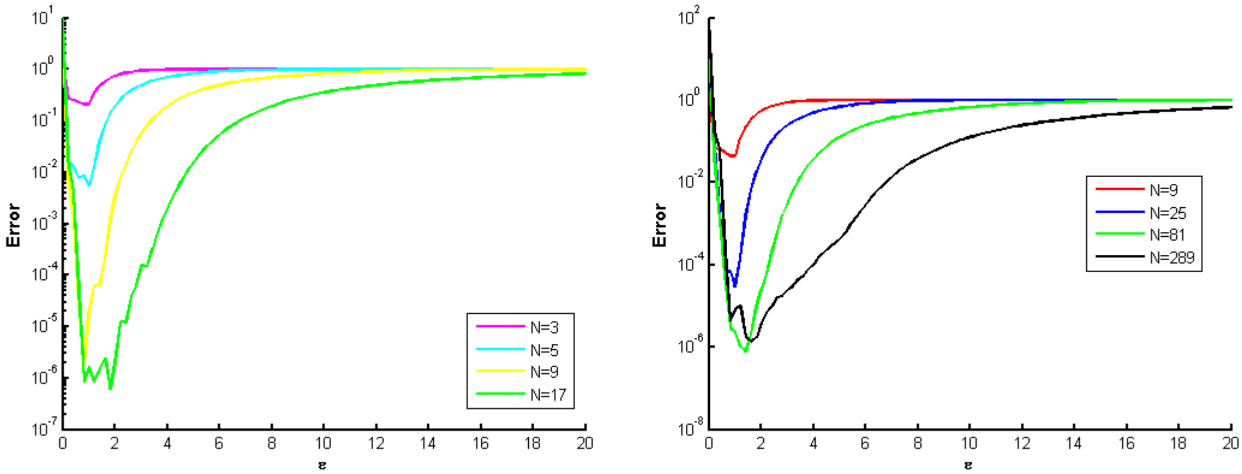

Figure 1.15: LOOCV strategy for the interpolation of the *sinc* function by gaussian RBF in one dimensional (left), and two dimensional (right) for $\epsilon \in [0, 20]$, taking 101 values of ϵ and for different equispaced data points.

Other algorithms in litterature for choosing optimal shape parameter are given in [17, 69, 76, 81].

1.3.6 Data sets

Depending on the type of approximation problem we are given, we may or may not be able to select where the data is collected, i.e., the location of the data sites or design. Standard choices in low space dimensions are depicted in figure 1.16. In higher space dimensions it is important to have space-filling (or low-discrepancy) quasi-random point sets. Examples include: Halton points, Sobol points, lattice designs, Latin hypercube designs and quite a few others (digital nets, Faure, Niederreiter, etc). The representations of Halton points, Sobol points, lattice and Latin points are given in figures 1.17-1.18.

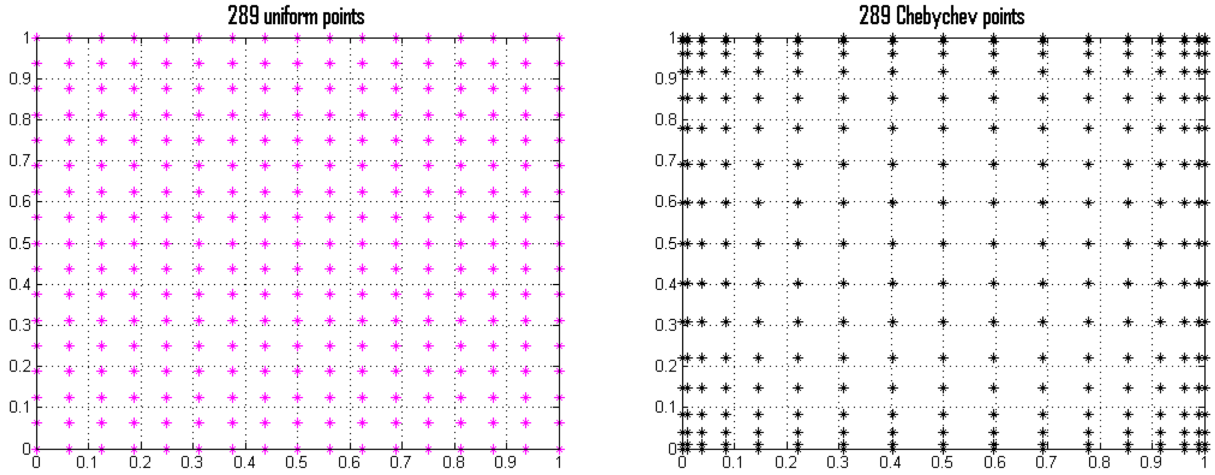

Figure 1.16: Tensor products of equally spaced points and tensor products of Chebyshev points.

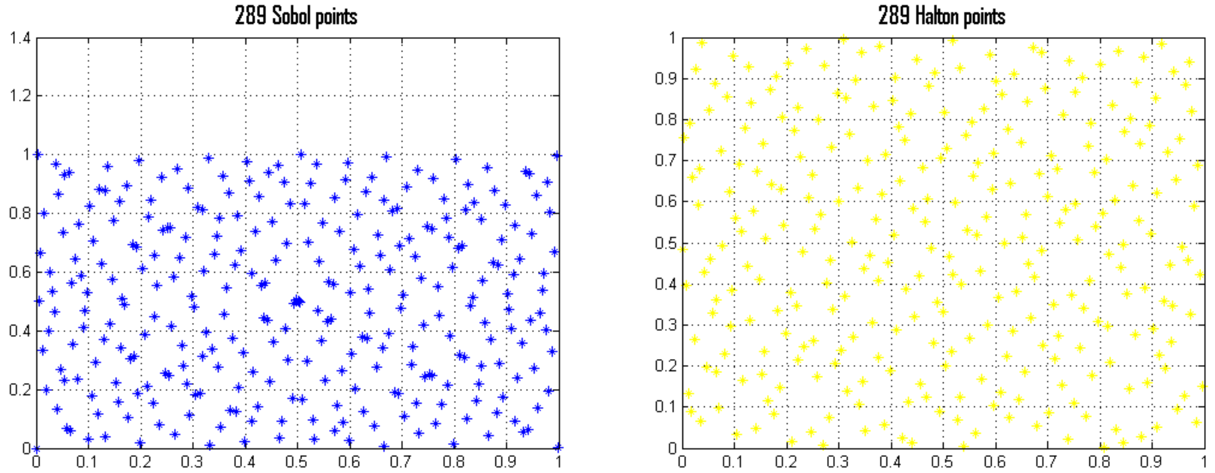

Figure 1.17: Graph of Sobol and Halton nodes.

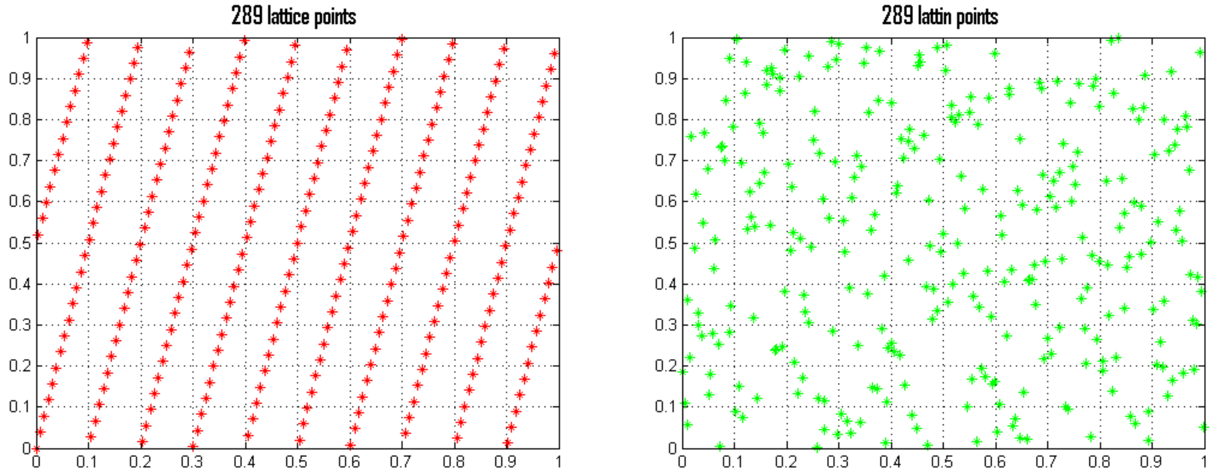

Figure 1.18: Graph of Lattice and Lattin nodes.

Algorithms for choosing centers of radial basis funcions

RBF methods have been praised for their simplicity and ease of implementation in multivariate scattered data approximation. But both the approximation quality and stability depend on the distribution of the center set. It leads immediately to the problem of finding good or even optimal point sets for the reconstruction process. Many methods are constructed for center choosing. We summarize some effective center location methods like greedy algorithms, arclength equipartition methods and k-mean clustering methods.

Greedy algorithm

In order to minimizing the power function, Marchi S.D., Schaback R. and Wendland H. constructed a numerical greedy algorithm produces near-optimal point sets by recursively adding one of the maxima points of the power function w.r.t. the preceding set [60, 61]. Obviously, greedy algorithm is data-dependent and adaptive algorithm. It is described as follow

Take a set $X = \{x_1, \dots x_N\} \subseteq \Omega \subseteq \mathbb{R}^s$ of N pairwise distinct points coming from a compact subset Ω of $\mathbb{R}^s$.

(1) Let $X_1 = \{x_1\}$ for $x_1 \in \Omega$ arbitrary

(2) Do $X_j := X_{j-1} \cup \{x_j\}$ with

$$P_{\phi,X_{j-1}}(x_j) = \parallel P_{\phi,X_{j-1}} \parallel_{L^\infty(\Omega)},\ j \geq 2, \tag{1.17}$$

until $\parallel P_{\phi,X_j} \parallel_{L^\infty(\Omega)}$ is small enough.

In practice, the maxima is taken over some large discrete set $X \subset \Omega \subset \mathbb{R}^s$. The convergence rate of the above Greedy algorithm is at least like

$$\parallel P_{\phi,X_j} \parallel_{L_\infty(\Omega)} \leq C j^{1-s}, \tag{1.18}$$

where C is a constant.

Based on numerous numerical experiments of Greedy Algorithm, the same authors suggested a geometric greedy algorithm which is data-independent.

ArcLength equipartition like algorithm

Based on the idea that to display a function with some finite discrete sampling data efficiently, one requires more sampling data where the function is more oscillatory, and less sampling data where the function is more flat, Wu Z.M. [88] and Sarra S.A. [71] both used arclength equipartition algorithm to solve partial differential equations.

k-means clustering algorithm

Finally, k-means clustering algorithm commonly used in radial basis function neural networks is easy to implement and of high performance [94]. The working process of k-means: first, choose arbitrary k points as the initial cluster centers. For all other points, compute their Euclidean distances to the k-cluster centers, and add each to its nearest cluster. Then recalculate the k cluster centers by taking the geometric center of each cluster, and repeat the above process until the center errors go below a given threshold. This is also a data-independent method.

Other Algorithms for choosing centers for RBFs are given in [21, 31, 46].

Chapter 2

Compactly supported RBF and some applications for solving integral and partial differential equations

This chapter is consacred to introducing the concept of compactly supported RBF and its applications with globaly RBF for solving different types of integral and partial differential equations. A comparaison between stationary and non-stationary approaches for RBFs approximation is also given.

2.1 Compactly supported radial basis functions

After successful studies and tests for the result of partial differential equations, such as the resolution of hydrodynamic equations by Hon et al, the numerical results demonstrated that the MQ diagrams are very assuring than the finite element method. However, the MQ method requires solving a linear system with a full matrix, which could make the method cumbersome and very expensive once you have to do hundreds of collocation points. To overcome this problem, researchers have developed a sheme based on radial basis functions with compact support (CSRBFs). The compactly supported radial basis functions can cover the global schemes that the simple RBF methods are weak to solve as in the case of the badly conditioned matrices.

The accuracy of the RBFs method depends on the value of the shape parameter which is still an unsolved problem, also the resulting interpolation matrix is dense and highly ill conditioned. So it was suggested the use of compactly supported radial basis functions which can reduce the resultant full matrix to a sparse one, also the operation of the banded matrix system could reduce the ill-conditioning of the resultant coefficient matrix when using the global radial basis functions.

Compactly supported radial functions can be strictly positive definite on $\mathbb{R}^s$ only for a fixed maximal s-value. It is not possible for a function to be strictly positive definite and radial on $\mathbb{R}^s$ for all s and also have a compact support. Therefore we give characterization and construction of functions that are compactly supported, strictly positive definite and radial on $\mathbb{R}^s$ for some fixed s.

According to Bochner's work [6], a function is strictly positive definite and radial on $\mathbb{R}^s$ if its s-variate Fourier transform is non-negative.

The Bessel function $\mathcal{J}$ of the first kind of order $v \in \mathbb{C}$ can be expressed as follow

$$J_\nu := \sum_{k=0}^{\infty} \frac{(-1)^k (z/2)^{2k+v}}{k!\Gamma(k+v+1)}, \text{ for } z \in \mathbb{C}\backslash 0.$$

such that the Γ-function is given as follow

$$\Gamma(z) := \lim_{n \longrightarrow \infty} \frac{n! n^z}{z(z+1)\dots(z+n)}, \text{ for } z \in \mathbb{C}.$$

Theorem 2.1. *[84](p.119) Suppose $\Phi \in L^1(\mathbb{R}^s) \cap C(\mathbb{R}^s)$ is radial, i.e. $\Phi = \phi(\| \,.\, \|_2)$, $x \in \mathbb{R}^s$. Then its Fourier transform $\widehat{\Phi}$ is also radial, i.e. $\widehat{\Phi} = \mathcal{F}_s(\| \,.\, \|_2)$ with*

$$\widehat{\Phi}(x) = \mathcal{F}_s \phi(r) = r^{-(s-2)/2} \int_0^\infty \phi(t) t^{s/2} \mathcal{J}_{(s-2)/2}(rt) dt.$$

Operators for Radial Functions and Dimension Walks

Schaback and Wu [73] defined an integral operator and its inverse differential operator, and discussed an entire calculus for how these operators act on radial functions. These operators will facilitate the construction of compactly supported radial functions.

Definition 2.1. [84](p.121) Let ϕ be given such that $t \rightarrow t\phi(t) \in L^1[0, \infty)$, then we define

$$(\mathcal{I}\phi)(r) = \int_r^{+\infty} t\phi(t) dt, \ r \geq 0.$$

For even $\phi \in C^2(\mathbb{R})$ we define

$$(\mathcal{D}\phi)(r) = -\frac{1}{r}\phi^{'}(r), \ r \geq 0.$$

In both cases the resulting functions are to be interpreted as even functions using even extension.

Theorem 2.2. *[84](p.121-122)*

1- Both $\mathcal{D}$ and $\mathcal{I}$ preserve compact support, i.e., if ϕ has compact support, then so do $\mathcal{D}\phi$ and $\mathcal{I}\phi$.

2- If $\phi \in C(\mathbb{R})$ and $t \rightarrow \phi(t) \in L^1[0, \infty)$, then $\mathcal{D}\mathcal{I}\phi = \phi$.

3- If $\phi \in C^2(\mathbb{R})$ and $\phi^{'} \in L^1[0, \infty)$, then $\mathcal{I}\mathcal{D}\phi = \phi$.

4- If $t \rightarrow t^{s-1}\phi(t) \in L^1[0, \infty)$ and $s \geq 3$, then $\mathcal{F}_s(\phi) = \mathcal{F}_{s-2}(\mathcal{I}\phi)$.

5- If $\phi \in C^2(\mathbb{R})$ is even and $t \rightarrow t^s \phi^{'}(t) \in L^1[0, \infty)$, then $\mathcal{F}_s\phi = \mathcal{F}_{s+2}(\mathcal{D}\phi)$.

The operators $\mathcal{I}$ and $\mathcal{D}$ allow us to express s-variate Fourier transforms as $(s-2)$ or $(s+2)-$variate Fourier transforms, respectively.

Wendland's Compactly Supported Functions

In [84] Wendland constructed a popular family of compactly supported radial functions by starting with the truncated power function which we know to be strictly positive definite and radial on $\mathbb{R}^s$ for $s \leq 2l - 1$, and then walking through dimensions by repeatedly applying the operator $\mathcal{I}$.

Definition 2.2. [84](p.128) With $\phi_l(r) = (1-r)_+^l$ we define

$$\phi_{s,k} = \mathcal{I}^k \phi_{\lfloor s/2 \rfloor + k + 1}.$$

It turns out that the functions $\phi_{s,k}$ are all supported on $[0,1]$ and have a polynomial representation there.

Theorem 2.3. *[84](p.128) The functions $\phi_{s,k}$ are strictly positive definite and radial on $\mathbb{R}^s$ and are of the form*

$$\phi_{s,k}(r) = \begin{cases} p_{s,k}(r), & r \in [0,1] \\ 0, & r > 1, \end{cases} \tag{2.1}$$

with a univariate polynomial $p_{s,k}$ of degree $\lfloor s/2 \rfloor + 3k + 1$. Moreover, $\phi_{s,k} \in C^{2k}(\mathbb{R})$ are unique up to a constant factor, and the polynomial degree is minimal for given space dimension s and smoothness $2k$.

Wendland gave recursive formulas for the functions $\phi_{s,k}$ for all s, k.

Example 2.1. Wendland's compactly supported functions $\phi_{s,k}$, for $k = 0, 1, 2, 3$, are written in the following form

- $\phi_{s,0}(r) = (1-r)_+^{\lfloor s/2 \rfloor + 1}$
- $\phi_{s,1}(r) \doteq (1-r)_+^{l+1}\left[(l+1)r + 1\right]$
- $\phi_{s,2}(r) \doteq (1-r)_+^{l+2}\left[(l^2+4l+3)r^2 + (3l+6)r + 3\right]$
- $\phi_{s,3}(r) \doteq (1-r)_+^{l+3}\left[(l^3+9l^2+23l+15)r^3 + (6l^2+36l+45)r^2 + (15l+45)r + 15\right],$

where $l := \lfloor s/2 \rfloor + k + 1$, and the symbol $\doteq$ denotes equality up to a multiplicative positive constant.

Example 2.2. For $s = 3$ we get some of the most commonly used functions as

- $\phi_{3,0}(r) = (1-r)_+^3, \ \in C^0 \cap SPD(\mathbb{R}^3)$
- $\phi_{3,1}(r) \doteq (1-r)_+^4\left[4r+1\right], \ \in C^2 \cap SPD(\mathbb{R}^3)$
- $\phi_{3,2}(r) \doteq (1-r)_+^6\left[35r^2+18r+3\right], \ \in C^4 \cap SPD(\mathbb{R}^3)$
- $\phi_{3,3}(r) \doteq (1-r)_+^8\left[32r^3 + (6l^2+36l+45)r^2 + (15l+45)r + 15\right], \ \in C^6 \cap SPD(\mathbb{R}^3).$

The abbreviation SPD means striclty positive definite.

Wu's Compactly Supported Functions

Wu presents another way to construct strictly positive definite radial functions with compact support [89]. He starts with the function

$$\phi(r) = (1-r^2)^l_+, \; l \in \mathbb{N}, \tag{2.2}$$

which is strictly positive definite and radial since we know that the truncated power function $\phi(\sqrt{.})$ is multiply monotone. Wu then constructs another function that is strictly positive definite and radial on $\mathbb{R}$ by convolution, i.e.,

$$\begin{aligned}\phi_l(r) &= (\phi * \phi)(2r)\\ &= \int_{-\infty}^{+\infty} (1-t^2)^l_+ + (1-(2r-t)^2)^l_+ dt.\end{aligned}$$

This function is strictly positive definite since its Fourier transform is essentially the square of the Fourier transform of ϕ. Just like the Wendland functions, this function is a polynomial on its support. In fact, the degree of the polynomial is $4l+1$, and $\phi_l \in C^{2l}(\mathbb{R})$.

Now, a family of strictly positive definite radial functions is constructed by a dimension walk using the $\mathcal{D}$ operator, i.e.,

$$\phi_{k,l} = \mathcal{D}^k \phi_l.$$

The functions $\phi_{k,l}$ are strictly positive definite and radial in $\mathbb{R}^s$ for $s \leq 2k+1$, are polynomials of degree $4l-2k+1$ on their support and in $C^2(l-k)$ in the interior of the support. On the boundary the smoothness increases to C^{2l-k}.

Example 2.3. For $l=3$ we can compute the three functions

$\phi_{k,3}(r)\mathcal{D}^k\phi_3(r) = \mathcal{D}^k((1-.^2)^3_+ + (1-.^2)^3_+)(2r),\ k = 0,1,2,3$. This results the following CSRBFs

$$\begin{aligned}
\phi_{0,3}(r) &\doteq (5 - 39r^2 + 143r^4 - 429r^6 + 429r^7 - 143r^9 + 39r^{11} - 5r^{13})_+\\
\phi_{0,3}(r) &\doteq (1-r)^7_+(5 + 35r + 101r^2 + 147r^3 + 101r^4 + 35r^5 + 5r^6),\ \in C^6 \cap SPD(\mathbb{R})\\
\phi_{1,3}(r) &\doteq (6 - 44r^2 + 198r^4 - 231r^5 + 99r^7 - 33r^9 + 5r^11)_+\\
\phi_{1,3}(r) &\doteq (1-r)^6_+(6 + 36r + 82r^2 + 72r^3 + 30r^4 + 5r^5),\ \in C^4 \cap SPD(\mathbb{R}^3)\\
\phi_{2,3}(r) &\doteq (8 - 72r^2 + 105r^3 - 63r^5 + 27r^7 - 5r^9)_+\\
\phi_{2,3}(r) &\doteq (1-r)^5_+(8 + 40r + 48r^2 + 25r^3 + 5r^4),\ \in C^2 \cap SPD(\mathbb{R}^5)\\
\phi_{3,3}(r) &\doteq (16 - 35r + 35r^3 - 21r^5 + 5r^7)_+\\
\phi_{3,3}(r) &\doteq (1-r)^4_+(16 + 29r + 20r^2 + 5r^3),\ \in C^0 \cap SPD(\mathbb{R}^7).
\end{aligned}$$

Remark 2.1. 1- For a prescribed smoothness the polynomial degree of Wendland's functions is lower than that of Wu's functions. For example, both Wendland's function $\phi_{3,2}$ and Wu's function $\phi_{3,1}$ are C^4 smooth and strictly positive definite and radial in $\mathbb{R}^3$. However, the polynomial degree of Wendland's function is 8, whereas that of Wu's function is 11.

2- While both families of strictly positive definite compactly supported functions are constructed via dimension walk, Wendland uses integration (and thus obtains a family of increasingly smoother functions), whereas Wu needs to start with a function of sufficient smoothness, and then obtains successively less smooth functions (via differentiation).

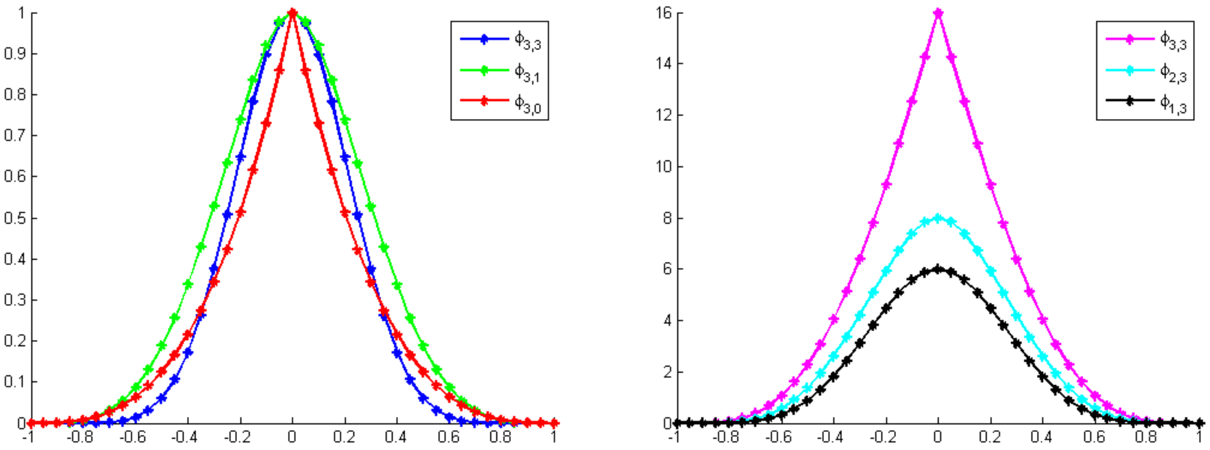

Figure 2.1: Graph of Wendland's CSRBFs (left), Wu's CSRBFs (right), for a center $x = 0$.

The figure 2.1 shows some of Wendland's and Wu's CSRBFs.

Buhmann's Compactly Supported Functions

A third family of compactly supported strictly positive definite radial functions that has been appeared in the literature is due to Buhmann (see [12]). Buhmann's functions contain a logarithmic term in addition to a polynomial. His functions have the general form

$$\phi(r) = \int_0^{+\infty} (1 - r^2/t)_+^{\lambda} t^{\alpha} (1 - t^{\delta})_+^{\rho} dt.$$

Here $0 < \delta \leq \dfrac{1}{2}$, $\rho \geq 1$, and in order to obtain functions that are strictly positive definite and radial on $\mathbb{R}^s$ for $s \leq 3$ the constraints for the remaining parameters are $\lambda \geq 0$, and $-1 < \alpha \leq \dfrac{\lambda - 1}{2}$.

Example 2.4. An example with $\alpha = \delta = \frac{1}{2}$, $\rho = 1$ and $\lambda = 2$ is listed in [13]: $\phi(r) \doteq 12r^4 log(r) - 21r^4 + 32r^3 - 12r^2 + 1$, $0 \leq r \leq 1$, $C^2 \cap SPD(\mathbb{R}^3)$, which is presented in figure 2.2.

Example 2.5. An example with $\alpha = 0$, $\delta = \frac{1}{2}$, $\rho = 4$ and $\lambda = 1$ is listed in [13] $\phi(r) \doteq \frac{1}{15} + \frac{19}{6}r^2 - \frac{16}{3}r^3 + 3r^4 - \frac{16}{15}r^5 + \frac{1}{6}r^6 + 2r^2 log(r)$, $0 \leq r \leq 1$, $C^2 \cap SPD(\mathbb{R}^3)$, which is presented in figure 2.3.

Remark 2.2. (1) While Buhmann [13] claims that his construction encompasses both Wendland's and Wu's functions, Wendland [86] gives an even more general theorem that shows that integration of a positive function $f \in L^1[0, \infty)$ against a strictly positive definite (compactly supported) kernel K results in a (compactly supported) strictly positive definite function, i.e.,

$$\phi(r) = \int_0^{+\infty} K(t, r) f(t) dt.$$

is strictly positive definite. Buhmann's construction then corresponds to choosing $f(t) = t^{\alpha}(1 - t^{\delta})_+^{\rho}$ and $K(t, r) = (1 - r^2/t)_+^{\lambda}$.

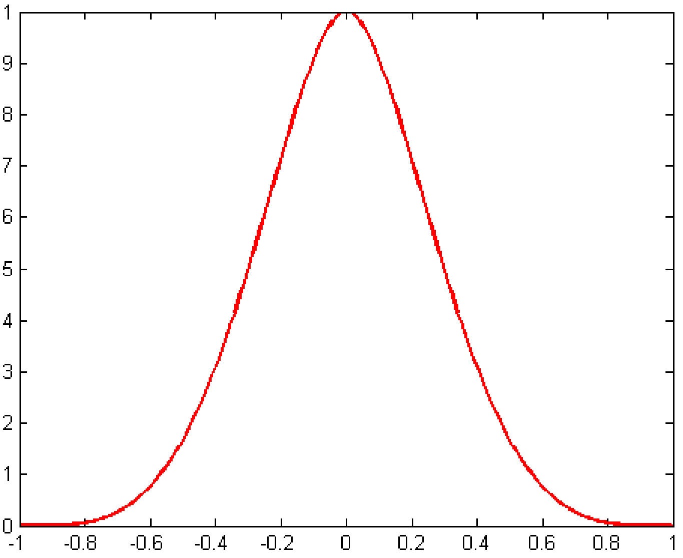

Figure 2.2: Graph of Buhmann's function of example 2.4.

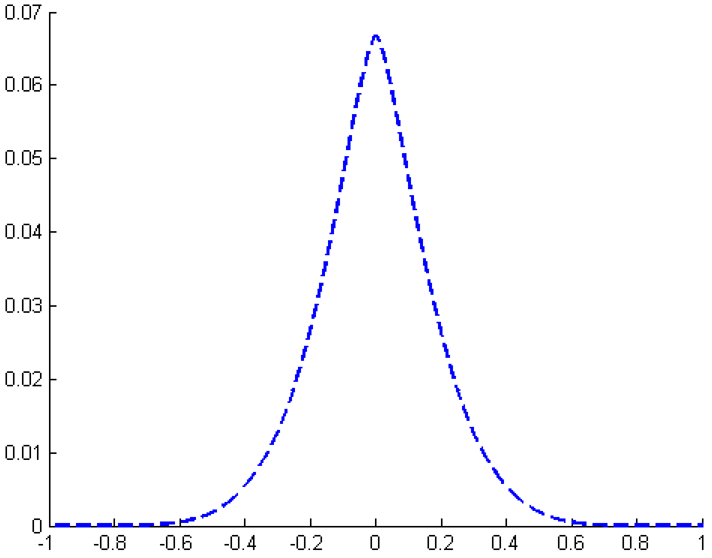

Figure 2.3: Graph of Buhmann's function of example 2.5.

(2) Multiply monotone functions are covered by this general theorem by taking $K(t, r) = (1-r^t)_+^{k-1}$ and f an arbitrary positive function in L^1 so that $d\mu(t) = f(t)dt$ in Williamson's characterization. Also, functions that are strictly positive definite and radial in $\mathbb{R}^s$ for all s (or equivalently completely monotone functions) are covered by choosing $K(t, r) = e^{-rt}$.

The approximation of a function U can be expended in serie's by compactly supported radial basis function ϕ as follow

$$U(x) \simeq \sum_{i=1}^{N} \alpha_i \phi\bigg(\frac{||x - x_i||}{\sigma}\bigg), \ x \in \mathbb{R}^s, \tag{2.3}$$

where x_i, $i = 1, \dots, N$ is a finite set of distinct points (centers) in $\mathbb{R}^s$, and σ is a positive integer. The coefficients α_i are calculated by using some collocation points.

2.1.1 Native space

There is a natural space in which consider the RBF approximation. In fact, for each positive definite and symmetric kernel Φ and for each region $\Omega \subset \mathbb{R}^s$ it is possible to define an associated real Hilbert space, the so-called Native space $\mathcal{N}_\Phi(\Omega)$.

Hilbert space is a space of functions real or complex, which is complete metric space w.r.t. the distance induced by the inner product.

Here the inner product between two functions f and g is thought as $(f, g) = \int_a^b f(x)g(x)dx$, in the real case or $(f, g) = \int_a^b f(x)\overline{g(x)}dx$, in the complex case, which has many of the familiar properties of the Euclidean (discrete) dot product. Examples of Hilbert spaces are: any finite dimensional inner product space (for example $\mathbb{R}^s$; $\mathbb{C}^s$ equipped with the dot product of two vectors), the Lebesgue spaces L^p, Sobolev spaces.

Definition 2.3. [84](p.134) Let $\mathcal{H}$ be a real Hilbert space of functions $f : \Omega \longrightarrow \mathbb{R}$, with inner product $(.,.)_\mathcal{H}$. A function $\Phi : \Omega \times \Omega \longrightarrow \mathbb{R}$ is called a reproducing kernel for $\mathcal{H}$ if

1- $\Phi(., y) \in \mathcal{H} \ \forall y \in \Omega$,

2- $f(y) = (f, \Phi(., y))_\mathcal{H} \ \forall f \in \mathcal{H}$, and $\forall y \in \Omega$. (reproducing property)

The reproducing kernel of a Hilbert space is uniquely determined. Suppose there are two reproducing kernels Φ_1 and Φ_2. Then property (2) gives $(f, \Phi_1(., y) - \Phi_2(., y))_\mathcal{H} = 0$ for all $f \in \mathcal{H}$ and all $y \in \Omega$. Setting $f = \phi_1(., y) - \Phi_2(., y)$ for a fixed y shows the uniqueness.

Theorem 2.4. *[84](p.135) Suppose that $\mathcal{H}$ is a reproducing-kernel Hilbert function space with reproducing kernel $\Phi : \Omega \times \Omega \to \mathbb{R}$. Then Φ is positive semi-definite. Moreover, Φ is positive definite if and only if the point evaluation functionals are linearly independent in $\mathcal{H}^*$.*

From the first property of definition 2.3 we know that $\mathcal{H}$ contains all functions of the form $f = \sum_{j=1}^{N} \alpha_j \Phi(., x_j)$ if $x_j \in \Omega$. Furthermore, we know that

$$\| f \|_\mathcal{H}^2 = (f, f) = \sum_{j=1}^{N}\sum_{i=1}^{N} \alpha_j \alpha_i (\Phi(., x_j), \Phi(., x_i))_\mathcal{H} = \sum_{j=1}^{N}\sum_{i=1}^{N} \alpha_j \alpha_i \Phi(x_j, x_i).$$

We will use this feature to construct a reproducing-kernel Hilbert space for a given positive definite kernel.

Hence, assume that $\Phi : \Omega \times \Omega \longrightarrow \mathbb{R}$ is a symmetric positive definite kernel. Define the $\mathbb{R}$-linear space $\mathcal{H}_\Phi(\Omega) := span\,\{\Phi(.,y) : y \in \Omega\}$, and equip it with the bilinear form

$$\left(\sum_{j=1}^{N} \alpha_j \Phi(.,x_j), \sum_{i=1}^{M} \alpha_i \Phi(.,y_i)\right)_\Phi := \sum_{j=1}^{N}\sum_{i=1}^{M} \alpha_j \alpha_i \Phi(x_j, y_i).$$

Theorem 2.5. *[84](p.137) If $\Phi : \Omega \times \Omega \to \mathbb{R}$ is a symmetric positive definite kernel then $(.,.)_\Phi$ defines an inner product on $\mathcal{H}_\Phi(\Omega)$. Furthermore, $\mathcal{H}_\Phi(\Omega)$ is a pre-Hilbert space with reproducing kernel Φ.*

Lemma 2.1. *[84](p.138) The linear mapping $R : \mathcal{H}_\Phi(\Omega) \to C(\Omega)$, $R(f)(x) := (f, \Phi(.,x))_\Phi$ is injective.*

Definition 2.4. [84](p.138) The native Hilbert function space corresponding to the symmetric positive definite kernel $\Phi : \Omega \times \Omega \to \mathbb{R}$ is defined by

$$\mathcal{N}_\Phi(\Omega) := R(\mathcal{H}_\Phi(\Omega)).$$

It carries the inner product

$$(f,g)_{\mathcal{N}_\Phi(\Omega)} := (R^{-1}f, R^{-1}g)_\Phi.$$

Indeed, the space so defined is a Hilbert space of continuous functions on Ω with reproducing kernel Φ. Since $\Phi(.,x)$ is an element of $\mathcal{H}_\Phi(\Omega)$ for $x \in \Omega$ it is unchanged under R and hence

$$f(x) = (R^{-1}f, \Phi(.,x))_\Phi = (f, \phi(.,x))_{\mathcal{N}_\Phi(\Omega)},$$

for all $f \in \mathcal{N}_\Phi(\Omega)$ and $x \in \Omega$.

Theorem 2.6. *[84](p.138) Suppose that $\Phi : \Omega \times \Omega \to \mathbb{R}$ is a symmetric positive definite kernel then its associated native space $\mathcal{N}_\Phi(\Omega)$, is a Hilbert function space with reproducing kernel Φ.*

The Native space $\mathcal{N}_\Phi(\Omega)$ for the kernel Φ is the completion of $\mathcal{H}_\Phi(\Omega)$ with respect to the $\| \,.\, \|_\Phi$-norm, so that $\| f \|_\Phi = \| f \|_{\mathcal{N}_\Phi(\Omega)}$, $\forall f \in \mathcal{H}_\Phi(\Omega)$.

2.1.2 Error bounds and stability estimates

We recall that, for a subset $\Omega \subset \mathbb{R}^s$, a discrete data-sites set $X \subset \Omega$ and a radial, positive definite kernel $\Phi \in \mathcal{C}(\Omega \times \Omega)$ the *RBF* interpolant $P_X[f]$ to a function $f \in \mathcal{N}_\Phi(\Omega)$ is computed as

$$P_X[f](x) = \sum_{j=1}^{N} c_j \Phi(x, x_j),\ \ P_X[f](x_i) = f(x_i)\ \forall x \in \Omega, x_i \in X.$$

The question is how well $P_X[f]$ can approximate the sampled function f, i.e. if $P_X[f]$ converges to f in some given norm when the data-sites X becomes dense in Ω.

There are two quantities used to relate the set X to these requirements: the fill distance

$$h_{X,\Omega} = \max_{x\in\Omega} \min_{x_i \in X} \| x - x_i \|_2,$$

and the separation distance

$$q_X = \frac{1}{2} \min_{x_i, x_j \in X} \parallel x_i - x_j \parallel_2 .$$

Clearly the shape parameter ε have also an important role, since it determines the radial amplitude of the kernel. The first estimate comes directly from the definition of the pointwise-error functional. Let ε_x be defined for all $x \in \Omega$ as

$$\varepsilon_x : \mathcal{N}_\Phi(\Omega) \to \mathbb{R}, \quad \varepsilon_x(f) = f(x) - P_X[f](x),$$

and let $\mathcal{P}_{\Phi,X}$ denote its norm, the so-called Power function. Then the basic estimate for the convergence is the following

Theorem 2.7. *[29](p.171) Let $\Omega \subset \mathbb{R}^s$, let $\Phi \in C(\Omega \times \Omega)$ be a strictly positive definite kernel, let $X \subset \Omega$ be a discrete set of data sites and let $f \in \mathcal{N}_\Phi(\Omega)$, and denote with $P_X[f]$ its interpolant on X. Then*

$$\mid f(x) - P_X[f](x) \mid \leqslant \mathcal{P}_{\Phi,X} \parallel f \parallel_{\mathcal{N}_\Phi(\Omega)}, \; \forall x \in \Omega.$$

Moreover, the Power function can be exactly computed introducing a Lagrange basis for $\mathcal{N}_\Phi(X)$.

Definition 2.5. [84](p.28) A set $\Omega \subset \mathbb{R}^s$ is said to satisfy an interior cone condition if there exists an angle $\theta \in (0, \pi/2)$ and a radius $r > 0$ such that for every $x \in \Omega$ a unit vector $\xi(x)$ exists such that the cone

$$C(x, \xi(x), \theta, r) := \left\{ x + \lambda y : y \in \mathbb{R}^s, \parallel y \parallel_2 = 1, y^T \xi(x) \geq cos\theta, \lambda \in [0, r] \right\}.$$

is contained in Ω.

Theorem 2.8. *[29](p.121) Let $\Omega \subset \mathbb{R}^s$ be a bounded set that satisfies an interior cone condition and let $\Phi \in C^{2k}(\Omega \times \Omega)$ be a symmetric positive definite. Then there exist positive constants h_0 and C independent of x, f and Φ, such that for all $X \subset \Omega$, for all $h_{X,\Omega} \leqslant h_0$, for all $f \in \mathcal{N}_\Phi(\Omega)$ and for all $x \in \Omega$*

$$\mid f(x) - P_X[f](x) \mid \leqslant C h_{X,\Omega}^k \sqrt{C_\Phi(x)} \parallel f \parallel_{\mathcal{N}_\Phi(\Omega)},$$

where $C_\Phi(x)$ is a given as follow

$$C_\Phi(x) = \max_{|\beta|=2k} \max_{\omega, z \in \Omega \cup B(x, c_2 h_{X,\Omega})} \cap D_2^\beta \Phi(\omega, z) \mid .$$

with $B(x, c_2 h_{X,\Omega})$ denoting the ball of radius $c_2 h_{X,\Omega}$ centered at x.

From the previous estimate we can expect that the approximation error goes to zero as $h_{X,\Omega} \longrightarrow 0$. When the data-sites set X becomes too big the interpolation can be instable. In fact, it is possible to prove that the condition number of the kernel matrix A grows if the separation distance q_X decreases, and this, together with a bad choice of the shape parameter ε, can produce very instable approximants. Various approaches are used to avoid this situation. A lot of efforts are made on the study of well-distributed data-sites set, for examples sets X such that the uniformity

$$\rho_{X,\Omega} = \frac{q_X}{h_{X,\Omega}},$$

is maximized. Another common way to try to avoid instability, and more related on the linear algebra part of the method, is to choose a shape parameter ε such that the kernel matrix is not ill-conditioned.

we assume the general finite-dimensional subspace to be $\pi_{m-1}(\mathbb{R}^s)$. In the following, the region $\Omega \subseteq \mathbb{R}^d$ is always assumed to be open. But this is only necessary for estimates on the derivatives. In the non-derivative case Ω has only to satisfy an interior cone condition. Moreover, if Ω is not open, the estimates on the derivatives hold in every interior point.

Theorem 2.9. *[84](p.183) Let Φ be one of the gaussians or the (inverse) multiquadrics. Suppose that Φ is conditionally positive definite of order m. Suppose further that $\Omega \subseteq \mathbb{R}^s$ is bounded and satisfies an interior cone condition. Denote the radial basis function interpolant to $f \in \mathcal{N}_\Phi(\Omega)$ based on Φ and $X = \{x_1, \ldots, x_N\}$ by $P_X[f]$. Fix $\alpha \in \mathbb{N}_0^s$. For every $l \in \mathbb{N}$ with $l \geq max\{| \alpha |, m-1\}$ there exist constants $h_0(l)$, $C_l > 0$ such that*

$$| D^\alpha f(x) - D^\alpha P_X[f](x) | \leq C_l h_{X,\Omega}^{l-|\alpha|} \mid f \mid_{\mathcal{N}_\phi(\Omega)} .$$

for all $x \in \Omega$, provided that $h_{X,\Omega} \leq h_0(l)$.

Theorem 2.10. *[84](p.184) Suppose that $\Omega \subseteq \mathbb{R}^s$ is bounded and satisfies an interior cone condition. Let $\Phi(x) = (-1)^{\lceil \beta/2 \rceil} \| x \|_2^\beta$, $\beta > 0$, $\beta \notin 2\mathbb{N}$. Denote the interpolant of a function $f \in \mathcal{N}_\Phi(\Omega)$ based on this basis function and the set of centers $X = \{x_1, \ldots, x_N\} \subseteq \Omega$ by $P_X[f]$. Then there exist constants h_0, $C > 0$ such that*

$$| D^\alpha f(x) - D^\alpha P_X[f](x) | \leq C h_{X,\Omega}^{\beta/2-|\alpha|} \mid f \mid_{\mathcal{N}_\Phi(\Omega)},$$

for all $x \in \Omega$ and all α with $| \alpha | \leq (\lceil \beta \rceil - 1) - 2$, provided that $h_{X,\Omega} \leq h_0$.

Theorem 2.11. *[84](p.184)*

Let $\Phi_{s,k}$ be a compactly supported radial basis functions . Suppose that $\Omega \subseteq \mathbb{R}^s$ is bounded and satisfies an interior cone condition. Denote the radial basis function interpolant of $f \in \mathcal{N}_{\Phi_{s,k}}(\Omega)$ based on $\Phi_{s,k}$ and $X = \{x_1, \ldots, x_N\} \subseteq \Omega$ by $P_X[f]$. Then there exist constants C, $h_0 > 0$ such that

$$| D^\alpha f(x) - D^\alpha P_X[f](x) | \leq C h_{X,\Omega}^{k+1/2-|\alpha|} \| f \|_{\mathcal{N}_\Phi(\Omega)}$$

for every $\alpha \in \mathbb{N}_0^s$ with $| \alpha | \leq k$ and every $x \in \Omega$, provided that $h_{X,\Omega} \leq h_0$.

2.2 Application of RBFs and CSRBFs for solving integral equations

Over the years, integral equations have motivated a large amount of research works. Integral equations have been the best way to formulate physics, mechanics, fluid, elastisity, radiation science and other fields problems. Moreover the numerical integral gives smaller relative errors than the numerical differentiation. Currently different numerical methods for finiding an approximate solution of integral equations were proposed, such as, collocation method [8], Wavelet-Galerkin sheme for solving Volterra integral equations of the second kind [70]. A lot of researchers used multiquadric, gaussian and inverse multiquadric radial basis functions for solving integral equations [45, 67].

2.2.1 CSRBF for solving nonlinear functional Volterra-Fredholm integral equations

The aim is to give the approximate solution of Volterra-Fredholm integral equations defined as

$$A(t)U(t)+B(t)U(h(t))+\lambda_1\int_0^1 W_1(s,t,U(h(s)))ds+\lambda_2\int_0^{h(t)} W_2(s,t,U(s))ds=f(t), \quad (2.4)$$

where $0\leq t\leq 1$, $f(t)$, $h(t)$ are known functions, and λ_1, λ_2 are real values, with

$$W_1(s,t,U(h(s)))=k_1(s,t)\,[U(h(s))]^{r_1}\,,$$

$$W_2(s,t,U(s))=k_2(s,t)\,[U(s)]^{r_2}\,,$$

such that r_1, r_2 are positive integers.

For our experiment we are going to use Wendland's and Wu's compactly supported radial basis functions $\phi^{\sigma}_{4,2}$ and $\phi^{\sigma}_{2,1}$ respectively.

To approximate the function $U(t)$ of equation (2.4), we apply RBF interpolation in distinct grids from a definite domain. To this purpose, the function $U(t)$ is approximated by a linear combination of a functions ϕ_j as follows

$$U(t)\approx\sum_{j=1}^{N}c_j\phi\left(\frac{\parallel t-t_j\parallel}{\sigma}\right), \quad (2.5)$$

where ϕ_j is Wu or Wendland's compactly supported radial basis functions $\phi^{\sigma}_{2,1}$ and $\phi^{\sigma}_{4,2}$ respectively. According to type of the target function which is desired to approximate, and similarly it is true for

$$U(h(t))\approx\sum_{j=1}^{N}c_j\phi\left(\frac{\parallel h(t)-t_j\parallel}{\sigma}\right). \quad (2.6)$$

The integrals have been approximated using the following integration formula

$$\lambda_1\int_0^1 W_1(s,t,U(h(s)))ds=\lambda_1\sum_{k=1}^{N}w_kW_1\left(s_k,t,\sum_{j=1}^{N}c_j\phi_j\left(\frac{h(s_k)}{\sigma}\right)\right), \quad (2.7)$$

where, s_k and w_k are shifted Legendre-Gauss-Lobatto nodes and weights .

By replacing equation (2.6) and (2.7) in equation (2.4), and by collocating at points $t=t_i$, such that

$$t_i=\frac{1}{2}-\frac{1}{2}cos\left(\frac{(i-1)\Pi}{N}\right),i=1\ldots N,$$

we obtain

$$\begin{aligned}&A(t_i)\sum_{j=1}^{N}c_j\phi\left(\frac{\parallel t_i-t_j\parallel}{\sigma}\right)+B(t_i)\sum_{j=1}^{N}c_j\phi\left(\frac{\parallel h(t_i)-t_j\parallel}{\sigma}\right)\\&+\lambda_1\sum_{k=1}^{N}w_kW_1\left(s_k,t_i,\sum_{j=1}^{N}c_j\phi\left(\frac{\parallel h(s_k)-s_j\parallel}{\sigma}\right)\right)\\&+\lambda_2\int_0^{h(t_i)}W_2\left(s,t_i,\sum_{j=1}^{N}c_j\phi\left(\frac{\parallel s-s_j\parallel}{\sigma}\right)\right)ds=f(t_i),\end{aligned}$$

the interval $[0, h(t_i)]$ has been transformed to $[0,1]$, by taking the changement $\mu = \frac{s}{h(t_i)}$, then, we get

$$
\begin{aligned}
&A(t_i)\sum_{j=1}^{n} c_j\varphi\bigg(\frac{\parallel t_i - t_j \parallel}{\sigma}\bigg) + B(t_i)\sum_{j=1}^{n} c_j\varphi\bigg(\frac{\parallel h(t_i) - t_j \parallel}{\sigma}\bigg)\\
&+\lambda_1\sum_{k=1}^{n} w_k W_1\bigg(s_k, t_i, \sum_{j=1}^{n} c_j\varphi\bigg(\frac{\parallel h(s_k) - s_j \parallel}{\sigma}\bigg)\bigg)\\
&+\lambda_2.h(t_i)\sum_{p=1}^{n} w_p W_2\bigg(h(t_i).\mu_p, t_i, \sum_{j=1}^{n} c_j\varphi\bigg(\frac{\parallel (h(t_i).\mu_p) - s_j \parallel}{\sigma}\bigg)\bigg) = f(t_i).
\end{aligned}
$$

This is a nonlinear system of equations, that can be solved via Newton's iteration method to obtain the unknwon vector C, such that $C = (c_1, c_2, c_3, ..., c_N)^T$.

Let $U_N(t)$ be the approximate solution of $U(t)$, in order to test the efficiency and the convergence accuracy of the proposed method, we are going to calculate the absolute error at some different points using the formula

$$
e(t) = |U_N(t) - U(t)|,\ 0 \le t \le 1, \tag{2.8}
$$

for $N = 7, 14$, and different values of the parameter σ.

Example 2.6. Let given the following functional Volterra-Fredholm integral equation

$$
U(t) + U(h(t)) = f(t) + \lambda_1\int_0^{h(t)}(t - \frac{s}{2})U(s)ds + \lambda_2\int_0^1 tsU(h(s))ds, \tag{2.9}
$$

such that $h(t) = \frac{t}{2}$, $\lambda_1 = \lambda_2 = 1$, and $f(t) = \frac{-1}{112}t + \frac{33}{16}t^5 - \frac{11}{2688}t^7$.

The exact solution is $U(t) = 2t^5$.

The numerical results are sitting in tables 2.1- 2.2 for Wendland's CSRBF $\phi_{4,2}$, and in tables 2.3-2.4 for Wu's CSRBF $\phi_{2,1}$.

Example 2.7. Consider the following nonlinear Fredholm integral equation,

$$
U(t) = e^{-3t} - t\bigg(-\frac{101}{729}e^{-9} + \frac{2}{729}\bigg) + \int_0^1 ts^2U^3(s)ds,
$$

where $\lambda_1 = 1$, $\lambda_2 = 0$. The exact solution is $y(t) = e^{-3t}$.

The numerical results are presented in tables 2.5- 2.6 for Wendland's CSRBF $\phi_{4,2}$, and in tables 2.7-2.8 for Wu's CSRBF $\phi_{2,1}$.

Example 2.8. Let given the following Volterra integral equation

$$
U(t) = e^t sin(t) + \lambda_2\int_0^{h(t)} cos(t)(t - s)U(s)ds, \tag{2.10}
$$

where $h(t) = t$, $\lambda_1 = 0$, $\lambda_2 = 2$. The exact solution is $U(t) = te^t$.

The numerical results are given in tables 2.9- 2.10 for Wendland's CSRBF $\phi_{4,2}$, and in tables 2.11-2.12 for Wu's CSRBF $\phi_{2,1}$.

Remark 2.3. From the numerical tests that we were able to establish, we conclude that:

- Both Wu and Wendland's CSRBFs method give reasonable accuracy.
- The accuracy of CSRBFs method depends on the choice of the value of parameter σ, and Number of data sets N.
- Wendland's CSRBF method gives better results compared to the use of Wu's CSRBFs.

Points	$\sigma = 12$	$\sigma = 20$
x_i		
0	$1.3336E-7$	$9.7799E-6$
0.01	$2.0450E-5$	$3.0494E-5$
0.02	$4.4889E-5$	$5.3677E-5$
0.03	$7.2258E-5$	$7.8765E-5$
0.04	$1.0158E-4$	$1.0516E-4$
0.05	$1.3178E-4$	$1.3216E-4$
0.06	$1.6171E-4$	$1.5901E-4$
0.07	$1.9024E-4$	$1.8496E-4$
0.08	$2.1632E-4$	$2.0932E-4$
0.09	$2.3918E-4$	$2.3157E-4$
0.1	$2.5834E-4$	$2.5149E-4$
0.2	$5.6826E-4$	$6.0851E-4$
0.3	$2.5188E-3$	$2.3858E-3$
0.4	$1.6431E-4$	$2.8639E-4$
0.5	$9.5399E-3$	$8.2185E-3$
0.6	$2.4712E-3$	$1.9396E-3$
0.7	$1.6906E-2$	$1.5158E-2$
0.8	$6.5934E-3$	$6.2441E-3$
0.9	$2.5450E-3$	$2.8301E-3$
1	$1.5437E-2$	$1.4351E-2$

Table 2.1: Computed errors using Wendland's CSRBF $\phi_{4,2}$ method for Example (2.9), for $N = 7$.

Points	$\sigma = 12$	$\sigma = 20$
x_i		
0	$1.8625E-5$	$3.7760E-5$
0.01	$1.5435E-5$	$1.4017E-5$
0.02	$2.2983E-5$	$1.1013E-5$
0.03	$4.0231E-5$	$2.7886E-5$
0.04	$6.5661E-5$	$6.2923E-5$
0.05	$9.6718E-5$	$1.1131E-4$
0.06	$1.2931E-4$	$1.6438E-4$
0.07	$1.5826E-4$	$2.0996E-4$
0.08	$1.7841E-4$	$2.3588E-4$
0.09	$1.8625E-4$	$2.3269E-4$
0.1	$1.7988E-4$	$1.9724E-4$
0.2	$3.8252E-4$	$4.4853E-4$
0.3	$6.3846E-4$	$7.0832E-4$
0.4	$4.6383E-4$	$4.8197E-4$
0.5	$6.8489E-4$	$3.6963E-4$
0.6	$2.2853E-3$	$2.0447E-3$
0.7	$1.7197E-3$	$1.2318E-3$
0.8	$3.8206E-3$	$3.5673E-3$
0.9	$4.1741E-3$	$4.0646E-3$
1	$3.1012E-3$	$3.4307E-3$

Table 2.2: Computed errors using Wendland's CSRBF $\phi_{4,2}$ method for Example (2.9), for $N = 14$.

Points	$\sigma = 18$	$\sigma = 25$
x_i		
0	$1.2144E-8$	$1.9684E-7$
0.01	$2.2766E-5$	$1.6849E-5$
0.02	$3.8376E-5$	$3.0117E-5$
0.03	$4.6940E-5$	$3.9676E-5$
0.04	$4.8609E-5$	$4.5643E-5$
0.05	$4.3910E-5$	$4.8355E-5$
0.06	$3.4797E-5$	$4.8877E-5$
0.07	$2.3698E-5$	$4.8567E-5$
0.08	$1.3187E-5$	$4.8923E-5$
0.09	$6.0044E-6$	$5.1616E-5$
0.1	$5.0844E-6$	$5.8505E-5$
0.2	$2.7116E-4$	$1.7570E-4$
0.3	$4.0221E-3$	$4.1391E-3$
0.4	$3.3376E-3$	$3.3727E-3$
0.5	$5.1256E-2$	$5.0686E-2$
0.6	$1.9943E-2$	$1.9774E-2$
0.7	$1.1169E-1$	$1.0999E-1$
0.8	$2.6012E-2$	$2.5776E-2$
0.9	$3.2195E-2$	$3.1439E-2$
1	$4.9237E-2$	$4.8795E-2$

Table 2.3: Computed errors using Wu's CSRBF $\phi_{2,1}$ method for Example (2.9), for $N = 7$.

Points	$\sigma = 18$	$\sigma = 25$
x_i		
0	$1.1709E-6$	$1.9811E-6$
0.01	$5.0099E-6$	$5.4553E-6$
0.02	$1.5824E-5$	$1.6499E-5$
0.03	$2.7610E-5$	$2.6792E-5$
0.04	$3.0697E-5$	$2.9301E-5$
0.05	$1.5942E-5$	$1.7393E-5$
0.06	$1.3436E-5$	$5.9272E-6$
0.07	$4.3234E-5$	$2.9092E-5$
0.08	$5.9007E-5$	$4.0378E-5$
0.09	$4.6476E-5$	$2.8225E-5$
0.1	$6.0242E-6$	$1.6766E-5$
0.2	$2.4493E-5$	$3.9787E-5$
0.3	$1.8029E-4$	$2.1595E-4$
0.4	$1.1989E-3$	$1.1749E-3$
0.5	$8.1861E-4$	$7.4995E-4$
0.6	$2.4034E-2$	$2.3674E-2$
0.7	$6.6337E-2$	$6.5362E-2$
0.8	$5.5429E-2$	$5.4661E-2$
0.9	$3.5231E-2$	$3.4795E-2$
1	$4.4090E-4$	$5.4714E-4$

Table 2.4: Computed errors using Wu's CSRBF $\phi_{2,1}$ method for Example (2.9), for $N = 14$.

Points	$\sigma = 6$	$\sigma = 8$
x_i		
0	$3.4853E-9$	$4.7196E-9$
0.01	$4.2460E-4$	$3.5598E-4$
0.02	$5.7369E-4$	$4.7767E-4$
0.03	$4.9506E-4$	$4.0634E-4$
0.04	$2.3606E-4$	$1.8267E-4$
0.05	$1.5639E-4$	$1.5325E-4$
0.06	$6.3598E-4$	$5.6205E-4$
0.07	$1.1578E-3$	$1.0057E-3$
0.08	$1.6791E-3$	$1.4483E-3$
0.09	$2.1608E-3$	$1.8570E-3$
0.1	$2.5678E-3$	$2.2024E-3$
0.2	$3.8020E-4$	$2.2971E-4$
0.3	$4.5009E-3$	$3.6297E-3$
0.4	$1.8742E-3$	$1.7448E-4$
0.5	$4.5845E-3$	$4.0421E-3$
0.6	$1.8849E-3$	$1.8404E-3$
0.7	$1.4575E-3$	$1.5568E-3$
0.8	$2.1769E-3$	$2.1820E-3$
0.9	$2.4101E-3$	$2.4064E-3$
1	$2.7264E-3$	$2.7264E-3$

Table 2.5: Computed errors using Wendland's CSRBF $\phi_{4,2}$ method for Example (2.10), for $N = 7$.

Points	$\sigma = 6$	$\sigma = 8$
x_i		
0	$5.2946E-8$	$1.2213E-7$
0.01	$1.9364E-5$	$2.0789E-5$
0.02	$8.7933E-5$	$8.2951E-5$
0.03	$1.5399E-4$	$1.4281E-4$
0.04	$1.7522E-4$	$1.6492E-4$
0.05	$1.3136E-4$	$1.3240E-4$
0.06	$3.5017E-5$	$5.5911E-5$
0.07	$7.1606E-5$	$2.9133E-5$
0.08	$1.3322E-4$	$7.6392E-5$
0.09	$9.9559E-5$	$4.3751E-5$
0.1	$5.6600E-5$	$9.1566E-5$
0.2	$5.9349E-5$	$3.7596E-5$
0.3	$1.7179E-3$	$1.5725E-3$
0.4	$6.6511E-4$	$7.3367E-4$
0.5	$1.3632E-3$	$1.3637E-3$
0.6	$1.7179E-3$	$1.7039E-3$
0.7	$1.8651E-3$	$1.8707E-3$
0.8	$2.1971E-3$	$2.1953E-3$
0.9	$2.4589E-3$	$2.4584E-3$
1	$2.7264E-3$	$2.7259E-3$

Table 2.6: Computed errors using Wendland's CSRBF $\phi_{4,2}$ method for Example (2.10), for $N = 14$.

Points	$\sigma = 6$	$\sigma = 10$
x_i		
0	$6.2578E - 10$	$2.9367E - 9$
0.01	$2.0489E - 3$	$1.8714E - 3$
0.02	$3.0189E - 3$	$2.7538E - 3$
0.03	$2.9354E - 3$	$2.6730E - 3$
0.04	$1.8232E - 3$	$1.6539E - 3$
0.05	$2.5485E - 3$	$2.4440E - 4$
0.06	$3.0694E - 3$	$2.8123E - 3$
0.07	$6.3469E - 3$	$5.8008E - 3$
0.08	$9.8146E - 3$	$8.9616E - 3$
0.09	$1.3200E - 2$	$1.2047E - 2$
0.1	$1.6233E - 2$	$1.4810E - 2$
0.2	$5.9512E - 3$	$5.3713E - 3$
0.3	$3.2131E - 2$	$2.9206E - 2$
0.4	$5.3180E - 3$	$4.9519E - 3$
0.5	$1.5465E - 2$	$1.4419E - 2$
0.6	$2.5581E - 3$	$2.5413E - 3$
0.7	$1.9852E - 4$	$5.2504E - 4$
0.8	$1.9859E - 3$	$1.9015E - 3$
0.9	$2.8439E - 3$	$3.0586E - 3$
1	$2.7264E - 3$	$2.7263E - 3$

Table 2.7: Computed errors using Wu's CSRBF $\phi_{2,1}$ method for Example (2.10), for $N = 7$.

Points	$\sigma = 6$	$\sigma = 10$
x_i		
0	$1.8537E - 9$	$5.0199E - 8$
0.01	$1.1187E - 5$	$9.9801E - 5$
0.02	$8.0814E - 5$	$7.4217E - 4$
0.03	$1.9631E - 3$	$1.7981E - 3$
0.04	$2.0847E - 3$	$1.9113E - 3$
0.05	$2.4065E - 6$	$9.9642E - 6$
0.06	$4.0684E - 3$	$3.6967E - 3$
0.07	$8.4237E - 3$	$7.6672E - 3$
0.08	$1.1418E - 2$	$1.0396E - 2$
0.09	$1.1380E - 2$	$1.0359E - 2$
0.1	$6.9198E - 3$	$6.2882E - 3$
0.2	$1.5871E - 2$	$1.4429E - 2$
0.3	$2.0981E - 2$	$1.9209E - 2$
0.4	$6.2034E - 3$	$5.5557E - 3$
0.5	$1.3632E - 3$	$1.3632E - 3$
0.6	$1.9493E - 3$	$1.8113E - 3$
0.7	$2.2107E - 3$	$2.4891E - 3$
0.8	$1.8817E - 3$	$1.6598E - 3$
0.9	$2.3218E - 3$	$2.2246E - 3$
1	$2.7263E - 3$	$2.7264E - 3$

Table 2.8: Computed errors using Wu's CSRBF $\phi_{2,1}$ method for Example (2.10), for $N = 14$.

Points	$\sigma = 6$	$\sigma = 9$
x_i		
0	$9.94966E - 10$	$4.7917E - 9$
0.01	$3.4569E - 5$	$3.3932E - 5$
0.02	$4.5864E - 5$	$4.4984E - 5$
0.03	$3.5040E - 5$	$3.4244E - 5$
0.04	$3.2911E - 6$	$2.8273E - 6$
0.05	$4.8114E - 5$	$4.8119E - 5$
0.06	$1.1799E - 4$	$1.1743E - 4$
0.07	$2.0505E - 4$	$2.0400E - 4$
0.08	$3.0823E - 4$	$3.0685E - 4$
0.09	$4.2667E - 4$	$4.2520E - 4$
0.1	$5.5987E - 4$	$5.5858E - 4$
0.2	$2.9718E - 3$	$2.9672E - 3$
0.3	$9.8215E - 3$	$9.6936E - 3$
0.4	$2.3542E - 2$	$2.3600E - 2$
0.5	$4.3521E - 2$	$4.4287E - 2$
0.6	$6.9306E - 2$	$6.9504E - 2$
0.7	$9.7314E - 2$	$9.6099E - 2$
0.8	$1.1796E - 1$	$1.1777E - 1$
0.9	$1.1941E - 1$	$1.1994E - 1$
1	$8.2429E - 2$	$8.2438E - 2$

Table 2.9: Computed errors using Wendland's CSRBF $\phi_{4,2}$ method for Example (2.11), for $N = 7$.

Points	$\sigma = 6$	$\sigma = 9$
x_i		
0	$6.1491E-10$	$3.2441E-8$
0.01	$2.5618E-7$	$1.1894E-7$
0.02	$4.0614E-6$	$4.1335E-6$
0.03	$1.1476E-5$	$1.1452E-5$
0.04	$2.1032E-5$	$2.1059E-5$
0.05	$3.3088E-5$	$3.3406E-5$
0.06	$4.9999E-5$	$5.0808E-5$
0.07	$7.5929E-5$	$7.7263E-5$
0.08	$1.1584E-4$	$1.1752E-4$
0.09	$1.7440E-4$	$1.7602E-4$
0.1	$2.5486E-4$	$2.5593E-4$
0.2	$2.0939E-3$	$2.0963E-3$
0.3	$7.0567E-3$	$7.0488E-3$
0.4	$1.5711E-2$	$1.5726E-2$
0.5	$2.8172E-2$	$2.8172E-2$
0.6	$4.2456E-2$	$4.2375E-2$
0.7	$5.4086E-2$	$5.4257E-2$
0.8	$5.5723E-2$	$5.5609E-2$
0.9	$3.4914E-2$	$3.4872E-2$
1	$2.5701E-2$	$2.5770E-2$

Table 2.10: Computed errors using Wendland's CSRBF $\phi_{4,2}$ method for Example (2.11), for $N = 14$.

Points	$\sigma = 2$	$\sigma = 4$
x_i		
0	$1.0811E-11$	$2.5678E-10$
0.01	$4.7875E-4$	$1.0298E-4$
0.02	$7.1202E-4$	$1.6154E-4$
0.03	$6.9750E-4$	$1.7233E-4$
0.04	$4.3317E-4$	$1.3192E-4$
0.05	$7.2809E-5$	$4.0360E-5$
0.06	$7.6924E-4$	$8.7248E-5$
0.07	$1.5932E-3$	$2.3184E-4$
0.08	$2.4815E-3$	$3.7426E-4$
0.09	$3.3709E-3$	$4.9567E-4$
0.1	$4.1975E-3$	$5.7713E-4$
0.2	$1.6402E-3$	$3.4429E-3$
0.3	$5.1166E-3$	$1.3332E-2$
0.4	$2.2872E-2$	$2.2566E-2$
0.5	$1.8456E-2$	$3.0641E-2$
0.6	$6.0025E-2$	$6.5357E-2$
0.7	$1.6223E-1$	$1.2555E-1$
0.8	$1.3016E-1$	$1.2342E-1$
0.9	$8.7025E-2$	$1.0572E-1$
1	$8.2526E-2$	$8.2704E-2$

Table 2.11: Computed errors using Wu's CSRBF $\phi_{2,1}$ method for Example (2.11), for $N = 7$.

Points	$\sigma = 2$	$\sigma = 4$
x_i		
0	$9.2867E-12$	$2.1312E-9$
0.01	$3.0763E-5$	$6.2139E-6$
0.02	$1.7134E-4$	$3.1133E-5$
0.03	$4.3068E-4$	$7.6665E-5$
0.04	$4.6247E-4$	$7.1383E-5$
0.05	$2.3585E-5$	$4.0511E-5$
0.06	$8.9593E-4$	$2.4760E-4$
0.07	$1.8501E-3$	$4.7853E-4$
0.08	$2.4862E-3$	$6.5979E-4$
0.09	$2.4262E-3$	$7.1779E-4$
0.1	$1.3562E-3$	$5.9172E-4$
0.2	$1.5554E-3$	$2.8629E-3$
0.3	$1.1664E-2$	$6.1490E-3$
0.4	$1.3268E-2$	$1.5722E-2$
0.5	$2.8152E-2$	$2.8156E-2$
0.6	$5.5959E-2$	$4.8137E-2$
0.7	$1.7264E-2$	$3.8403E-2$
0.8	$8.5592E-2$	$6.8463E-2$
0.9	$4.8052E-2$	$4.0519E-2$
1	$2.5701E-2$	$2.5703E-2$

Table 2.12: Computed errors using Wu's CSRBF $\phi_{2,1}$ method for Example (2.11), for $N = 14$.

2.2.2 MQ-RBF for solving nonlinear Fredholm integral equations

The aim is to give the approximate solution of nonlinear Fredholm integral equation defined as

$$A(t)U(t) + B(t)U(h(t)) + \lambda \int_0^T k(s,t) \left[U(h(s))\right]^r ds = f(t), \tag{2.11}$$

where $f(t)$, $h(t)$, $A(t)$, $B(t)$, and $k(s,t)$ are known functions, λ, r are real values. To approximate the function $U(t)$, we apply RBF interpolation in distinct grids from a definite domain. To this purpose, a linear combination of functions ϕ_j is replaced in $U(t)$ as follows

$$U(t) \approx U_N(t) = \sum_{j=1}^{N} c_j \phi\Big(\parallel t - t_j \parallel \Big), \tag{2.12}$$

where ϕ_j is the multiquadric radial basis function defined in previous sections.

According to type of the target function which is desired to approximate, and similarly it is true for

$$U(h(t)) \approx \sum_{j=1}^{N} c_j \phi\Big(\parallel h(t) - t_j \parallel \Big). \tag{2.13}$$

The integral has been approximated using the following integration formula

$$\int_0^1 k(s,t) \left[U(h(s))\right]^r ds = \sum_{k=1}^{N} w_k k(s_k,t) \left[\sum_{j=0}^{N} c_j \phi\Big(\parallel h(s_k) - t_j \parallel \Big) \right]^r, \tag{2.14}$$

where, s_k, w_k are shifted Legendre-Gauss-Lobatto nodes and weights .

By replacing equation (2.14) in equation (2.11), and by collocating at points $t = t_i$, we obtain

$$A(t_i) \sum_{j=0}^{N} c_j \phi\Big(\parallel t_i - t_j \parallel \Big) + B(t_i) \sum_{j=1}^{N} c_j \phi\Big(\parallel h(t_i) - t_j \parallel \Big) + \lambda \int_0^T k(s,t_i) \left[\sum_{j=1}^{N} c_j \phi\Big(\parallel h(s) - t_j \parallel \Big) \right]^r ds = f(t_i).$$

The interval $[0,T]$ has been transformed to $[0,1]$, by taking the changement $\mu = \frac{s}{T}$.

$$\begin{aligned} & A(t_i) \sum_{j=1}^{N} c_j \phi\Big(\parallel t_i - t_j \parallel \Big) + B(t_i) \sum_{j=1}^{N} c_j \phi\Big(\parallel h(t_i) - t_j \parallel \Big) \\ & + \lambda . T \sum_{k=1}^{N} w_k k(T.\mu_k, t_i) \left[\sum_{j=0}^{N} c_j \phi\Big(\parallel h(T.\mu_k) - t_j \parallel \Big) \right]^r = f(t_i). \end{aligned} \tag{2.15}$$

From equation (2.15), we get a nonlinear system which can be solved via iteration methods to get the unknwon vector C, such that $C = (c_1, c_2, ..., c_N)^T$.
Finaly, the meshless discrete solution using RBFs is given by

$$U_N(t) = \sum_{j=1}^{N} c_j \phi_j(t).$$

Let $U_N(t)$ be the approximate solution of $U(t)$, we are going to calculate the absolute errors at some different points using the formula (2.8), for different shape parameters obtained using Cross Validation method.

Example 2.9. Consider the following linear Fredholm integral equation of the second kind

$$U(t)+A(t)U(h(t))+\int_0^1 k(s,t)U(s)ds=f(t), \tag{2.16}$$

where $A(t)=t$, $k(s,t)=s-t$, $h(t)=t$, $f(t)=t^3+t^2+\frac{1}{3}t-\frac{1}{4}$.
The exact solution is $U(t)=t^2$. The numerical results are sitting in table 2.13.

Example 2.10. Consider the following nonlinear Fredholm integral equation

$$U(t)=e^{-3t}-t\left(-\frac{101}{729}e^{-9}+\frac{2}{729}\right)+\int_0^1 ts^2U^3(s)ds.$$

The exact solution is given by $U(t)=e^{-3t}$. The numerical results are given in table 2.14.

Example 2.11. Let given the following linear Fredholm integral equation of the second kind

$$U(t)+A(t)U(h(t))+\int_0^{1.1} k(t,s)U(s)ds=f(t), \tag{2.17}$$

where $A(t)=t$, $k(t,s)=t-s$, $h(t)=\frac{t^2}{2}$, $f(t)=e^t+te^{t^2/2}+(t-0.1)e^{1.1}-t-1$.
The exact solution is given by $U(t)=e^t$. The numerical results are summarized in table 2.15 .

x_i	$\epsilon=0.9$	$\epsilon=2$	x_i	$\epsilon=0.9$	$\epsilon=2$
0	$1.8625E-13$	$2.7081E-8$	0.1	$2.1737E-5$	$7.8261E-7$
0.01	$9.9178E-5$	$4.2692E-6$	0.2	$3.03188E-6$	$9.4842E-8$
0.02	$1.4643E-4$	$5.9981E-6$	0.3	$5.2506E-5$	$1.28372E-6$
0.03	$1.5311E-4$	$6.0481E-6$	0.4	$4.5557E-6$	$1.1128E-7$
0.04	$1.2944E-4$	$4.9220E-6$	0.5	$7.2784E-5$	$1.4376E-6$
0.05	$8.4499E-5$	$3.0381E-6$	0.6	$5.3562E-6$	$6.3657E-8$
0.06	$2.6270E-5$	$7.4346E-7$	0.7	$7.2076E-5$	$1.4008E-6$
0.07	$3.8360E-5$	$1.6838E-6$	0.8	$4.7951E-6$	$1.1831E-7$
0.08	$1.0356E-4$	$4.0235E-6$	0.9	$3.8922E-5$	$8.4941E-7$
0.09	$1.6451E-4$	$6.1107E-6$	1	$3.2885E-10$	$7.5692E-8$

Table 2.13: Computed errors using multiquadric RBF method for Example (2.9), with $\epsilon=0.9$ (obtained using LOOCV strategy) and $\epsilon=2$, for $N=7$.

x_i	$\epsilon = 1$	$\epsilon = 1.5$	x_i	$\epsilon = 1$	$\epsilon = 1.5$
0	$2.6102E-11$	$4.1822E-8$	0.1	$4.4348E-4$	$3.2048E-4$
0.01	$6.6320E-5$	$1.9665E-6$	0.2	$5.4306E-4$	$5.2908E-4$
0.02	$8.1213E-5$	$1.2495E-5$	0.3	$4.6960E-4$	$6.7104E-4$
0.03	$5.7655E-5$	$3.8937E-5$	0.4	$9.3627E-4$	$9.2246E-4$
0.04	$6.8161E-6$	$7.3656E-5$	0.5	$1.2605E-3$	$1.0909E-3$
0.05	$6.1827E-5$	$1.1358E-4$	0.6	$1.0655E-3$	$1.0554E-3$
0.06	$1.4037E-4$	$1.5621E-4$	0.7	$8.3439E-4$	$9.3402E-4$
0.07	$2.2235E-4$	$1.9957E-4$	0.8	$8.0252E-4$	$7.9872E-4$
0.08	$3.0263E-4$	$2.4210E-4$	0.9	$5.2229E-4$	$5.0084E-4$
0.09	$3.7730E-4$	$3.8267E-4$	1	$2.0448E-5$	$2.0530E-5$

Table 2.14: Computed errors using multiquadric RBF method for Example (2.10), with $\epsilon = 1$ (obtained using LOOCV strategy) and $\epsilon = 1.5$, for $N = 7$.

x_i	$\epsilon = 1.8$	$\epsilon = 2.5$	x_i	$\epsilon = 1.8$	$\epsilon = 2.5$
0	$5.8759E-7$	$5.7587E-5$	0.1	$1.7988E-4$	$1.9724E-4$
0.01	$3.7887E-5$	$8.8951E-4$	0.2	$3.8252E-4$	$4.5391E-4$
0.02	$5.1358E-5$	$7.7452E-4$	0.3	$9.9853E-4$	$7.0206E-4$
0.03	$4.8974E-4$	$7.1598E-3$	0.4	$4.1255E-3$	$4.8743E-3$
0.04	$6.5661E-3$	$7.9297E-3$	0.5	$3.1656E-3$	$7.1402E-4$
0.05	$9.6718E-4$	$9.5684E-4$	0.6	$5.2898E-3$	$2.0592E-3$
0.06	$8.7342E-4$	$5.8972E-4$	0.7	$7.9801E-3$	$5.0089E-3$
0.07	$7.1287E-4$	$3.9465E-4$	0.8	$4.2698E-3$	$3.5351E-3$
0.08	$2.5603E-4$	$2.1283E-4$	0.9	$8.9689E-3$	$6.9894E-3$
0.09	$6.5601E-5$	$9.9872E-5$	1	$6.1470E-4$	$7.9758E-4$

Table 2.15: Computed errors using multiquadric RBF method for Example (2.11), with $\epsilon = 1.8$ (obtained using LOOCV strategy) and $\epsilon = 2.5$, for $N = 7$.

Remark 2.4. From the numerical tests that we were able to establish, we remark that:

- Multiquadric RBFs method provide good accuracy for solving linear and nonlinear Fredholm integral equations.
- LOOCV strategy is a good strategy for choosing optimal shape parameter ϵ.

2.2.3 CSRBFs and composite RBFs method for solving integral equations

Global interpolation methods based on radial basis functions have been found to be efficient for surface fitting of scattered data sampled at n-dimensional scattered nodes. However, such interpolation leads to the solution of ill-conditioned system of equations [?, 53]. A lot of approaches are used to deal with the ill-conditioned problem in the global RBF [?, 30]. We propose a composite radial basis function using the gaussian and generalized multiquadric radial basis function which significantly improves the condition of the system matrix avoiding the above mentioned ill-conditionals in RBF interpolation schemes. Also in fact that generalized multiquadric RBF is a good choice for improving the accuracy of this approach because it depend on an exponent β which has some optimal strategies. We give a numerical comparaison between Wendland's compactly supported radial basis functions, our composite approach is necessary and important, in fact that the Wendland's CSRBFs have less accuracy compared with global RBFs.

The composite approach is given to approximate solutions of Volterra-Fredholm integral equations defined as

$$U(t) + \int_0^1 W_1(s,t,U(s))ds + \int_0^t W_2(s,t,U(s))ds = f(t), \tag{2.18}$$

$0 \leq t \leq 1$, where $f(t)$ is a known function, and

$$W_1(s,t,U(s)) = k_1(s,t)\,[U(s)]^{r_1}\,,$$

$$W_2(s,t,U(s)) = k_2(s,t)\,[U(s)]^{r_2}\,,$$

such that r_1, r_2 are positive integers.

The geometric representation of gaussian RBF and generalized RBF with some optimal values of the exponent β is given in figure **??**. Also the visualization of composite methods included gaussian and generalized multiquadric with the optimal values of the exponents β and with choosing a center $t_j = 0.5$ is given in figure 2.5.

Let consider the nonlinear Volterra-Fredholm integral equations written as equation (2.18), to approximate the function $U(t)$, we apply *RBF* interpolation in distinct grids from a definite domain. To this purpose, a linear convex combination of functions Φ_1, and Φ_2 is replaced in $U(t)$ as follows

$$U(t) \approx U_N = \sum_{j=1}^{N} c_j\left(\alpha_1 \Phi_{j,1}(t) + \alpha_2 \Phi_{j,2}(t)\right) = \sum_{j=1}^{N} c_j\left(\alpha_1 \Phi_1(\| t - t_j \|) + \alpha_2 \Phi_2(\| t - t_j \|)\right), \tag{2.19}$$

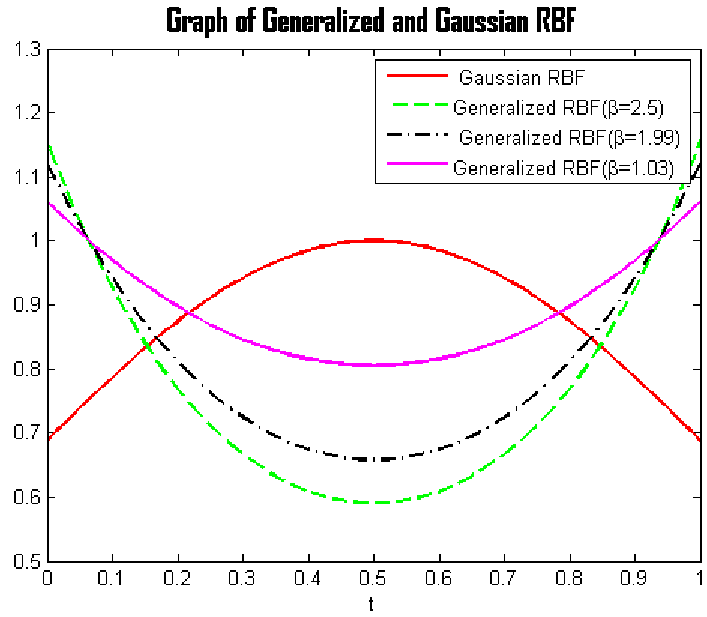

Figure 2.4: Representation of generalized multiquadric ($\phi(r) = (r^2 + \epsilon^2)^\beta$) and gaussian radial basis functions ($\phi(r) = e^{-\epsilon r^2}$), for a center $t_j = 0.5$, with $\epsilon_1 = 1.5$, $\epsilon_2 = 0.9$.

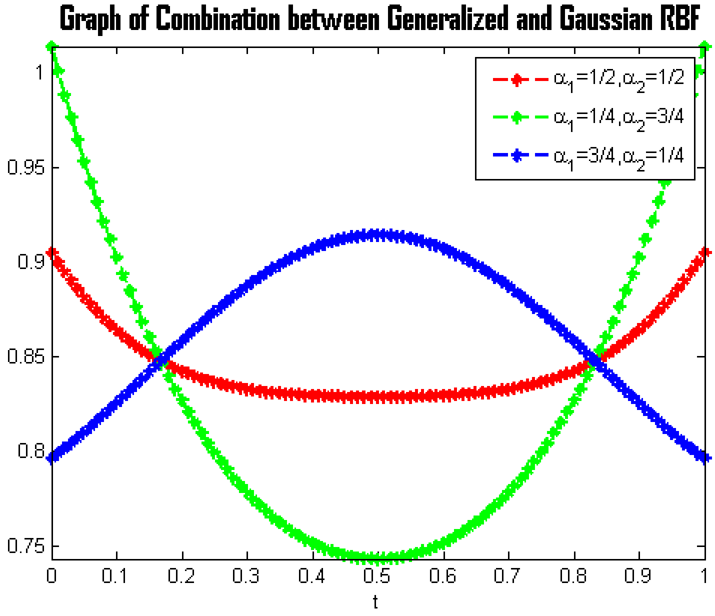

Figure 2.5: Representation of combination between generalized multiquadric and gaussian radial basis functions, for $\beta = 1.99$, $t_j = 0.5$, with $\epsilon_1 = 1.5$, $\epsilon_2 = 0.9$.

where $\Phi_{j,1}(t) = exp(-\epsilon_1(t-t_j)^2)$ is gaussian radial basis functions , $\Phi_{j,2}(t) = (\epsilon_2^2 + (t-t_j)^2)^\beta$ is generalized multiquadric radial basis functions, such that

$$\alpha_1 + \alpha_2 = 1.$$

The first integral in equation (2.18) can be approximated using the following integration formula

$$\int_0^1 W_1(s,t,U(s))ds = \sum_{k=1}^{N} w_k W_1(s_k, t, \sum_{j=1}^{N} c_j(\alpha_1\Phi_{j,1}(s_k) + \alpha_2\Phi_{j,2}(s_k))), \tag{2.20}$$

where, s_k, w_k are shifted Legendre-Gauss-Lobatto nodes and weights [38]. By replacing relation (2.20) in equation (2.18), and by collocating at points $t = t_i$, $i = 1 \dots N$ we obtain

$$\sum_{j=1}^{N} c_j(\alpha_1\Phi_1(\| t_i - t_j \|) + \alpha_2\Phi_2(\| t_i - t_j \|)) + \sum_{k=1}^{N} w_k W_1(s_k, t_i, \sum_{j=1}^{N} c_j(\alpha_1\Phi_1(\| s_k - s_j \|) \tag{2.21}$$

$$+\alpha_2\Phi_2(\| s_k - s_j \|))) + \int_0^{t_i} W_2(s, t_i, \sum_{j=1}^{N} c_j(\alpha_1\Phi_1(\| s - s_j \|) + \alpha_2\Phi_2(\| s - s_j \|)))ds = f(t_i),$$

the interval $[0, t_i]$ has been transformed to $[0, 1]$ by taking the changement $\mu = \frac{1}{t_i}s$, then equation (2.21) becomes

$$\sum_{j=1}^{N} c_j(\alpha_1\Phi_1(\| t_i - t_j \|) + \alpha_2\Phi_2(\| t_i - t_j \|)) + \sum_{k=1}^{N} w_k W_1(s_k, t_i, \sum_{j=1}^{N} c_j(\alpha_1\Phi_1(\| s_k - s_j \|)$$

$$+ \alpha_2\Phi_2(\| s_k - s_j \|))) + t_i \sum_{p=1}^{N} w_p W_2(t_i.\mu_p, t_i, \sum_{j=1}^{N} c_j(\alpha_1\Phi_1(\| (t_i.\mu_p) - s_j \|) + \alpha_2\Phi_2(\| sp - s_j \|)))$$

$$= f(t_i).$$

This is a nonlinear system of equations, that can be solved via Newton's iteration method to obtain the unknwon vector C, such that $C = (c_1, c_2, c_3, ..., c_N)^T$.

For our numerical experiments we are going to use Wendland's compactly supported radial basis function $\phi_{4,2}(r) = (1-r)_+^6(3 + 18r + 35r^2)$.

Some linear and nonlinear test equations are illustrated to show the efficiency and the applicability of the proposed approach, using different values of the shape parameter ϵ, and the parameter σ, and also in order to compare the condition number of the resulting interpolation matrix of exact solutions using both composite globaly RBFs and CSRBFs. The condition numbers have been calculated for different values of N. Tables (2.16)-(2.18) give a comparaison between the condition number of compactly supported RBF and composite RBF methods.

Example 2.12. Let given the following linear Volterra integral equation,

$$U(t) = cos(t) - sin(t)e^t + \int_0^t e^t U(s)ds,$$

such that $0 \leq t \leq 1$, where $U(t) = cos(t)$ is the exact solution. The numerical results using composite RBFs method are prestented in tables 2.20 − 2.22. The results for CSRBFs are sitting in table 2.19.

Example 2.13. Let given the following linear Volterra integral equation,

$$U(t) = \frac{1}{1+t^2} - \int_0^t \frac{s}{1+t^2} U(s)ds,$$

Table 2.16: The condition number, for $\alpha_1 = \frac{1}{2}, \alpha_2 = \frac{1}{2}$, with $\epsilon_1 = 2$, $\epsilon_2 = 1$, for CSRBF with $\sigma = 2$.

Points	$N = 8$	$N = 16$	$N = 26$
$CSRBF$	2.2449e+004	2.7446e+007	5.5641e+009
$\beta = 2.50$	5.8355e+005	1.4286e+017	2.9208e+018
$\beta = 1.99$	5.8282e+005	1.8006e+016	1.3761e+018
$\beta = 1.03$	4.6863e+005	2.2395e+016	2.3590e+018
$\beta = 1/2$	1.5176e+006	6.8947e+014	3.9602e+018
$\beta = -1/2$	6.0870e+005	3.3954e+014	6.6161e+018

Table 2.17: The condition number, for $\alpha_1 = \frac{1}{4}, \alpha_2 = \frac{3}{4}$, with $\epsilon_1 = 2$, $\epsilon_2 = 1$, for CSRBF with $\sigma = 4$.

Points	$N = 8$	$N = 16$	$N = 26$
$CSRBF$	8.0284e+005	1.0833e+009	2.2719e+011
$\beta = 2.50$	1.3254e+006	1.5005e+016	1.8195e+018
$\beta = 1.99$	1.3755e+006	1.7695e+016	1.1106e+018
$\beta = 1.03$	1.4163e+006	1.3366e+016	8.5205e+017
$\beta = 1/2$	1.1906e+006	4.3130+016	1.1758e+018
$\beta = -1/2$	1.1754e+006	3.2649e+014	7.9291e+018

Table 2.18: The condition number, for $\alpha_1 = \frac{3}{4}, \alpha_2 = \frac{1}{4}$, with $\epsilon_1 = 2$, $\epsilon_2 = 1$, for CSRBF with $\sigma = 6$.

Points	$N = 8$	$N = 16$	$N = 26$
$CSRBF$	6.1998e+006	8.6434e+009	1.8327e+012
$\beta = 2.50$	42.7593e+005	2.6945e+016	2.2988e+018
$\beta = 1.99$	2.4821e+005	1.1023e+016	2.4418e+018
$\beta = 1.03$	29.6331e+005	1.8556e+016	2.6759e+018
$\beta = 1/2$	4.6513e+005	3.4425e+015	2.5299 e+019
$\beta = -1/2$	3.8105e+005	4.4327e+014	1.2450e+020

Table 2.19: Computed errors using CSRBFs method for example (2.12), for $N = 8$.

x_i	$\sigma = 1.5$	$\sigma = 5$	$\sigma = 10$	$\sigma = 17$
0	0.0000	0.0000	0.0000	0.0000
0.1	8.8477×10^{-4}	9.7564×10^{-6}	1.7964×10^{-5}	1.2570×10^{-5}
0.2	1.7991×10^{-4}	2.2993×10^{-6}	3.7244×10^{-6}	2.5397×10^{-6}
0.3	9.3080×10^{-4}	1.9144×10^{-5}	2.1101×10^{-5}	1.2945×10^{-5}
0.4	3.8919×10^{-6}	4.9272×10^{-7}	2.7247×10^{-7}	1.1019×10^{-7}
0.5	5.4415×10^{-4}	3.2812×10^{-5}	1.5171×10^{-5}	4.7477×10^{-6}
0.6	8.6641×10^{-5}	8.5426×10^{-6}	2.8192×10^{-6}	2.9720×10^{-7}
0.7	1.3910×10^{-4}	5.0884×10^{-5}	1.2654×10^{-5}	1.6142×10^{-6}
0.8	6.3028×10^{-5}	2.6910×10^{-6}	7.3324×10^{-8}	5.6877×10^{-7}
0.9	1.5270×10^{-4}	4.3025×10^{-5}	9.8141×10^{-6}	2.1247×10^{-6}
1	9.2935×10^{-5}	2.7863×10^{-8}	8.0116×10^{-7}	6.5655×10^{-7}

Table 2.20: Computed errors for different values of β for example (2.12), for $N = 8$, with $\epsilon_1 = 2$, $\epsilon_2 = 1.4$.

Points	**Case 1:** $\alpha_1 = \frac{1}{2}$, $\alpha_2 = \frac{1}{2}$				
x_i	$\beta = 2.5$	$\beta = 1.99$	$\beta = 1.03$	$\beta = 1/2$	$\beta = -1/2$
0	6.1448×10^{-11}	1.1472×10^{-10}	1.1417×10^{-11}	1.0468×10^{-10}	2.5198×10^{-11}
0.1	1.4981×10^{-4}	1.5584×10^{-4}	1.1684×10^{-4}	2.4368×10^{-5}	1.7139×10^{-5}
0.2	3.5608×10^{-5}	3.7265×10^{-5}	2.7240×10^{-5}	1.8489×10^{-6}	3.1560×10^{-6}
0.3	2.8942×10^{-4}	3.0432×10^{-4}	2.1267×10^{-4}	3.5923×10^{-5}	1.2952×10^{-5}
0.4	1.0743×10^{-5}	1.1367×10^{-5}	7.7328×10^{-6}	2.5459×10^{-6}	2.0201×10^{-7}
0.5	3.0574×10^{-4}	3.2197×10^{-4}	2.0352×10^{-4}	1.4291×10^{-4}	1.4031×10^{-5}
0.6	5.4675×10^{-5}	5.7432×10^{-5}	3.3664×10^{-5}	3.7949×10^{-5}	5.9082×10^{-6}
0.7	2.4132×10^{-4}	2.5240×10^{-4}	1.4156×10^{-4}	1.9437×10^{-4}	3.3943×10^{-5}
0.8	6.8421×10^{-6}	7.0386×10^{-6}	3.7046×10^{-6}	6.2990×10^{-6}	1.2474×10^{-6}
0.9	1.1984×10^{-4}	1.2361×10^{-4}	5.9127×10^{-5}	1.3789×10^{-4}	2.9144×10^{-5}
1	3.2934×10^{-7}	3.6626×10^{-7}	9.7643×10^{-7}	3.3585×10^{-6}	9.3363×10^{-7}

Table 2.21: Computed errors for different values of β for example (2.12), for $N = 8$, with $\epsilon_1 = 2$, $\epsilon_2 = 1.4$.

Points	**Case 2:** $\alpha_1 = \frac{1}{4}$, $\alpha_2 = \frac{3}{4}$				
x_i	$\beta = 2.5$	$\beta = 1.99$	$\beta = 1.03$	$\beta = 1/2$	$\beta = -1/2$
0	2.4914×10^{-10}	4.1172×10^{-10}	3.1159×10^{-10}	1.8658×10^{-10}	3.8014×10^{-12}
0.1	1.6148×10^{-4}	2.0227×10^{-4}	1.5929×10^{-4}	1.1235×10^{-4}	7.6660×10^{-6}
0.2	3.8338×10^{-5}	4.8465×10^{-5}	3.8480×10^{-5}	2.5974×10^{-5}	2.7232×10^{-6}
0.3	3.1254×10^{-4}	3.9818×10^{-4}	3.1761×10^{-4}	1.9947×10^{-4}	3.3700×10^{-5}
0.4	1.1584×10^{-5}	1.4897×10^{-5}	1.1983×10^{-5}	7.1808×10^{-6}	1.5267×10^{-6}
0.5	3.3478×10^{-4}	4.2832×10^{-4}	3.4028×10^{-4}	1.8309×10^{-4}	5.9382×10^{-5}
0.6	6.0643×10^{-5}	7.7422×10^{-5}	6.0993×10^{-5}	2.9194×10^{-5}	1.3426×10^{-5}
0.7	2.7048×10^{-4}	3.4341×10^{-4}	2.6803×10^{-4}	1.2018×10^{-4}	6.5394×10^{-5}
0.8	7.8807×10^{-6}	9.7807×10^{-6}	7.4088×10^{-6}	3.0516×10^{-6}	2.0411×10^{-6}
0.9	1.3914×10^{-4}	1.7301×10^{-4}	1.3056×10^{-4}	4.5823×10^{-5}	4.1905×10^{-5}
1	1.9824×10^{-7}	2.4764×10^{-7}	3.0559×10^{-7}	1.1943×10^{-6}	7.2315×10^{-7}

Table 2.22: Computed errors for different values of β for example (2.12), for $N = 8$, with $\epsilon_1 = 2$, $\epsilon_2 = 1.4$.

Points	**Case 3:** $\alpha_1 = \frac{3}{4}$, $\alpha_2 = \frac{1}{4}$				
x_i	$\beta = 2.5$	$\beta = 1.99$	$\beta = 1.03$	$\beta = 1/2$	$\beta = -1/2$
0	1.5930×10^{-11}	5.7316×10^{-11}	3.6928×10^{-10}	2.4075×10^{-11}	1.0325×10^{-11}
0.1	1.4680×10^{-4}	1.3448×10^{-4}	1.0830×10^{-4}	1.2591×10^{-5}	4.6829×10^{-7}
0.2	3.4794×10^{-5}	3.1741×10^{-5}	1.8034×10^{-5}	1.4292×10^{-6}	1.0380×10^{-5}
0.3	2.8080×10^{-4}	2.5381×10^{-4}	4.6260×10^{-5}	9.2679×10^{-6}	7.2673×10^{-5}
0.4	1.0396×10^{-5}	9.3558×10^{-6}	6.4747×10^{-7}	8.2161×10^{-7}	2.4733×10^{-6}
0.5	2.9114×10^{-4}	2.5719×10^{-4}	1.5933×10^{-4}	5.4093×10^{-5}	4.9071×10^{-5}
0.6	5.1294×10^{-5}	4.4507×10^{-5}	5.3213×10^{-5}	1.5009×10^{-5}	5.2126×10^{-6}
0.7	2.2410×10^{-4}	1.9231×10^{-4}	2.8937×10^{-4}	7.8222×10^{-5}	1.4191×10^{-5}
0.8	6.2136×10^{-7}	5.2292×10^{-6}	9.9079×10^{-6}	2.6138×10^{-6}	2.5990×10^{-8}
0.9	1.0748×10^{-4}	8.8730×10^{-5}	2.2789×10^{-4}	5.7418×10^{-5}	6.9541×10^{-6}
1	4.9694×10^{-7}	6.5803×10^{-7}	6.7732×10^{-6}	1.4415×10^{-6}	1.0370×10^{-6}

Table 2.23: Computed errors using CSRBFs method for example (2.13), for $N = 8$.

x_i	$\sigma = 2$	$\sigma = 7$	$\sigma = 12$	$\sigma = 20$
0	0.0000	0.0000	0.0000	0.0000
0.1	1.7060×10^{-4}	1.0122×10^{-4}	7.4862×10^{-5}	5.5344×10^{-5}
0.2	1.5147×10^{-5}	8.7529×10^{-6}	6.4506×10^{-6}	4.7501×10^{-6}
0.3	1.9390×10^{-4}	1.1035×10^{-4}	8.1489×10^{-5}	6.0304×10^{-5}
0.4	1.4912×10^{-5}	8.4504×10^{-6}	6.3623×10^{-6}	4.8505×10^{-6}
0.5	1.1898×10^{-4}	6.5150×10^{-5}	5.1434×10^{-5}	4.1946×10^{-5}
0.6	4.0736×10^{-6}	1.5724×10^{-6}	1.2627×10^{-6}	1.0763×10^{-6}
0.7	7.6549×10^{-5}	1.8076×10^{-5}	1.6456×10^{-5}	1.6534×10^{-5}
0.8	7.3822×10^{-6}	1.0342×10^{-7}	3.1817×10^{-7}	6.4414×10^{-7}
0.9	4.8168×10^{-5}	2.8062×10^{-6}	2.6136×10^{-7}	2.8698×10^{-6}
1	5.3201×10^{-8}	2.0763×10^{-8}	2.9469×10^{-8}	2.8494×10^{-8}

Table 2.24: Computed errors for different values of β for example (2.13), for $N = 8$, with $\epsilon_1 = 1.2$, $\epsilon_2 = 1.8$.

Points	**Case 1:** $\alpha_1 = \frac{1}{2}$, $\alpha_2 = \frac{1}{2}$				
x_i	$\beta = 2.5$	$\beta = 1.99$	$\beta = 1.03$	$\beta = 1/2$	$\beta = -1/2$
0	2.7321×10^{-9}	2.6885×10^{-8}	8.5788×10^{-10}	1.2254×10^{-8}	4.2775×10^{-10}
0.1	3.3737×10^{-5}	1.1025×10^{-5}	2.8653×10^{-6}	6.2702×10^{-6}	1.0061×10^{-6}
0.2	3.4501×10^{-6}	1.0186×10^{-6}	2.01135×10^{-7}	4.0897×10^{-7}	1.7555×10^{-7}
0.3	5.7603×10^{-5}	1.5627×10^{-5}	2.1928×10^{-6}	3.6338×10^{-6}	3.8144×10^{-6}
0.4	5.8170×10^{-6}	4.4627×10^{-6}	1.2278×10^{-7}	7.1455×10^{-8}	4.5584×10^{-7}
0.5	6.4828×10^{-5}	1.5128×10^{-5}	3.4877×10^{-7}	2.8697×10^{-6}	5.7207×10^{-6}
0.6	2.6069×10^{-6}	5.8220×10^{-7}	9.9686×10^{-9}	2.4320×10^{-7}	2.3924×10^{-7}
0.7	6.1824×10^{-5}	1.3269×10^{-5}	7.2680×10^{-7}	8.2425×10^{-5}	5.8774×10^{-6}
0.8	5.8559×10^{-6}	1.2356×10^{-6}	9.1993×10^{-8}	1.0174×10^{-6}	5.4877×10^{-7}
0.9	3.7614×10^{-5}	7.9349×10^{-6}	6.2590×10^{-7}	7.7193×10^{-6}	3.8340×10^{-6}
1	9.1338×10^{-8}	9.9908×10^{-9}	1.1817×10^{-9}	7.7016×10^{-9}	1.2181×10^{-10}

Table 2.25: Computed errors for different values of β for example (2.13), for $N = 8$, with $\epsilon_1 = 1.2$, $\epsilon_2 = 1.8$.

Points	**Case 2:** $\alpha_1 = \frac{1}{4}$, $\alpha_2 = \frac{3}{4}$				
x_i	$\beta = 2.5$	$\beta = 1.99$	$\beta = 1.03$	$\beta = 1/2$	$\beta = -1/2$
0	6.3737×10^{-9}	3.0382×10^{-8}	1.5538×10^{-7}	2.1907×10^{-9}	2.5828×10^{-9}
0.1	1.2796×10^{-6}	8.2210×10^{-6}	7.9484×10^{-5}	2.3492×10^{-6}	6.8278×10^{-7}
0.2	2.0098×10^{-6}	9.0816×10^{-7}	8.1290×10^{-6}	1.4831×10^{-7}	1.4331×10^{-7}
0.3	4.2115×10^{-6}	1.6000×10^{-5}	1.3541×10^{-4}	1.3008×10^{-6}	3.2868×10^{-6}
0.4	4.9357×10^{-7}	1.6849×10^{-6}	1.3629×10^{-5}	3.1870×10^{-8}	4.0351×10^{-7}
0.5	6.1112×10^{-6}	1.9453×10^{-5}	1.5104×10^{-4}	6.7969×10^{-7}	5.1492×10^{-6}
0.6	2.5339×10^{-7}	7.9609×10^{-7}	6.0305×10^{-6}	5.2104×10^{-8}	2.1678×10^{-7}
0.7	6.1805×10^{-6}	1.9163×10^{-5}	1.4212×10^{-4}	1.7402×10^{-6}	5.3555×10^{-6}
0.8	5.7251×10^{-7}	1.8215×10^{-6}	1.3339×10^{-5}	1.8977×10^{-7}	5.0083×10^{-7}
0.9	3.4973×10^{-6}	1.1656×10^{-6}	8.4759×10^{-5}	1.2633×10^{-6}	3.0878×10^{-6}
1	8.9967×10^{-8}	1.4092×10^{-7}	1.5912×10^{-7}	6.5870×10^{-10}	1.8122×10^{-10}

Table 2.26: Computed errors for different values of β for example (2.13), for $N = 8$, with $\epsilon_1 = 1.2$, $\epsilon_2 = 1.8$.

Points	**Case 3:** $\alpha_1 = \frac{3}{4}$, $\alpha_2 = \frac{1}{4}$				
x_i	$\beta = 2.5$	$\beta = 1.99$	$\beta = 1.03$	$\beta = 1/2$	$\beta = -1/2$
0	7.9699×10^{-9}	1.3492×10^{-9}	7.2788×10^{-11}	1.7852×10^{-9}	1.8707×10^{-10}
0.1	4.5650×10^{-6}	3.5568×10^{-6}	1.7885×10^{-6}	5.6234×10^{-7}	1.4231×10^{-6}
0.2	3.6850×10^{-7}	2.6927×10^{-7}	7.9677×10^{-8}	1.2280×10^{-7}	2.1754×10^{-7}
0.3	4.9181×10^{-6}	3.2988×10^{-6}	8.8163×10^{-10}	2.8323×10^{-6}	4.5048×10^{-6}
0.4	3.9391×10^{-7}	2.3264×10^{-7}	1.1443×10^{-7}	3.4699×10^{-7}	5.2448×10^{-7}
0.5	3.3589×10^{-6}	1.5656×10^{-6}	2.5055×10^{-6}	4.3804×10^{-6}	6.4704×10^{-6}
0.6	1.1279×10^{-7}	3.9494×10^{-8}	1.3306×10^{-7}	1.8040×10^{-7}	2.6873×10^{-7}
0.7	2.2172×10^{-6}	4.5623×10^{-7}	3.8033×10^{-6}	4.3878×10^{-6}	6.5631×10^{-6}
0.8	1.9483×10^{-7}	2.2792×10^{-8}	4.0027×10^{-7}	3.9761×10^{-7}	6.1202×10^{-7}
0.9	1.2811×10^{-6}	1.3382×10^{-7}	2.6935×10^{-6}	2.3598×10^{-6}	3.7781×10^{-6}
1	7.3671×10^{-9}	2.0734×10^{-9}	7.2099×10^{-10}	2.4654×10^{-9}	2.5999×10^{-10}

such that $0 \leq t \leq 1$, where $U(t) = (1 + t^2)^{\frac{-3}{2}}$ is the exact solution. The numerical results using composite RBFs are prestented in tables 2.24 – 2.26. The results for CSRBFs are sitting in table 2.23.

Analysis of experimental results

From the test examples, we remark that, for a small number of mesh points ($N = 8$), the presented methods provide remarkable accurate solutions. The use of optimal values of exponent β for generalized multiquadric radial basis functions give a good accuracy compared with the results obtained by the compactly supported radial basis functions (CSRBFs). For fixed values of the shape parameter ϵ and the parameter σ, the condition number also grows with N increasing. The condition number when using CSRBFs is much smaller compared to the use of composite RBFs, but it is approximately close for small N. Generally for a fixed number of collocation points N, smaller values of ϵ produce better approximations, but the matrix A will be more ill-conditioned . When the number of mesh points increases, the accuracy of the solution can be increased but the condition number of the matrix becomes very large and the matrix tends to be ill-conditioned. Compactly supported RBF method gives a balance between good accuracy and small conditioning number, the accuracy of CSRBFs method can be improved by using strategies to find the optimal parameter σ.

2.2.4 CSRBFs and FDM for solving PDEs.

The classical methods like Finite elements method (FEM), Finite difference method (FDM) and Finite Volume method (FVM) are still the most used methods for solving systems of partial differential equations in physical modeling problems [47, 56, 65].
These methods benefit from a very solid theoritical foundation and many techniques have come to improve them over the years. However, their implementation remains difficulties and costly in certain cases, notably in the field of modeling large deformations in lagrangian formulation. However in the recent years considerable attention was paid to the so called meshess methods which operate with nodes rather than meshes. The motivation came mainly from the following considerations

- RBF methods do not require grid generation which can be not an easy task in the three dimensional cases.

- RBF methods may be successfully applied to achieve exponential accuracy where traditional methods either have difficulties or fail.

- RBF methods are more appropriate than FEM or FDM methods in the case of very large mesh deformation and moving discontinuities.

3- RBF methods are easy to apply for the approximation of multivariate scattered data, and easy to improve the numerical accuracy by adding more points around large gradient regions.

The use of RBF for PDEs discretizations offers some nice possibilities. First, some RBF-based discretizations have potential for providing convergence rates dependent on exact solutions smoothness only rather than on degrees of underlying polynomial approximations. In certain cases, they can be exponential. Second, good RBF performance in three-dimensional cases is theoretically expected. At present, the most popular lines of attack when constructing RBF-PDE solvers seem to be collocation and boundary elements approaches [35, 54]. In [90, 91] convergence proofs and error estimates of the collocation procedure are presented. A profound impact on the RBF-collocation technique applications is due to papers [54, 87].

FDM for solving Burger's equation

Burger's equation, which takes its name from the German physicist Johannes Martinus Burgers(1895- 1981) is a partial differential equation fundamental to fluid mechanics. it is used in various areas of applied mathematics, such as the study of fluid mechanics, the modeling of gas dynamics, traffic flow. This equation is used to study the evolution of the speed of fluids in a given environment.

Given a speed u and a viscosity coefficient v, the general form of Burger's equation is

$$u_t + uu_x = vu_{xx},\ v = \frac{1}{R}, \tag{2.22}$$

with initial and boundary conditions

$$\begin{cases} u(0,t) = u(1,t) = 0, & t > 0 \\ u(x,0) = f(x), & 0 \leq x \leq 1, \end{cases}$$

where $f(x)$ is a given function and R is the number of Reynolds. The resolution of this equation presents many difficulties like the problems of stability and entropy.

For study in good conditions, we have to take into account several conditions like the condition of stability and Rankine - Hugoniot conditions.

Using the shemes that respect these conditions, we will note here only those which are in conservative form for this equation as follow

$$u_i^{j+1} = u_i^j - \frac{\Delta t}{\Delta x}\left(g(u_i^j, u_i^{j+1}) - g(u_{i-1}^j, u_j, j)\right) + \frac{1}{R}u_{xx}, \tag{2.23}$$

where $g(u,v)$ represents the digital flow of the equation, those shemes respect the following C.F.L condition given by

$$\beta^j = sup|u_i^j|\frac{\Delta t^n}{\Delta x} \leqslant 1.$$

We use Lax-Friedrichs sheme [92]

$$g(u,v) = \frac{1}{2}\left(\frac{1}{2}(u^2+v^2) - \frac{\Delta x}{\Delta t}(v-u)\right),$$

and Lax-Wendroff sheme [92]

$$g(u,v) = \frac{1}{2}\left(\frac{1}{2}(u^2+v^2) - \frac{\Delta x}{\Delta t}\left(\frac{u+v}{2}\right)\left(\frac{1}{2}(v^2-u^2)\right)\right).$$

By replacing the second partial derivative by

$$u^{''}(x) = \frac{u(x+\Delta x) - 2u(x) + u(x-\Delta x)}{(\Delta x)^2},$$

the descritization of Burger's equation becomes

$$u_i^{j+1} = \left(\frac{1}{2} + \frac{\Delta t}{R(\Delta x)^2}\right)(u_{i+1}^j + u_{i-1}^j) - \frac{2\Delta t}{R(\Delta x)^2} u_i^j + \frac{\Delta t}{4\Delta x}\left((u_{i-1}^j)^2 - (u_{i+1}^j)^2\right).$$

The initial condition f is defined as $f(x) = sin(\pi x)$ and $R = 10000$. For $T_{max} = 0.14$ and $\Delta t = 0.02$ and $\Delta x = 0.125$, the numerical results are represented in table 2.27. For different values of T_{max}, the curves of approximate solution are given in figure 2.6.

x/t	0	0.02	0.04	0.06	0.08	0.1	0.12	0.14
0	0	0	0	0	0	0	0	0
0.125	0.383	0.336	0.297	0.267	0.243	0.226	0.205	0.189
0.25	0.707	0.625	0.560	0.507	0.462	0.426	0.391	0.359
0.375	0.924	0.834	0.756	0.689	0.631	0.579	0.534	0.492
0.5	1.000	0.924	0.851	0.783	0.721	0.665	0.613	0.565
0.625	0.924	0.874	0.818	0.762	0.706	0.653	0.603	0.556
0.75	0.707	0.682	0.649	0.610	0.570	0.529	0.489	0.451
0.875	0.383	0.374	0.359	0.341	0.320	0.298	0.276	0.254
1	0	0	0	0	0	0	0	0

Table 2.27: The numerical results using FDM for solving Burger's equation for $Tmax = 0.14$.

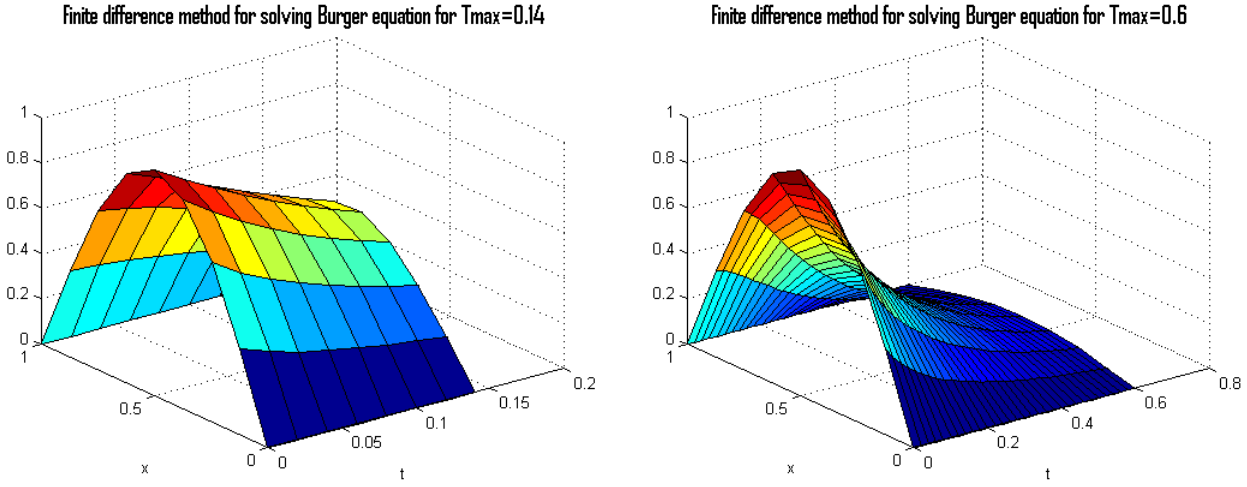

Figure 2.6: Burgers-FDM, with different values of T_{max}.

CSRBFs collocation method for solving Burger's equation

The compactly supported basis functions consist of a polynomial which are non-zero on $[0, 1)$ and vanish on $[1, \infty)$. This reduces the original resultant full matrix to a sparse matrix. The operation of the banded matrix system could reduce the ill-conditioning of the resultant coefficient matrix due to the use of the global radial basis functions.

The approximate solution using CSRBFs is given by

$$u_n(x_i, y_j) = \sum_{j=1}^{N} \alpha_j \phi_{l,k}\left(\frac{r_{i,j}}{\sigma_j}\right),$$

where σ is parameter, α_j are the coefficients to be determined using collocation points (x_i, y_j).

For our experiment, we use Wendland's compactly supported radial basis function $\phi^{\sigma}_{3,1}$, which is given by

$$\phi^{\sigma}_{3,1}(r) = \Big(1 - \frac{r}{\sigma}\Big)^4_+ \Big(1 + 4\frac{r}{\sigma}\Big)^2,$$

with $r = ||x - x_j||$, $\sigma > 0$, we obtain

$$\frac{\partial \widehat{u}}{\partial t} + \frac{\partial}{\partial x}\left(\frac{\widehat{u}^2}{2}\right) = \frac{1}{R}\frac{\partial^2 \widehat{u}}{\partial x^2},$$

because of stability's reasons and under C.F.L conditions, we keep the sheme of Lax-Friedrichs [92] for solving our equation, so the equation becomes

$$u_i^{n+1} = \frac{1}{2}(u^n_{i+1}) + \frac{\Delta t}{4\Delta x}\Big((u^n_{i-1})^2 - (u^n_{i+1})^2\Big) + \frac{\Delta t}{R}\sum_{j=1}^{n} \alpha_j(t_n)\frac{4}{\sigma^2}\Big(1 - \frac{r}{\sigma}\Big)^2_+ \Big(-7 + 12\frac{r}{\sigma}\Big).$$

The numerical results using $\Delta t = 0.02$,$\Delta x = 0.125$, $T_{max} = 0.14$ and $\sigma = 5$ are sitting in table 2.28, for different values of T_{max}, the curves of approximate solution are given in figure 2.7.

x/t	0	0.02	0.04	0.06	0.08	0.1	0.12	0.14
0	0	0	0	0	0	0	0	0
0.125	0.383	0.333	0.297	0.268	0.243	0.223	0.205	0.189
0.25	0.708	0.625	0.560	0.507	0.462	0.424	0.390	0.359
0.375	0.924	0.834	0.756	0.690	0.631	0.579	0.534	0.492
0.5	1.000	0.924	0.851	0.783	0.721	0.665	0.613	0.565
0.625	0.924	0.874	0.818	0.762	0.706	0.653	0.603	0.557
0.75	0.707	0.682	0.649	0.611	0.570	0.529	0.489	0.451
0.875	0.383	0.374	0.359	0.341	0.320	0.298	0.276	0.254
1	0	0	0	0	0	0	0	0

Table 2.28: The numerical results using CSRBF for solving Burger's equation for $Tmax = 0.14$.

FDM for solving Poisson's equation

The Poisson's equation is a second order PDE named after Simeon Denis Poisson, it is generalization of Laplace's equation, it arises, for instance, to describe the potential field caused by a given charge or mass density distribution, with the potential field known, one can then calculate gravitational or electric field.
Our aim now, is to solve Poisson's equation with homogeneous Dirichlet boundary conditions

$$\begin{cases} \Delta u = \dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} = f, & (x, y) \in]0, 1[\times]0, 1[\\ u(x, 1) = u(0, y) = u(1, y) = 0, & \\ u(x, 0) = sin(\pi x). & \end{cases}$$

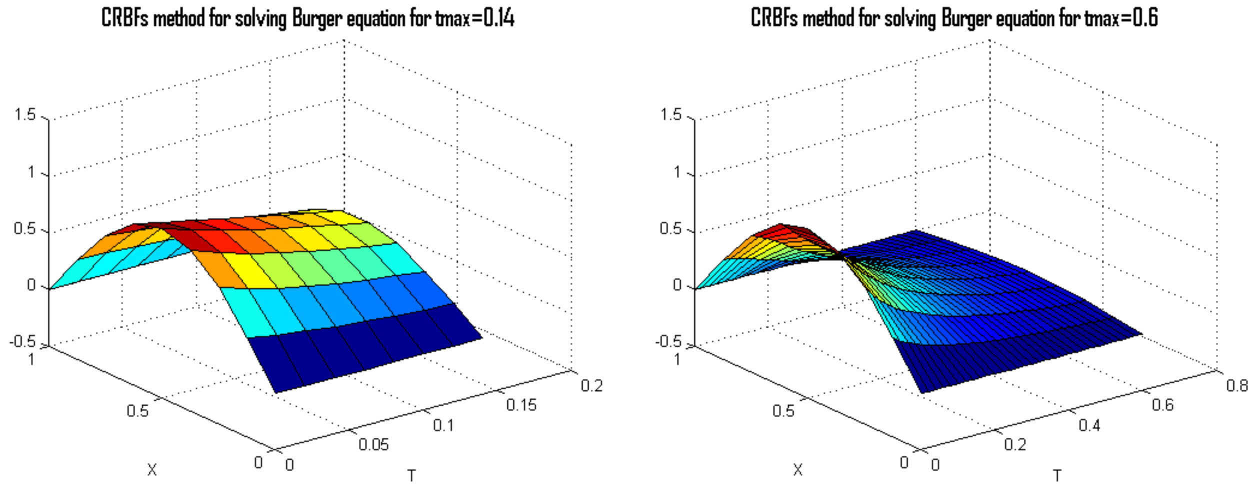

Figure 2.7: Burger-CSRBFs, with different values of T_{max}.

We see that if $f = 0$, we obtain Laplace's equation, where $\Delta x = 1/n$ and $\Delta y = 1/m$, with n, m are the numbers of subregions. As this equation is only made up of second derivatives, we will approximate them by using the Taylor's formula of order 2 which gives us the centred approximation of the second partial derivatives, given as follow

$$\frac{\partial^2 u}{\partial x^2}(x,y) = \frac{u(x+\Delta x, y) - 2u(x,y) + u(x-\Delta x, y)}{(\Delta x)^2},$$

and

$$\frac{\partial^2 u}{\partial y^2}(x,y) = \frac{u(x, y+\Delta y) - 2u(x,y) + u(x, y-\Delta y)}{(\Delta y)^2}.$$

Subtituting by point (x_i, y_j), we have

$$\frac{\partial^2 u}{\partial x^2}(x_i, y_j) = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2},$$

$$\frac{\partial^2 u}{\partial y^2}(x_i, y_j) = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta y)^2}.$$

Replacing these approximations into Poisson's equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f,$$

yields to

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta y)^2} = f_{i,j}.$$

If $\Delta x = \Delta y = h$, the Poissson equation can be written as

$$u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j} = h^2 f_{i,j}.$$

We notice that at each step, we need to know the points $u_{i-1,j}$, $u_{i,j-1}$, $u_{i+1,j}$ and $u_{i,j+1}$ to calculate the value of $u_{i,j}$ at the point (x_i, y_j).

The corresponding matrix system $Au = b$ is given by

$$k = (j-1) \times n + i, \ \forall i = 2 : n-1, \ \forall j = 2 : m-1$$

$$\begin{cases} A(k,k) = -4, & A(k,k-1) = 1, \quad A(k,k+1) = 1 \\ A(k,k+n) = 1, & A(k,k-n) = 1, \quad b(k,1) = (h)^2 f_{i,j} \end{cases}$$

$$\begin{cases} A(k,k) = 1, & i = 1, \forall j = 1 : m \\ b(k,1) = 0, \end{cases}$$

$$\begin{cases} A(k,k) = 1, & i = n, \forall j = 1 : m \\ b(k,1) = sin(\pi x), \end{cases}$$

$$\begin{cases} A(k,k) = 1, & j = 1, j = m, \forall i = 1 : n \\ b(k,1) = 0. \end{cases}$$

From the numerical results using Finite difference method, we obtain the approach values of u for $h = 0.1$ and $f = \frac{-5\pi^2}{4} sin(\pi x) cos(\frac{\pi y}{2})$, which are sitting in table 2.29. The curves of approximate solution are given in figure 2.8.

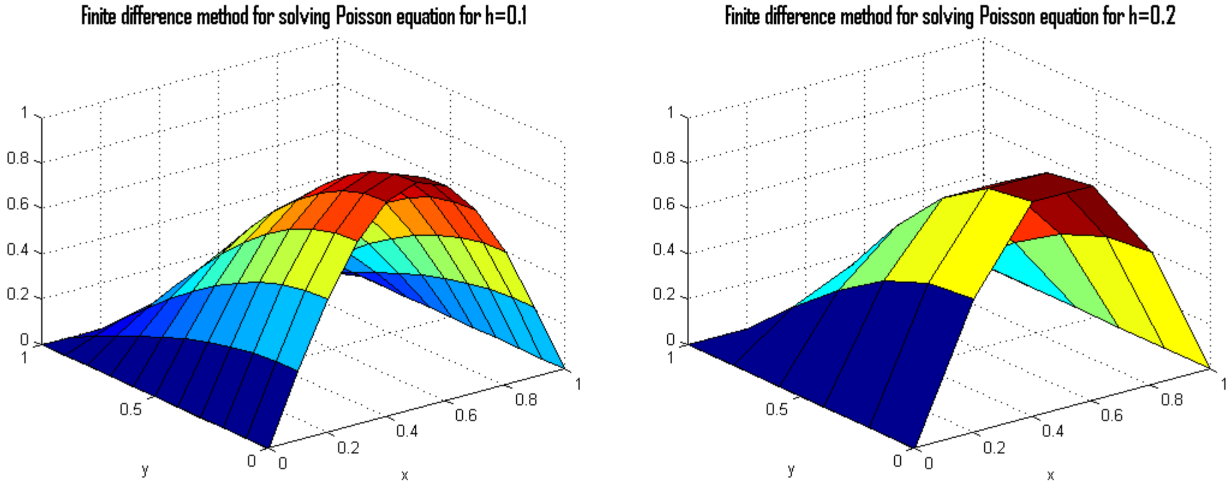

Figure 2.8: Poisson-FDM, with different values of T_{max}.

CSRBFs collocatin method for solving Poisson's equation

Now, we are going to solve Poisson's equation by using CSRBF, let given the following problem with boundary conditions:

$$\begin{cases} \Delta u = \dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} = f, & (x,y) \in]0,1[\times]0,1[, \\ u(x,1) = u(0,y) = u(1,y) = 0, \\ u(x,0) = sin(\pi x). \end{cases}$$

We suppose that u_n is the approximate solution of u written as

$$u_n = \sum_{j=1}^{N} \alpha_j \phi^\sigma,$$

x/y	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0	0	0	0	0	0	0	0	0	0	0
0.1	0.309	0.306	0.295	0.276	0.251	0.220	0.183	0.141	0.096	0.049
0.2	0.588	0.582	0.561	0.526	0.478	0.418	0.347	0.268	0.183	0.093
0.3	0.809	0.801	0.772	0.724	0.658	0.575	0.478	0.369	0.251	0.127
0.4	0.951	0.941	0.907	0.851	0.773	0.676	0.562	0.434	0.296	0.150
0.5	1.000	0.990	0.954	0.895	0.813	0.711	0.591	0.457	0.311	0.157
0.6	0.951	0.941	0.907	0.851	0.773	0.676	0.562	0.434	0.296	0.150
0.7	0.809	0.801	0.772	0.724	0.658	0.575	0.478	0.369	0.251	0.127
0.8	0.588	0.582	0.561	0.526	0.478	0.418	0.347	0.268	0.183	0.093
0.9	0.309	0.306	0.295	0.276	0.251	0.220	0.183	0.141	0.096	0.049
1	0	0	0	0	0	0	0	0	0	0

Table 2.29: The numerical results using FDM for solving Poisson's equation.

we have $\dfrac{\partial^2 u}{\partial x^2} = \displaystyle\sum_{j=1}^{N} \alpha_j \dfrac{\partial^2 \phi^\sigma}{\partial x^2}$, and $\dfrac{\partial^2 u}{\partial y^2} = \displaystyle\sum_{j=1}^{N} \alpha_j \dfrac{\partial^2 \phi^\sigma}{\partial y^2}$.

For this equation, we use the compactly supported radial basis function $\phi^\sigma_{4,2}$ of Wendland which is written in the form

$$\phi^\sigma_{4,2}(r) = \left(1 - \frac{r}{\sigma}\right)^6_+ \left(3 + 18\frac{r}{\sigma} + 35\left(\frac{r}{\sigma}\right)^2\right),$$

with r is the Euclidean norm.

Then, we obtain

$$\frac{\partial^2 u}{\partial x^2} = \frac{-56}{\sigma^2}(1 - \frac{r}{\sigma})^4_+ \left(1 + 4\frac{r}{\sigma} - 5(\frac{r}{\sigma})^2 - 30\big(\frac{x - x_j}{\sigma})^2\big)\right)$$

$$\frac{\partial^2 u}{\partial y^2} = \frac{-56}{\sigma^2}(1 - \frac{r}{\sigma})^4_+ \left(1 + 4\frac{r}{\sigma} - 5\Big(\frac{r}{\sigma}\Big)^2 - 30\Big(\frac{y - y_j}{\sigma}\Big)^2\right).$$

This yields to

$$\Delta u = \sum_{j=1}^{N} \alpha_j \frac{-56}{\sigma^2}(1 - \frac{r}{\sigma})^4_+ \left(2 + 8\frac{r}{\sigma} - 40\Big(\frac{r}{\sigma}\Big)^2\right) = f,$$

where α is the vector that should be determined.
The matrix system is given by $A\alpha = B$, where

$$A(i,j) = \begin{cases} \frac{-56}{\sigma^2}(1 - \frac{r_{i,j}}{\sigma_j})^4_+ \left(2 + 8\frac{r_{i,j}}{\sigma_j} - 40(\frac{r_{i,j}}{\sigma_j})^2\right) & if\ [x,y] \in \Omega \\ (1 - \frac{r_{i,j}}{\sigma_j})^6_+ \left(3 + 18\frac{r_{i,j}}{\sigma_j} + 35(\frac{r_{i,j}}{\sigma_j})^2\right) & if[x,y] \in \partial\Omega. \end{cases}$$

and B takes the boundary values.
The approximate values of u using CSRBFs method ,for $h = 0.1$ and $\sigma = 1.3$ and $f = \frac{-5\pi^2}{4} sin(\pi x) cos(\dfrac{\pi y}{2})$, are sitting in table 2.30, the curves of approximate solution are given in figure 2.9 .

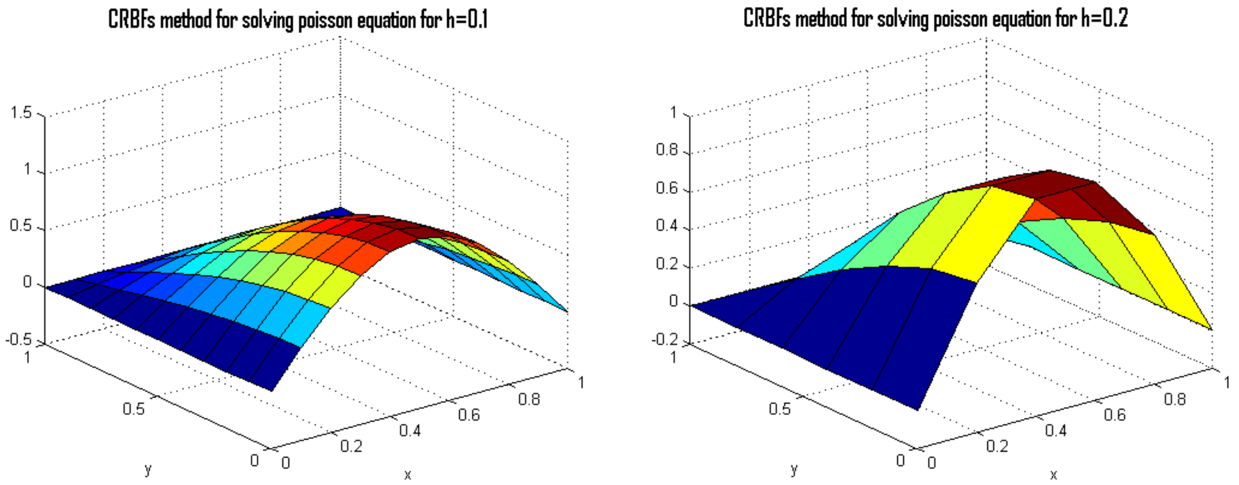

Figure 2.9: Poisson-CSRBF, with different values of T_{max}.

x/y	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0	0	0.005	0.007	0.004	0.003	0.0004	0.009	0.004	0.001	0.002
0.1	0.309	0.301	0.288	0.269	0.244	0.214	0.175	0.135	0.092	0.046
0.2	0.588	0.580	0.556	0.520	0.472	0.412	0.341	0.263	0.179	0.089
0.3	0.809	0.799	0.768	0.718	0.651	0.569	0.472	0.364	0.247	0.124
0.4	0.951	0.941	0.904	0.846	0.767	0.670	0.556	0.429	0.291	0.146
0.5	1.000	0.989	0.951	0.890	0.807	0.705	0.585	0.451	0.3071	0.154
0.6	0.951	0.941	0.904	0.846	0.768	0.670	0.557	0.429	0.292	0.146
0.7	0.809	0.800	0.769	0.720	0.653	0.570	0.473	0.365	0.2481	0.124
0.8	0.588	0.581	0.558	0.522	0.474	0.414	0.343	0.265	0.180	0.090
0.9	0.309	0.304	0.291	0.272	0.247	0.216	0.179	0.138	0.094	0.047
1	0	0	0	0	0	0	0	0	0	0

Table 2.30: The numerical results using CSRBF for solving Poisson's equation.

Remark 2.5. The numerical tests which we were able to establish here have proved that the method using CRBFs are very economical in terms of calculation time and programming compared to finite difference. Also it is very flexible and easy method to apply, the degree of precision does not depend on the choice of the function and the factor σ does not really have a great influence in the approximations.

GRBF for solving 2D time-dependent Schrodinger equation.

The time-dependent Schrodinger equation is a linear differential equation that governs the wave equation of a quantum-mechanical system.

Let us consider the two-dimensional time-dependent Schrodinger equation

$$-i\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + \omega(x,y)U, \tag{2.24}$$

in some continuous domain with suitable initial and Dirichlet boundary conditions, and an arbitrary potential function $\omega(x,y)$, with initial condition

$$U(x,y,0) = U_0(x,y),\ 0 \leq x,y \leq 1, \tag{2.25}$$

and boundary conditions

$$\begin{aligned} U(0,y,t) &= g_1(x,y,t), \qquad u(1,y,t) = g_2(x,y,t), \\ U(x,0,t) &= g_3(x,y,t), \quad u(x,1,t) = g_4(x,y,t),\ 0 \leq t \leq 1. \end{aligned} \tag{2.26}$$

Let $\phi(r)$ be the radial basis function, the approximation of a distribution $U(x)$ using radial basis function takes the form

$$U(x,y) \simeq \sum_{i=0}^{N}\sum_{j=0}^{M} c_{ij}\phi_{ij}(x,y) + \Upsilon(x,y),\ (x,y) \in \mathbb{R}^2, \tag{2.27}$$

where $\phi_{ij} = \phi_{ij}(x,y) = \phi(||(x,y)-(x_i,y_j)||)$, (x_i,y_j), $i = 0,\dots,N$, $j = 0,...,M$, here we are going to use generalized multiquadric radial basis function which is mentioned before.

If $\mathbb{P}_q^d$ denotes the space of d-variate polynomials of order not exceeding than q, and letting the polynomials $P_1, P_2, P_3, \dots, P_m$ be the basis of $\mathbb{P}_q^d$ in $\mathbb{R}^d$, then the polynomial $\Upsilon(x,y)$ is written in the form

$$\Upsilon(x,y) = \sum_{i=1}^{m} a_i P_i(x,y), \tag{2.28}$$

where $m = (q-1+d)!/(d!(q-1)!)$. Collocating equation (2.27) at $N \times M$ points, and m equations are required as

$$\sum_{i=0}^{N}\sum_{j=0}^{M} \lambda_j P_i(x,y) = 0,\ i = 1,\dots m. \tag{2.29}$$

We determine the unknown coefficients $(\lambda_1,\dots,\lambda_{NM})$ and $(a_1,\dots,a_m)$. If $\mathbb{L}$ is a linear differential operator then the approximation $\mathbb{L}U$ is given by

$$\mathbb{L}U = \sum_{i=0}^{N}\sum_{j=0}^{M} c_{ij}\mathbb{L}\Phi_{ij}(x,y) + \mathbb{L}\Upsilon(x,y). \tag{2.30}$$

In our approach, we are going to use Crank-Nicolson scheme.

Lets discretize equation (2.24) as follows

$$-i\frac{U(x,y,t+dt)-U(x,y,t)}{dt} = \theta\left[\nabla^2 U(x,y,t+dt)+\omega(x,y)U(x,y,t+dt)\right]$$
$$+(1-\theta)\left[\nabla^2 U(x,y,t)+\omega(x,y)U(x,y,t)\right],$$

where $\theta = \frac{1}{2}$, dt is the time step size. Using the notation $U^n = U(x,y,t^n)$, where $t^{n+1} = t^n + dt$, we obtain

$$-iU^{n+1} - \theta dt\left[\nabla^2 U^{n+1}+\omega(x,y)U^{n+1}\right] = -iU^n + (1-\theta)dt\left[\nabla^2 U^n + \omega(x,y)U^n\right] \tag{2.31}$$

Assuming that there are a total of $N \times M$ interpolation points, and let $\Upsilon(x,y) = 1 + x + y$, $U(x,y,t^n)$ can be approximated by

$$U^n(x_i,y_j) \simeq \sum_{j=1}^{N\times M} \lambda_j^n \phi(||x-x_i, y-y_j||) + \lambda^n_{N\times M+1}x_i + \lambda^n_{N\times M+2}y_j + \lambda^n_{N\times M+3}. \tag{2.32}$$

The additional condition (2.29) can be written as

$$\sum_{j=1}^{N\times M} \lambda_j^n = 0, \tag{2.33}$$

$$\sum_{j=1}^{N\times M} \lambda_j^n x_j = 0, \tag{2.34}$$

$$\sum_{j=1}^{N\times M} \lambda_j^n y_j = 0. \tag{2.35}$$

Writting equation (2.32) together with equation (2.33), equation (2.34), and equation (2.35), we obtain a matrix system $U^n = A\lambda^n$, where, $U^n = [U_1^n, U_2^n, \dots U^n_{N\times M}, 0, 0, 0]^T$, $\lambda^n = [\lambda_1^n, \lambda_2^n, \dots, \lambda^n_{N\times M+3}]^T$, and the matrix A is defined as

$$A = [a_{ij}, 1 \le i,j \le N\times M+3] = \begin{bmatrix} \phi_{11} & \phi_{12} & \dots & \phi_{1(N\times M)} & x_1 & y_1 & 1 \\ \phi_{21} & \phi_{22} & \dots & \phi_{2(N\times M)} & x_1 & y_1 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \phi_{(N\times M)1} & \phi_{(N\times M)2} & \dots & \phi_{(N\times M)(N\times M)} & x_1 & y_1 & 1 \\ 1 & 1 & \dots & \dots & 0 & 0 & 0 \\ x_1 & x_2 & \dots & \dots & 0 & 0 & 0 \\ y_1 & y_2 & \dots & \dots & 0 & 0 & 0 \end{bmatrix}. \tag{2.36}$$

Suppose that there are k internal points and $N\times M-k$ boundary points, then the $(N\times M+3$, $N\times M+3)$ matrix A can be divided into

$$A = B + C + D, \tag{2.37}$$

where,

B=$[b_{ij},$ for $(1 \le i \le k, 1 \le j \le N \times M + 3)$ and else 0],

C=[$c_{ij},$ for $(k+1 \le i \le N \times M, 1 \le j \le N \times M + 3)$ and else 0],

D=$[d_{ij},$ for $(N \times M + 1 \le i \le N \times M + 3, 1 \le j \le N \times M + 3)$ and else 0].

Equation (2.31) can be written in the following form

$$\left[-iB - \theta dt \left(\nabla^2 B + W * B\right) + C + D\right] \lambda^{n+1} = \left[-iB + (1-\theta)dt \left(\nabla^2 B + W * B\right)\right] \lambda^n + G^{n+1}, \tag{2.38}$$

where $G^n = [0 \ldots 0\, G^n_{k+1} \, \ldots \, G^n_{N \times M}\, 0\, 0\, 0]^T$, and $W = [\omega_1\, \omega_2\, \omega_k\, 0 \, \ldots \, 0]^T$.

Using the notation

$$\lambda^n = \lambda^n_r + i\lambda^n_i, \tag{2.39}$$

and

$$G^n = G^n_r + iG^n_i. \tag{2.40}$$

Equation (2.38) can be written as

$$E\lambda^{n+1}_r + B\lambda^{n+1}_i + i\left(E\lambda^{n+1}_i - B\lambda^{n+1}_r\right) = F\lambda^{n+1}_r + B\lambda^n_i + G^{n+1}_r + i\left(-B\lambda^n_r + F\lambda^n_i + G^{n+1}_i\right), \tag{2.41}$$

such that: $E = -\theta dt \left(\nabla^2 B + W * B\right) + C + D$, $F = (1-\theta)dt \left(\nabla^2 B + W * B\right)$.

Equation (2.41), can be written in the following real variable form

$$\begin{pmatrix} E & B \\ -B & E \end{pmatrix} \begin{pmatrix} \lambda^{n+1}_r \\ \lambda^{n+1}_i \end{pmatrix} = \begin{pmatrix} F & B \\ -B & F \end{pmatrix} \begin{pmatrix} \lambda^n_r \\ \lambda^n_i \end{pmatrix} + \begin{pmatrix} G^{n+1}_r \\ G^{n+1}_i \end{pmatrix}.$$

Thus, the solution of complex system has been reduced to solving the real variable system. Since the coefficients matrix is unchanged in time steps.

Example 2.14. Lets given Schrodinger equation with a potential function

$$\omega(x,y) = 1 - \frac{2}{x^2} - \frac{2}{y^2},$$

with initial condition

$$u(x,y,0) = x^2y^2,\ 0 \le x, y \le 1,$$

and boundary condition

$$u(0,y,t) = 0,\ u(1,y,t) = y^2 exp(it),$$
$$u(x,0,t) = 0,\ u(x,1,t) = x^2 exp(it),$$

The exact solution is

$$u(x,y,t) = x^2y^2 exp(it).$$

For $dx = dy = 0.2$, $dt = 0.0005$ and $0 \le t \le 1$. The obtained results with ($\beta = \frac{1}{2}$, $\epsilon = 0.79$) , are sitting in table 2.31, and figure 2.10, for ($\beta = 1.03$, $\epsilon = 0.82$) are presented in table 2.32, for ($\beta = 1.99$, $\epsilon = 0.8$) are given in table 2.33, and for ($\beta = 2.5$, $\epsilon = 1$) are shown in table 2.34.

(s,t)	Real part	Imaginary part
$(0,0)$	1.5180×10^{-10}	5.0590×10^{-11}
$(0.01,0.01)$	5.8344×10^{-5}	2.8995×10^{-7}
$(0.02,0.02)$	8.8344×10^{-5}	9.9016×10^{-7}
$(0.03,0.03)$	8.0440×10^{-5}	2.1778×10^{-6}
$(0.04,0.04)$	5.8854×10^{-5}	3.6819×10^{-6}
$(0.05,0.05)$	2.4097×10^{-5}	5.3906×10^{-6}
$(0.06,0.06)$	1.8323×10^{-5}	7.1931×10^{-6}
$(0.07,0.07)$	6.3652×10^{-5}	8.9826×10^{-6}
$(0.08,0.08)$	1.0789×10^{-4}	1.0659×10^{-5}
$(0.09,0.09)$	1.4778×10^{-4}	1.2131×10^{-5}

(s,t)	Real part	Imaginary part
$(0,0)$	2.9422×10^{-10}	3.5695×10^{-11}
$(0.01,0.01)$	7.8184×10^{-5}	2.3817×10^{-5}
$(0.02,0.02)$	1.2550×10^{-4}	4.7061×10^{-5}
$(0.03,0.03)$	1.4855×10^{-4}	6.8836×10^{-5}
$(0.04,0.04)$	1.5321×10^{-4}	8.8157×10^{-5}
$(0.05,0.05)$	1.4459×10^{-4}	1.0393×10^{-4}
$(0.06,0.06)$	1.2710×10^{-4}	1.1495×10^{-4}
$(0.07,0.07)$	1.0444×10^{-4}	1.1987×10^{-4}
$(0.08,0.08)$	7.9664×10^{-5}	1.1720×10^{-4}
$(0.09,0.09)$	5.5231×10^{-5}	1.0529×10^{-4}

Table 2.31: Computed errors using GRBFs method for Example (2.14) $\Big($ t=0.0005(left), t=0.1(right)$\Big)$, with $\beta = \frac{1}{2}$.

(s,t)	Real part	Imaginary part
$(0,0)$	1.6302×10^{-5}	2.8483×10^{-8}
$(0.01,0.01)$	2.0211×10^{-4}	3.9188×10^{-4}
$(0.02,0.02)$	2.5086×10^{-6}	6.6224×10^{-4}
$(0.03,0.03)$	5.3878×10^{-4}	7.2407×10^{-4}
$(0.04,0.04)$	1.3770×10^{-3}	7.2956×10^{-4}
$(0.05,0.05)$	2.4670×10^{-3}	6.7126×10^{-4}
$(0.06,0.06)$	3.7639×10^{-3}	5.8079×10^{-4}
$(0.07,0.07)$	5.5223×10^{-3}	4.8851×10^{-4}
$(0.08,0.08)$	6.8026×10^{-3}	4.2301×10^{-4}
$(0.09,0.09)$	8.4609×10^{-3}	4.1066×10^{-4}

(s,t)	Real part	Imaginary part
$(0,0)$	3.8562×10^{-4}	8.3182×10^{-7}
$(0.01,0.01)$	5.4510×10^{-3}	1.0829×10^{-2}
$(0.02,0.02)$	1.3643×10^{-2}	2.2474×10^{-2}
$(0.03,0.03)$	2.5344×10^{-2}	3.4702×10^{-2}
$(0.04,0.04)$	4.0926×10^{-2}	4.7229×10^{-2}
$(0.05,0.05)$	6.0744×10^{-2}	5.9722×10^{-2}
$(0.06,0.06)$	8.5136×10^{-2}	7.1798×10^{-2}
$(0.07,0.07)$	1.1441×10^{-2}	8.3029×10^{-2}
$(0.08,0.08)$	1.4883×10^{-1}	9.2943×10^{-2}
$(0.09,0.09)$	1.8863×10^{-1}	1.0103×10^{-1}

Table 2.32: Computed errors using GRBFs method for Example (2.14) $\Big($ t=0.0005(left), t=0.0375(right)$\Big)$, with $\beta = 1.03$.

(s,t)	Real part	Imaginary part
$(0,0)$	1.7616×10^{-4}	8.8295×10^{-8}
$(0.01,0.01)$	1.5103×10^{-4}	6.8656×10^{-5}
$(0.02,0.02)$	3.5086×10^{-4}	9.0989×10^{-5}
$(0.03,0.03)$	4.4470×10^{-4}	7.4369×10^{-5}
$(0.04,0.04)$	4.5312×10^{-4}	2.5827×10^{-5}
$(0.05,0.05)$	3.9576×10^{-4}	4.7919×10^{-5}
$(0.06,0.06)$	2.9108×10^{-4}	1.4052×10^{-4}
$(0.07,0.07)$	1.5626×10^{-4}	2.4604×10^{-4}
$(0.08,0.08)$	7.0203×10^{-7}	3.5901×10^{-4}
$(0.09,0.09)$	1.4253×10^{-4}	4.7448×10^{-4}

(s,t)	Real part	Imaginary part
$(0,0)$	1.0096×10^{-3}	1.1330×10^{-6}
$(0.01,0.01)$	8.6039×10^{-4}	3.6595×10^{-4}
$(0.02,0.02)$	3.1837×10^{-3}	4.1454×10^{-4}
$(0.03,0.03)$	5.9365×10^{-3}	1.3490×10^{-4}
$(0.04,0.04)$	9.0901×10^{-3}	4.8421×10^{-4}
$(0.05,0.05)$	1.2611×10^{-3}	1.4530×10^{-3}
$(0.06,0.06)$	1.6459×10^{-2}	2.7807×10^{-3}
$(0.07,0.07)$	2.0592×10^{-2}	4.4748×10^{-3}
$(0.08,0.08)$	2.4962×10^{-2}	6.5414×10^{-3}
$(0.09,0.09)$	2.9516×10^{-2}	8.9847×10^{-3}

Table 2.33: Computed errors using GRBFs method for Example (2.14) $\Big($ t=0.0005(left), t=0.1(right)$\Big)$, with $\beta = 1.99$.

(s,t)	Real part	Imaginary part
$(0,0)$	5.1195×10^{-5}	2.6510×10^{-8}
$(0.01,0.01)$	2.0942×10^{-4}	1.0238×10^{-4}
$(0.02,0.02)$	3.2546×10^{-4}	1.3338×10^{-4}
$(0.03,0.03)$	4.0492×10^{-4}	1.0309×10^{-4}
$(0.04,0.04)$	4.5309×10^{-4}	2.1050×10^{-5}
$(0.05,0.05)$	4.7491×10^{-4}	1.0377×10^{-4}
$(0.06,0.06)$	4.7498×10^{-4}	2.6301×10^{-4}
$(0.07,0.07)$	4.5753×10^{-4}	4.4892×10^{-4}
$(0.08,0.08)$	4.2644×10^{-4}	6.5442×10^{-4}
$(0.09,0.09)$	3.8523×10^{-4}	8.7305×10^{-4}

(s,t)	Real part	Imaginary part
$(0,0)$	3.7432×10^{-3}	5.0240×10^{-8}
$(0.01,0.01)$	1.5549×10^{-3}	2.3129×10^{-3}
$(0.02,0.02)$	1.4707×10^{-3}	3.7799×10^{-3}
$(0.03,0.03)$	3.1897×10^{-3}	4.5546×10^{-3}
$(0.04,0.04)$	6.4242×10^{-3}	4.7795×10^{-3}
$(0.05,0.05)$	1.0901×10^{-2}	4.5856×10^{-3}
$(0.06,0.06)$	1.6364×10^{-2}	4.0924×10^{-3}
$(0.07,0.07)$	2.2570×10^{-2}	3.4074×10^{-3}
$(0.08,0.08)$	2.9295×10^{-2}	2.6267×10^{-3}
$(0.09,0.09)$	3.6333×10^{-2}	1.8345×10^{-3}

Table 2.34: Computed errors using GRBFs method for Example (2.14) $\Big($ t=0.0005(left), t=0.05(right)$\Big)$, with $\beta = 2.5$.

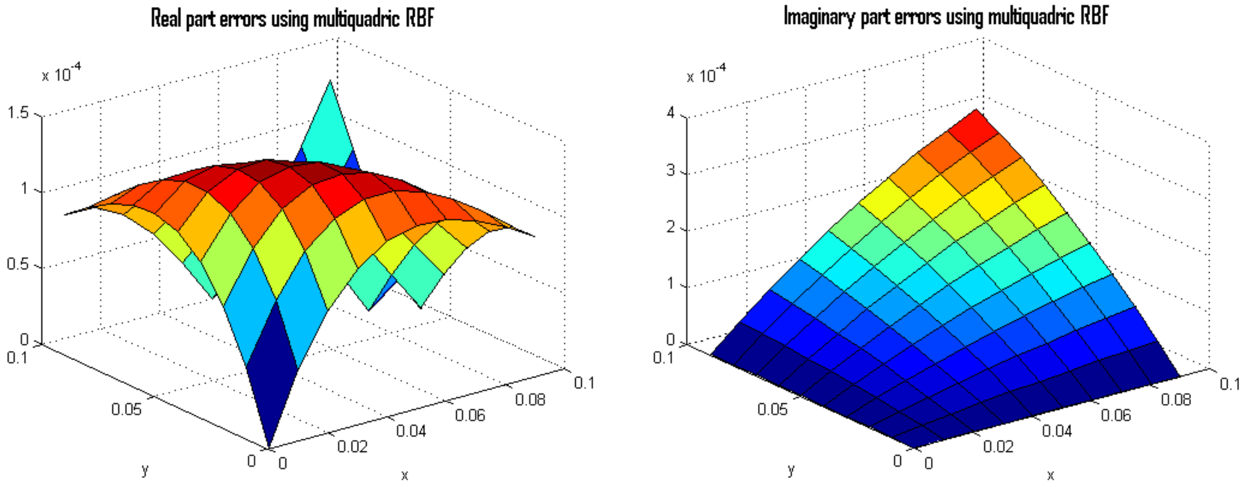

Figure 2.10: The Absolute errors , for $t = 0.05$, real part(left), imaginary part(right), with $\beta = \frac{1}{2}$.

Example 2.15. Let us given Schrodinger equation with a potential function

$$\omega(x, y) = 0,$$

with initial condition

$$u(x, y, 0) = e^{-x^2-y^2-c_0x}, \ -2 \leq x, y \leq 2,$$

and boundary condition

$$u(0, y, t) = \frac{i}{i-4t}e^{i((y^2+ic_0^2t)/(i-4t))}, \ u(1, y, t) = \frac{i}{i-4t}e^{i((1+y^2+ic_0+ic_0^2t)/(i-4t))},$$

$$u(x, 0, t) = \frac{i}{i-4t}e^{i((x^2+ic_0x+ic_0^2t)/(i-4t))},$$

$$u(x, 1, t) = \frac{i}{i-4t}e^{i((x^2+1+ic_0x+ic_0^2t)/(i-4t))},$$

The exact solution is given by

$$u(x, y, t) = \frac{i}{i-4t}e^{i((x^2+y^2+ic_0x+ic_0^2t)/(i-4t))}, \ -2 \leq x, y \leq 2.$$

For $dx = dy = 0.5$, $dt = 0.0002$ and $0 \leq t \leq 1$. The obtained results with ($\beta = \frac{1}{2}$, $\epsilon = 0.55$), are sitting in table 2.35 and figure 2.11, for ($\beta = 1.03$, $\epsilon = 0.5$) are presented in table 2.36, for ($\beta = 1.99$, $\epsilon = 0.4$) are given in table 2.37, and for $\beta = 2.5$, $\epsilon = 0.4$ are shown in table 2.38.

(s,t)	Real part	Imaginary part
$(0,0)$	3.6724×10^{-6}	1.6642×10^{-4}
$(0.01,0.01)$	1.0966×10^{-3}	4.7663×10^{-2}
$(0.02,0.02)$	2.0815×10^{-3}	9.5420×10^{-2}
$(0.03,0.03)$	2.9561×10^{-3}	1.4302×10^{-2}
$(0.04,0.04)$	3.7196×10^{-3}	1.9039×10^{-1}
$(0.05,0.05)$	4.3727×10^{-3}	2.3743×10^{-1}
$(0.06,0.06)$	4.9175×10^{-3}	2.8408×10^{-1}
$(0.07,0.07)$	5.3571×10^{-3}	3.3026×10^{-1}
$(0.08,0.08)$	5.6961×10^{-3}	3.7587×10^{-1}
$(0.09,0.09)$	5.9399×10^{-3}	4.2086×10^{-1}

(s,t)	Real part	Imaginary part
$(0,0)$	7.1509×10^{-3}	1.4923×10^{-2}
$(0.01,0.01)$	4.0534×10^{-3}	3.0590×10^{-2}
$(0.02,0.02)$	1.5171×10^{-2}	7.6129×10^{-2}
$(0.03,0.03)$	2.6174×10^{-2}	1.2161×10^{-2}
$(0.04,0.04)$	3.7038×10^{-2}	1.6697×10^{-1}
$(0.05,0.05)$	4.7740×10^{-2}	2.1212×10^{-1}
$(0.06,0.06)$	5.8253×10^{-2}	2.5699×10^{-1}
$(0.07,0.07)$	6.8560×10^{-2}	3.0151×10^{-1}
$(0.08,0.08)$	7.8640×10^{-2}	3.4560×10^{-1}
$(0.09,0.09)$	8.8473×10^{-2}	3.8917×10^{-1}

Table 2.35: Computed errors using GRBFs method for Example (2.15) $\Big($ t=0.0002(left), t=0.016(right)$\Big)$, with $\beta = \frac{1}{2}$.

(s,t)	Real part	Imaginary part
$(0,0)$	1.8941×10^{-2}	3.1833×10^{-2}
$(0.01,0.01)$	2.0214×10^{-2}	1.6196×10^{-2}
$(0.02,0.02)$	2.1393×10^{-2}	6.4146×10^{-2}
$(0.03,0.03)$	2.2475×10^{-2}	8.1193×10^{-2}
$(0.04,0.04)$	2.3456×10^{-2}	1.5948×10^{-1}
$(0.05,0.05)$	2.4340×10^{-2}	2.0670×10^{-1}
$(0.06,0.06)$	2.5112×10^{-2}	2.5320×10^{-1}
$(0.07,0.07)$	2.5811×10^{-2}	2.9983×10^{-1}
$(0.08,0.08)$	2.6404×10^{-2}	3.4558×10^{-1}
$(0.09,0.09)$	2.6906×10^{-2}	3.9069×10^{-1}

(s,t)	Real part	Imaginary part
$(0,0)$	1.6881×10^{-2}	9.9516×10^{-2}
$(0.01,0.01)$	2.9550×10^{-2}	6.8410×10^{-1}
$(0.02,0.02)$	4.1877×10^{-2}	7.2821×10^{-1}
$(0.03,0.03)$	5.3834×10^{-2}	7.7160×10^{-1}
$(0.04,0.04)$	6.5395×10^{-2}	8.1418×10^{-1}
$(0.05,0.05)$	7.6539×10^{-2}	8.5591×10^{-1}
$(0.06,0.06)$	8.7242×10^{-2}	8.9673×10^{-1}
$(0.07,0.07)$	9.7487×10^{-2}	9.3665×10^{-1}
$(0.08,0.08)$	9.8216×10^{-2}	9.7532×10^{-1}
$(0.09,0.09)$	1.1654×10^{-1}	9.9801×10^{-1}

Table 2.36: Computed errors using GRBFs method for Example (2.15) $\Big($ t=0.0002(left), t=0.036(right)$\Big)$, with $\beta = 1.03$.

(s,t)	Real part	Imaginary part
$(0,0)$	6.4793×10^{-2}	4.2012×10^{-2}
$(0.01,0.01)$	6.5824×10^{-2}	6.6113×10^{-3}
$(0.02,0.02)$	6.6786×10^{-2}	5.5157×10^{-5}
$(0.03,0.03)$	6.7676×10^{-2}	1.0354×10^{-2}
$(0.04,0.04)$	6.8489×10^{-2}	1.5168×10^{-2}
$(0.05,0.05)$	6.9225×10^{-2}	1.9947×10^{-1}
$(0.06,0.06)$	6.9883×10^{-2}	2.4687×10^{-1}
$(0.07,0.07)$	7.0460×10^{-2}	2.9376×10^{-1}
$(0.08,0.08)$	7.0960×10^{-2}	3.4007×10^{-1}
$(0.09,0.09)$	7.1382×10^{-2}	3.8572×10^{-1}

(s,t)	Real part	Imaginary part
$(0,0)$	6.2007×10^{-2}	8.9252×10^{-2}
$(0.01,0.01)$	6.8899×10^{-2}	2.4920×10^{-1}
$(0.02,0.02)$	7.5646×10^{-2}	2.9639×10^{-1}
$(0.03,0.03)$	8.2229×10^{-2}	3.4311×10^{-1}
$(0.04,0.04)$	8.8633×10^{-2}	3.8931×10^{-1}
$(0.05,0.05)$	9.4839×10^{-2}	4.3488×10^{-1}
$(0.06,0.06)$	9.8896×10^{-2}	4.7977×10^{-1}
$(0.07,0.07)$	9.9901×10^{-2}	5.2390×10^{-1}
$(0.08,0.08)$	9.9201×10^{-2}	5.6719×10^{-1}
$(0.09,0.09)$	9.9316×10^{-2}	6.0956×10^{-1}

Table 2.37: Computed errors using GRBFs method for Example (2.15) $\Big($ t=0.0002(left), t=0.0172(right)$\Big)$, with $\beta = 1.99$.

(s,t)	Real part	Imaginary part
$(0,0)$	5.9901×10^{-5}	2.9172×10^{-3}
$(0.01,0.01)$	1.0905×10^{-3}	5.1498×10^{-2}
$(0.02,0.02)$	2.1802×10^{-3}	1.0000×10^{-2}
$(0.03,0.03)$	3.2048×10^{-3}	1.4836×10^{-1}
$(0.04,0.04)$	4.1605×10^{-3}	1.9646×10^{-1}
$(0.05,0.05)$	5.0445×10^{-3}	2.4424×10^{-1}
$(0.06,0.06)$	5.8549×10^{-3}	2.9160×10^{-1}
$(0.07,0.07)$	6.5901×10^{-3}	3.3846×10^{-1}
$(0.08,0.08)$	7.2495×10^{-3}	3.8474×10^{-1}
$(0.09,0.09)$	7.8333×10^{-3}	4.3036×10^{-1}

(s,t)	Real part	Imaginary part
$(0,0)$	3.5521×10^{-2}	2.2318×10^{-2}
$(0.01,0.01)$	4.1224×10^{-2}	2.7150×10^{-2}
$(0.02,0.02)$	4.6827×10^{-2}	3.1955×10^{-1}
$(0.03,0.03)$	5.2313×10^{-2}	3.6723×10^{-1}
$(0.04,0.04)$	5.7670×10^{-2}	4.1447×10^{-1}
$(0.05,0.05)$	6.2885×10^{-2}	4.6119×10^{-1}
$(0.06,0.06)$	6.7948×10^{-2}	5.0729×10^{-1}
$(0.07,0.07)$	7.2847×10^{-2}	5.5272×10^{-1}
$(0.08,0.08)$	7.7573×10^{-2}	5.9738×10^{-1}
$(0.09,0.09)$	8.2117×10^{-2}	6.4120×10^{-1}

Table 2.38: Computed errors using GRBFs method for Example (2.15) $\Big($ t=0.0002(left), t=0.018(right)$\Big)$, with $\beta = 2.5$.

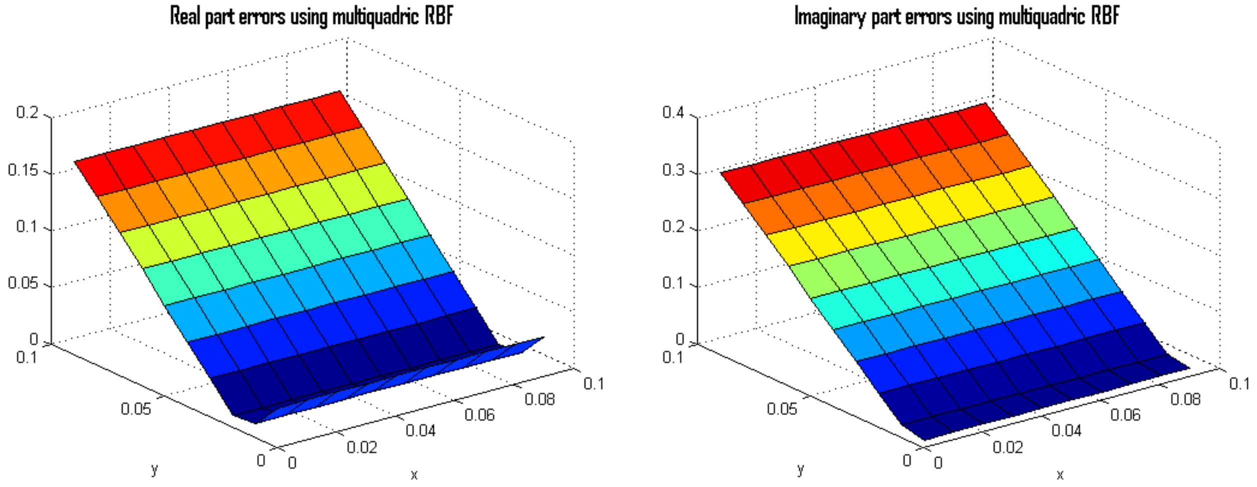

Figure 2.11: The Absolute errors , for $t = 0.4$, real part(left), imaginary part(right), with $\beta = \frac{1}{2}$.

2.3 Comparaison between stationary and non-stationary approaches in interpolating some one and two-dimensional functions

The convergence of RBF methods can be discussed in terms of two different types of approximation, stationary and non-stationary. In stationary approximation, the number of centers N is fixed and the shape parameter ϵ is refined towards zero. Non-stationary approximation fixes the value of the shape parameter and N is increased. The theoritical convergence rates may be difficult to achieve computationally due to the condition number of the resulting matrix growing with decreasing both fill distance and shape parameter. In the following, we analyse the efficiency and applicability of the two approaches for scattered data approximation by globally RBFs. In the following, we analyse the efficiency and applicability of the two approaches for scattered data approximation by globally RBFs.

The set of interpolation data points

The most known set of interpolation data points are the Chebyshev, Halton and Sobol points.

Chebyshev nodes

For a given positive integer n the Chebychev nodes in the interval $[-1, 1]$ are

$$x_k = cos\left(\frac{2k-1}{2n}\Pi\right),\ k = 1, \dots, n.$$

In our experiments we are going to use the shifted Chebychev nodes in $[0, 1]$ given by

$$x_k = \frac{1}{2} + \frac{1}{2}cos\left(\frac{2k-1}{2n}\Pi\right),\ k = 1, \dots, n.$$

Halton nodes

In statistics, Halton sequence are sequences used to generate points in space for numerical methods such as Monte Carlo simulations. Although these sequences are deterministic, they are of low discrepancy, that is appear to be random for many purposes. They were first introduced in 1960 and are an example of a quasi-random number sequence. They generalize the one-dimensional var der Corput sequence.

Sobol nodes

Sobol sequences are an example of quasi-random low-discrepancy sequences. They were first introduced by thr Russian mathematician Ilya M. Sobol in 1967. These sequences use a base of two to form successively finer uniform partitions of the unit interval and then reorder the coordinates in each dimension.

Numerical examples

Let $\mathcal{P}_f(x)$ be the interpolant of $f(x)$, in order to show the convergence accuracy of the radial basis function interpolation , we are going to calculate the absolute errors at the equally spaced points in $[0, 1]$ using the formula

$$e(x) = |\mathcal{P}_f(x) - f(x)|,\ 0 \leq x \leq 1.$$

Also, we have,

Root-mean-square error (RMS):

$$RMS\text{-error} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}[\mathcal{P}_f(x_i) - f(x_i)]^2} = \frac{1}{\sqrt{N}}||\mathcal{P}_f - f||_2, \tag{2.42}$$

where x_i, $i = 1, \dots, N$ are the evaluation points.

For our experiments we are going to use the following 2D and 1D Franke's test functions:

2D Franke's test functions

$f_1(x, y) = 0.75e^{(-(9x-2)^2 + \frac{(9y-2)^2}{4})}$, $x \in [0, 1] \times [0, 1]$.

$f_2(x, y) = 0.75e^{(-((9x+1)^2/49 + (9y+1)^2/10))}$, $x \in [0, 1] \times [0, 1]$.

$f_3(x, y) = sinc(x)sinc(y)$, $x \in [0, 1] \times [0, 1]$.

1D Franke's test functions

$f_1(x) = x^3 + x^2 + 1$, $x \in [0, 1]$.

$f_2(x) = sinc(x) + 1$, $x \in [0, 1]$.

2.3.1 Stationary approach

For 2D and 1D stationary approach, we fix the number of centers $N = 289$, $N = 17$ respectively. RMS-errors, Max-errors and absolute errors are computed for different values of shape parameter ϵ using multiquadric RBF and with Halton, Sobol and Chebyshev points. The numerical tests are given in tables 2.39-2.43 and figures 2.12-2.16.

Table 2.39: 2D stationary interpolation ($N = 289$) to the test function $f_1(x, y)$ with Multiquadric RBF using Halton, Sobol and Chebyshev points.

	Halton points		Sobol points		Chebychev points	
ϵ	**MAX**-error	**RMS**-error	**MAX**-error	**RMS**-error	**MAX**-error	**RMS**-error
0.2	8.2487e-03	5.2283e-04	9.7009e-03	1.3026e-03	2.5595e-03	5.5118e-04
0.7	1.8078e-04	13389e-05	3.9934e-03	4.3877e-04	2.8352e-03	7.9621e-04
1.2	3.3123e-02	2.3527e-03	8.2889e-02	1.1020e-02	2.7342e-03	8.7432e-04
1.6	8.8136e-02	9.0501e-03	4.0381e-01	3.8320e-02	2.6120e-03	8.7284e-04
3	4.5229e-01	1.4229e-01	6.6016e-01	7.8839e-02	.2.7617e-01	1.699e-01

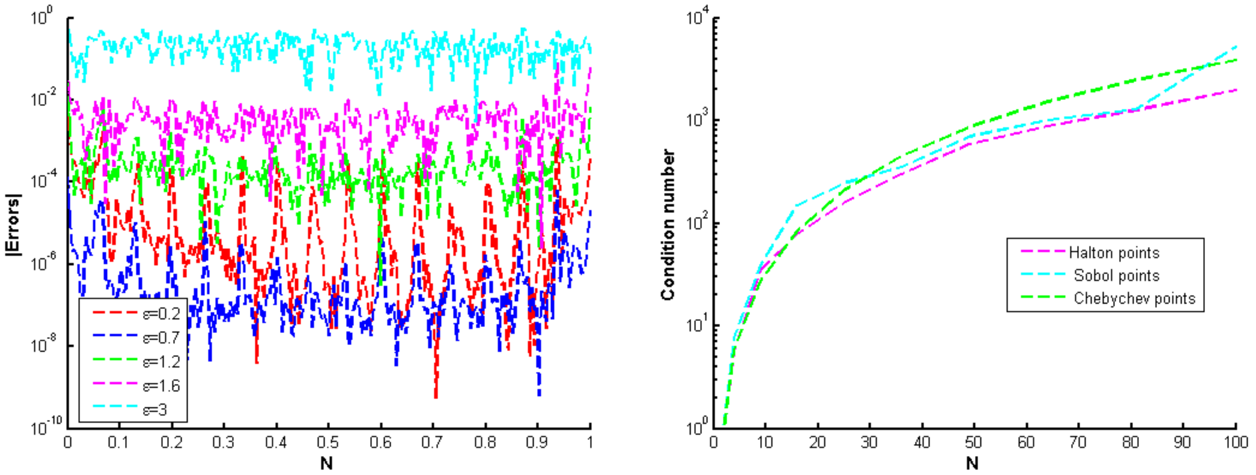

Figure 2.12: Absolute errors for stationary interpolation using Halton points to the test function $f_1(x, y)$ with multiquadric RBF based on $M = 225$ uniformly spaced points in $[0, 1]$ and ϵ= 0.2,0.7,1.2,1.6,3, (left), the condition number (right).

Table 2.40: 2D stationary interpolation ($N = 289$) to the test function $f_2(x, y)$ with multiquadric RBF using Halton, Sobol and Chebyshev points.

	Halton points		Sobol points		Chebyshev points	
ϵ	**MAX**-error	**RMS**-error	**MAX**-error	**RMS**-error	**MAX**-error	**RMS**-error
0.2	4.3052e-03	4.1163e-04	7.0019e-04	1.2867e-04	8.7577e-05	1.9101e-05
0.7	1.0122e-04	7.1236e-06	9.9876e-05	8.1483e-07	2.2902e-06	6.91147e-07
1.2	4.0902e-05	3.9870e-06	1.0606e-05	1.3681e-06	1.40252e-06	8.0095e-07
1.6	8.2537e-05	84913e-06	2.2973e-04	3.5719e-05	4.5406e-05	2.3142e-05
3	1.1461e-02	2.8493e-03	7.9248e-03	3.0039e-03	3.5045e-03	1.3482e-03

Table 2.41: 2D stationary interpolation ($N = 289$) to the test function $f_3(x, y)$ with Multiquadric RBF using Halton, Sobol and Chebyshev points.

	Halton points		Sobol points		Chebyshev points	
ϵ	**MAX**-error	**RMS**-error	**MAX**-error	**RMS**-error	**MAX**-error	**RMS**-error
0.2	6.5239e-03	5.433e-04	8.6697e-04	11483e-04	4.1378e-05	1.3596e-05
0.7	2.1649e-05	1.3475e-06	2.0369e-06	2.5393e-07	7.53968e-08	2.42744e-08
1.2	1.9017e-06	2.6104e-07	1.2744e-06	1.8414e-07	8.1136e-08	5.45705e-08
1.6	1.1837e-06	1.1038e-07	2.4734e-06	5.2780e-07	1.7337e-07	7.8064e-08
3	3.0919e-06	6.9209e-07	2.0032e-05	8.4621e-06	1.0456e-06	4.0600e-07

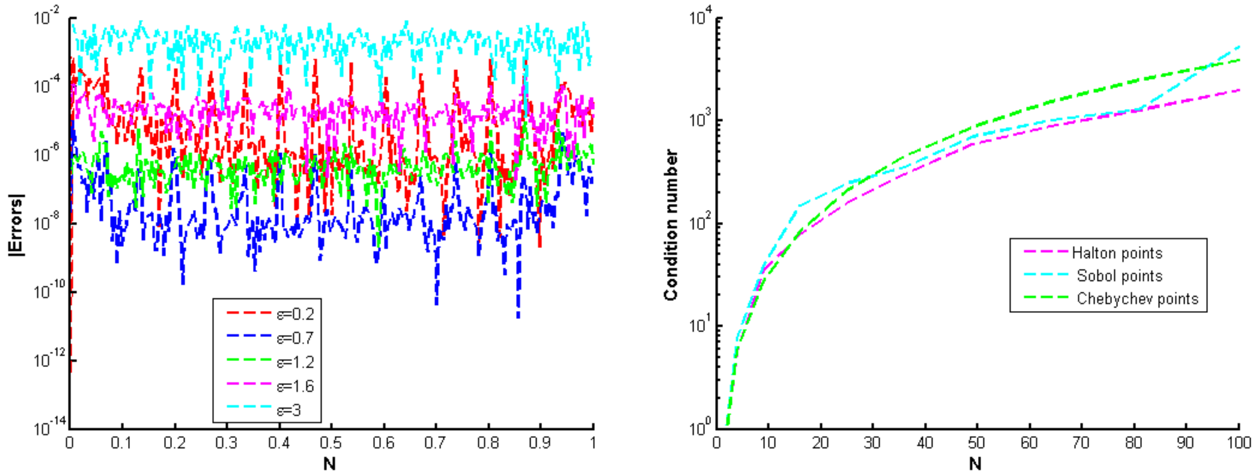

Figure 2.13: Absolute errors for stationary interpolation with Sobol points the test function $f_2(x, y)$ with multiquadric RBF based on $M = 225$ uniformly spaced points in $[0, 1]$ and ϵ= 0.2,1.7,1.2,1.6,3, (left), the condition number (right).

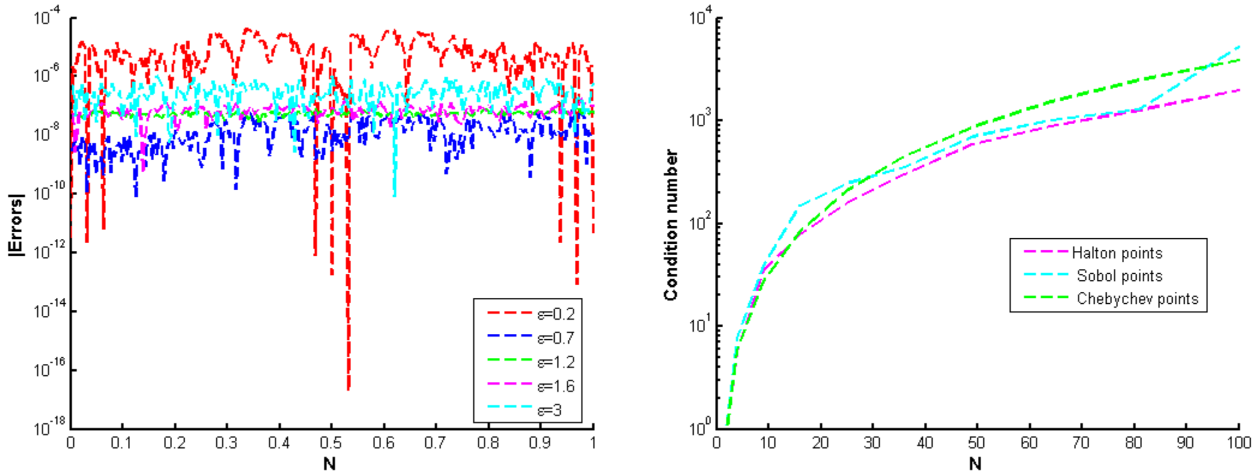

Figure 2.14: Absolute errors for stationary interpolation using Chebyshev to the test function $f_3(x, y)$ with multiquadric RBF based on $M = 225$ uniformly spaced points in $[0, 1]$ and ϵ= 0.2,1.7,1.2,1.6,3, (left), the condition number (right).

Table 2.42: 1D stationary interpolation ($N = 17$) to the test function $f_1(x)$ with Multiquadric RBF using Halton, Sobol and Chebyshev points.

	Halton points		Sobol points		Chebyshev points	
ϵ	**MAX**-error	**RMS**-error	**MAX**-error	**RMS**-error	**MAX**-error	**RMS**-error
0.2	2.1189e-04	6.0906e-05	1.3000e-02	3.1981e-04	1.9005e-04	8.2560e-05
0.6	1.4190e-04	1.1925e-04	8.7795e-05	4.5640e-05	8.9861e-04	8.2265e-04
1.5	1.9652e-04	1.7681e-04	4.3145e-04	3.2300e-04	3.4495e-04	2.8935e-04
4	3.3761e-04	3.3333e-04	5.6112e-04	5.5600e-04	3.2464e-04	3.2006e-04
9	5.5214e-05	3.4000e-05	5.5765e-05	3.8200e-05	5.1522e-05	3.4852e-05

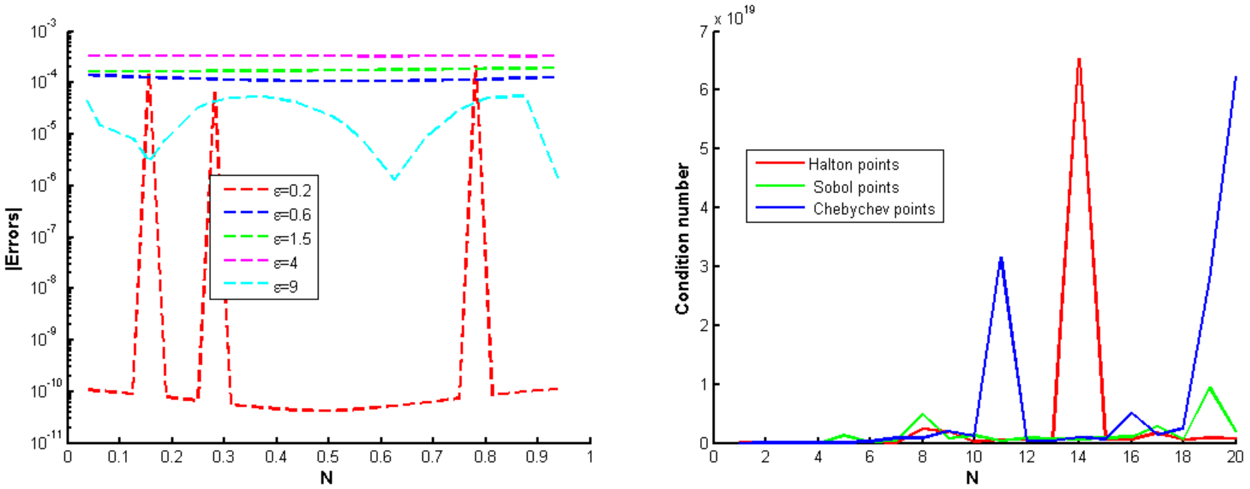

Figure 2.15: Absolute errors for stationary interpolation using Halton points to the test function $f_1(x)$ with multiquadric RBF based on $M = 20$ uniformly spaced points in $[0, 1]$ and ϵ= 0.2,0.6,1.5,4,9 , (left), the condition number (right).

Table 2.43: 1D stationary interpolation ($N = 17$) to the test function $f_2(x)$ with multiquadric RBF using Halton, Sobol and Chebyshev points.

	Halton points		Sobol points		Chebyshev points	
ϵ	**MAX**-error	**RMS**-error	**MAX**-error	**RMS**-error	**MAX**-error	**RMS**-error
0.2	1.3773e-04	4.3354-05	6.7480e-04	1.7075e-04	7.0838e-05	4.0763e-05
0.6	1.5740e-04	1.2784e-04	3.1620e-04	2.3401e-04	1.6270e-04	1.2083e-04
1.5	1.3713e-04	1.0902e-04	1.5000e-03	1.2000e-03	5.7895e-05	4.2606e-05
4	6.0397e-05	5.8292e-05	2.0285-04	2.0190e-04	1.5183e-05	1.1386e-05
9	5.0826e-04	3.0552e-04	7.9172e-04	5.3070e-04	7.0278e-04	4.9092e-04

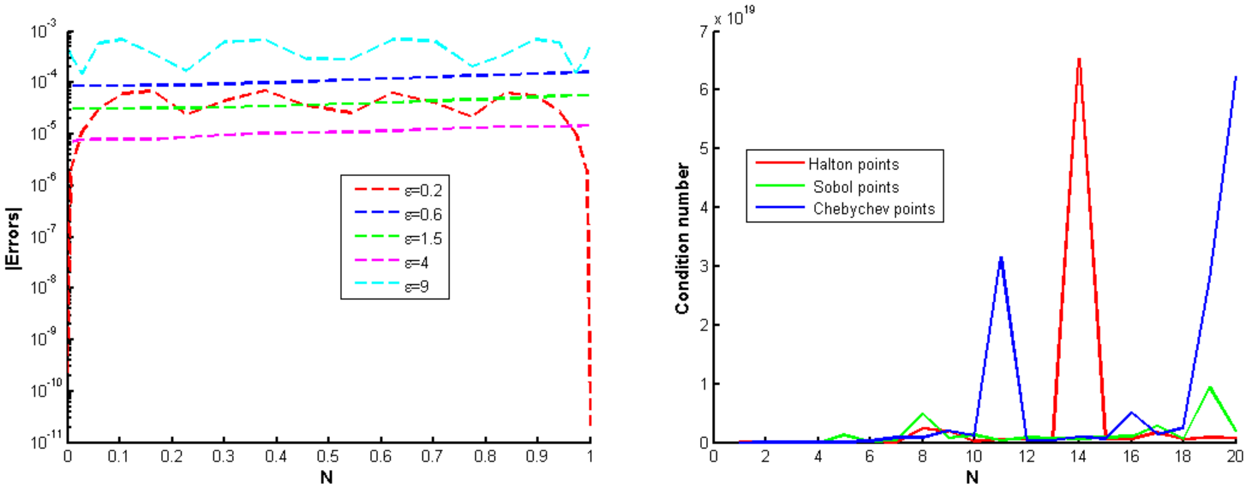

Figure 2.16: Absolute errors for stationary interpolation with Chebyshev points to the test function $f_2(x)$ with multiquadric RBF based on $M = 20$ uniformly spaced points in $[0, 1]$ and ϵ= 0.2,0.6,1.5,4,9, (left), the condition number (right).

2.3.2 Non-stationary approach

For 2D and 1D non-stationary approach, we fix the value of shape parameter ϵ (obtained using different strategies). RMS-errors, Max-errors and absolute errors are computed for different values of N using multiquadric RBF and with Halton, Sobol and Chebyshev points. The numerical tests are given in tables (2.44-2.48) and figures 2.17-2.21.

Table 2.44: 2D non-stationary interpolation to the test function $f_1(x, y)$ with multiquadric RBF using Halton, Sobol and Chebyshev points, with $\epsilon = 0.404$ based on leave-one-out cross validation strategy.

	Halton points		Sobol points		Chebyshev points	
N	**MAX**-error	**RMS**-error	**MAX**-error	**RMS**-error	**MAX**-error	**RMS**-error
9	5.1315e-01	1.0457e-01	.2.9910e-01	8.6722e-02	6.7251e-01	1.6202e-01
25	2.9913e-01	4.9196e-02	9.2517e-02	2.1972e-02	7.4536e-02	1.6952e-02
81	4.4413e-02	3.6867e-03	4.3246e-03	6.9907e-04	4.7335e-03	1.4676e-03
289	3.1021e-04	2.1030e-05	2.4821e-05	1.9365e-06	1.9380e-06	4.3116e-07
1089	8.9652e-06	6.0805e-07	1.5201e-07	2.8906e-08	6.1803e-09	2.4224e-09

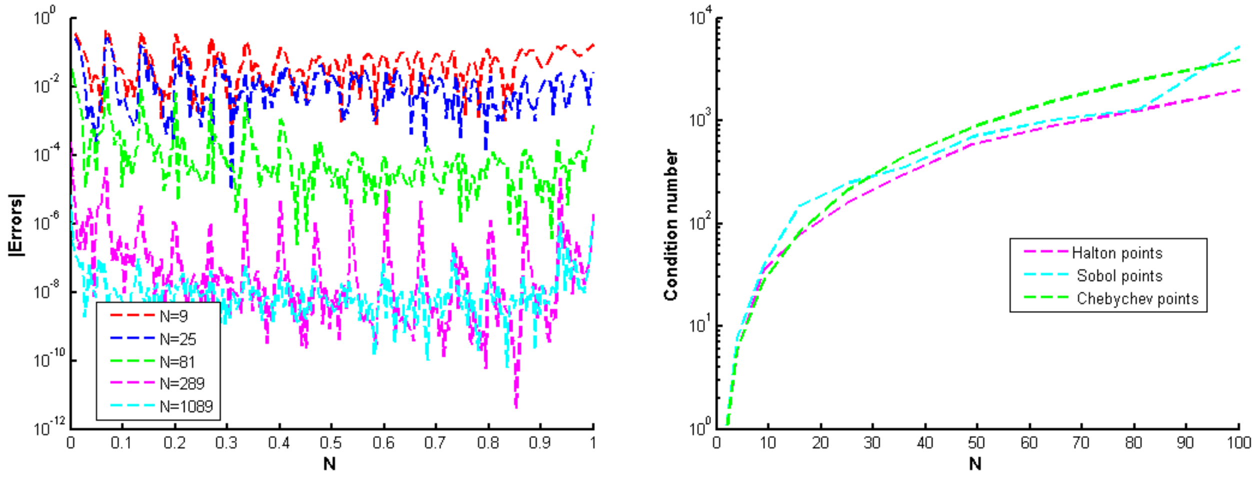

Figure 2.17: Absolute errors for non-stationary interpolation with $\epsilon = 0.404$ using Halton points to the test function $f_1(x, y)$ with multiquadric RBF based on $M = 225$ uniformly spaced points in $[0, 1]$ and $N = 9, 25, 81, 289, 1089$, (left), the condition number (right).

Table 2.45: 2D non-stationary interpolation to the test function $f_2(x, y)$ with Multiquadric RBF using Halton, Sobol and Chebyshev points, with $\epsilon = 0.5$ based on power functions strategy.

	Halton points		Sobol points		Chebyshev points	
N	**MAX**-error	**RMS**-error	**MAX**-error	**RMS**-error	**MAX**-error	**RMS**-error
9	2.8640e-01	4.9658e-02	1.0342e-01	2.9548e-02	7.0008e-02	2.5809e-02
25	6.1532e-02	8.5300e-03	4.4339e-02	7.7823e-03	2.0312e-02	6.5735e-03
81	5.3381e-03	5.2796e-04	5.4857e-03	7.0515e-04	7.6353e-04	2.3434e-04
289	3.7393e-04	2.5547e-05	6.2459e-05	5.0461e-06	2.2616e-06	5.3155e-07
1089	1.7536e-05	1.4928e-06	9.3336e-07	1.1966e-07	4.13205e-08	1.9331e-08

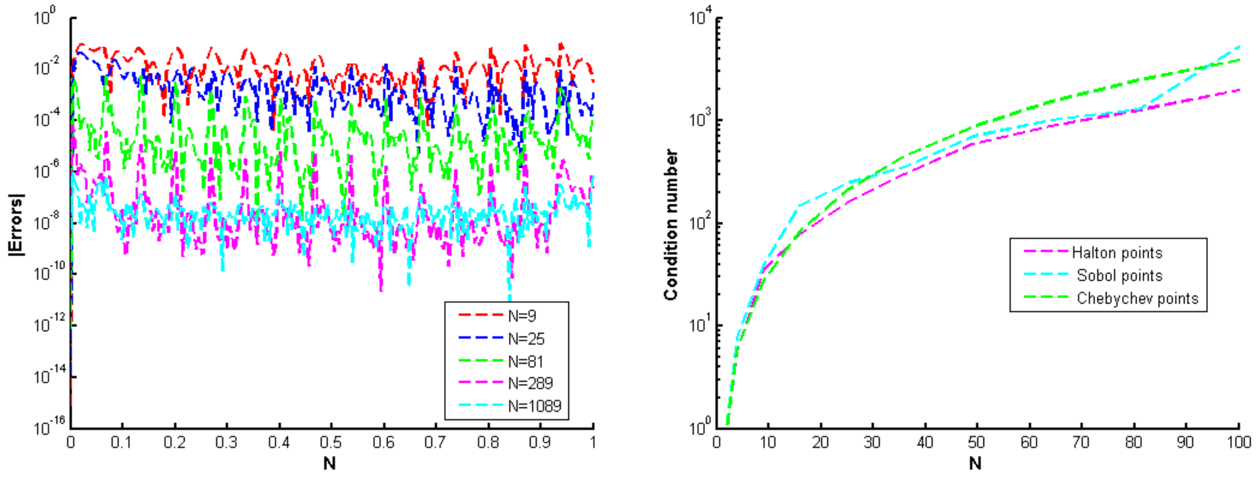

Figure 2.18: Absolute errors for non-stationary interpolation with $\epsilon = 0.5$ using Sobol points to the test function $f_2(x, y)$ with multiquadric RBF based on $M = 225$ uniformly spaced points in $[0, 1]$ and $N = 9, 25, 81, 289, 1089$, (left), the condition number (right).

Table 2.46: 2D non-stationary interpolation to the test function $f_3(x, y)$ with multiquadric RBF using Halton, Sobol and Chebyshev points, with $\epsilon = 1.6$ based on linear variable shape parameter strategy.

	Halton points		Sobol points		Chebyshev points	
N	**MAX**-error	**RMS**-error	**MAX**-error	**RMS**-error	**MAX**-error	**RMS**-error
9	2.6950e-02	6.3979e-03	8.0187e-01	6.2289e-02	3.14308e-02	1.3940e-02
25	3.0421e-03	5.8786e-04	5.1026e-02	2.9727e-03	7.7250e-04	2.9675e-04
81	4.5207e-05	3.8551e-06	3.8587e-04	1.37e-05	6.6310e-07	2.67398e-07
289	1.8378e-06	1.1038e-07	1.3422e-05	6.1619e-06	1.73378e-07	7.8064e-08
1089	1.0175e-06	3.7494e-07	3.7494e-07	6.1646e-08	4.90822e-07	1.7926e-08

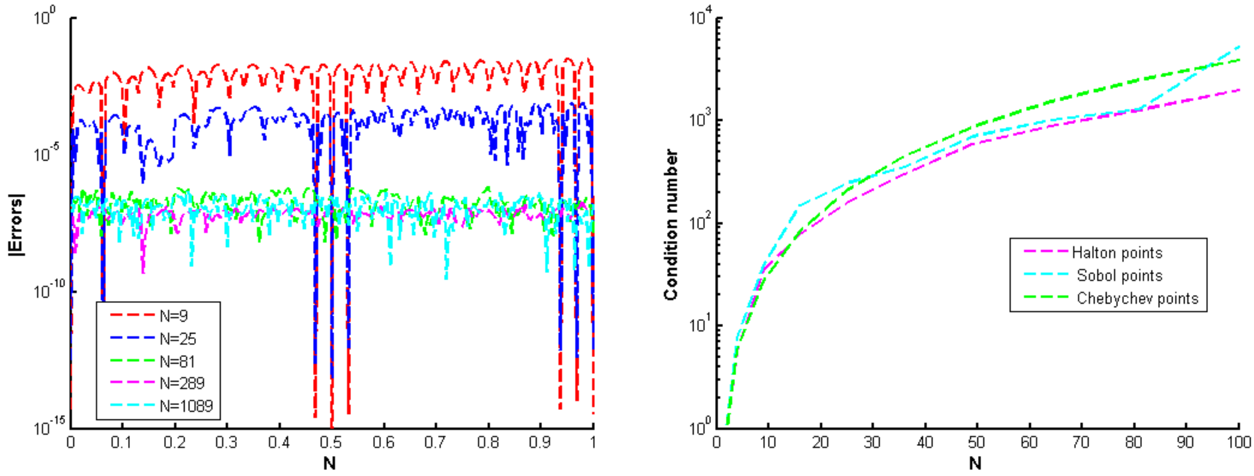

Figure 2.19: Absolute errors for non-stationary interpolation with $\epsilon = 1.6$ using Chebyshev points to the test function $f_3(x, y)$ with multiquadric RBF based on $M = 225$ uniformly spaced points in $[0, 1]$ and $N = 9, 25, 81, 289, 1089$, (left), the condition number (right).

Table 2.47: 1D non-stationary interpolation ($\epsilon = 1.1818$) obtained using LOOCV strategy to the test function $f_1(x)$ with multiquadric RBF using Halton, Sobol and Chebyshev points.

	Halton points		Sobol points		Chebyshev points	
N	**MAX**-error	**RMS**-error	**MAX**-error	**RMS**-error	**MAX**-error	**RMS**-error
3	8.5246e4e-02	2.1517e-02	2.1047e-02	9.6999e-03	1.8302e-02	8.9470e-03
5	1.4259e-02	3.3936e-03	1.2666e-03	5.4619e-04	1.1480e-03	5.0893e-04
7	7.2383e-04	1.6244e-04	1.36676e-04	4.1719e-05	8.1498e-05	3.3618e-05
9	3.2763e-04	1.6094e-04	3.5024e-05	1.7944e-05	4.8233e-06	2.4234e-06
17	7.1340e-04	5.8457e-04	1.1443e-03	1.0710e-03	5.4030e-04	3.9382e-04

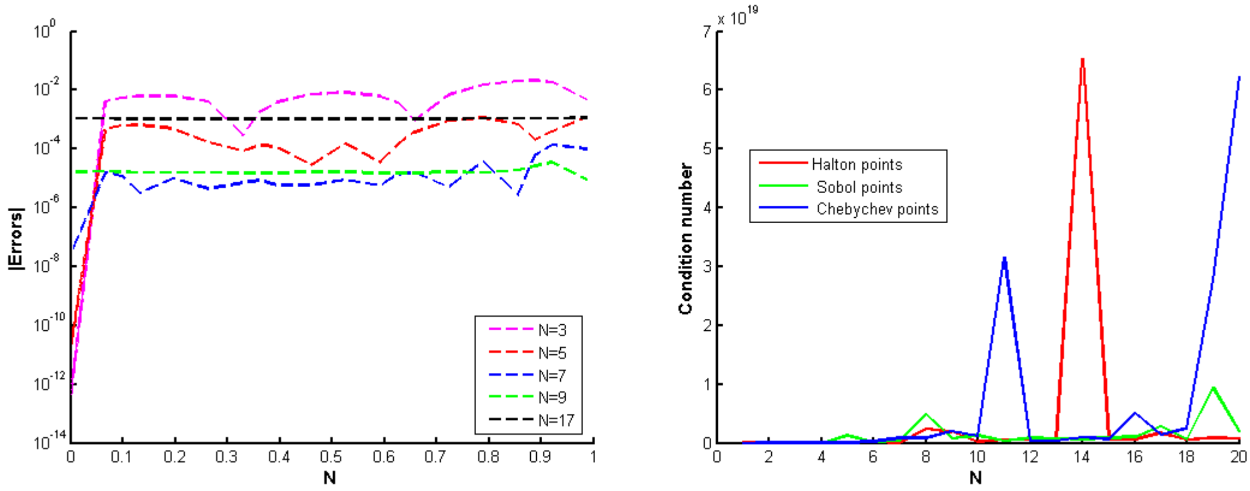

Figure 2.20: Absolute errors for non-stationary interpolation $\epsilon = 1.1818$ using Sobol points to the test function $f_1(x)$ with multiquadric RBF based on $M = 20$ uniformly spaced points in $[0, 1]$ and $N = 3, 5, 7, 9, 17$, (left), the condition number (right).

Table 2.48: 1D non-stationary interpolation ($\epsilon = 5.4545$) obtained using LOOCV strategy to the test function $f_2(x)$ with multiquadric RBF using Halton, Sobol and Chebyshev points.

	Halton points		Sobol points		Chebyshev points	
N	**MAX**-error	**RMS**-error	**MAX**-error	**RMS**-error	**MAX**-error	**RMS**-error
3	8.1496e-03	2.2664e-03	3.4965e-03	1.5535e-03	2.8222e-03	1.4316e-03
5	5.2554e-03	1.3007e-03	2.8605e-04	2.7069e-04	4.8607e-05	2.8909e-05
7	1.2072e-03	1.1692e-03	2.2625e-04	2.1388e-04	4.8279e-05	3.5853e-05
9	5.7929e-04	5.6974e-04	1.3540e-04	1.1491e-04	3.3140e-05	2.1139e-05
17	3.0636e-05	2.4027e-05	1.1582e-04	1.0469e-04	4.3451e-05	3.1373e-05

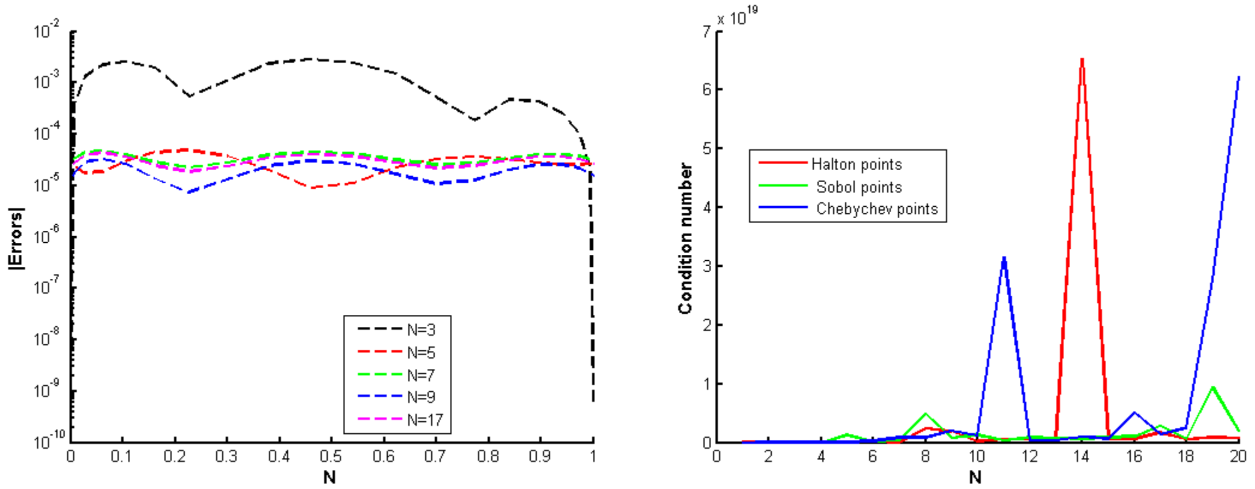

Figure 2.21: Absolute errors for non-stationary interpolation with $\epsilon = 5.4545$ using Chebyshev points the test function $f_2(x)$ with multiquadric based on power $M = 20$ uniformly spaced points in $[0, 1]$ and $N = 3, 5, 7, 9, 17$, (left), the condition number (right).

Analysis of numerical results

The subject in this work is to give a comparaison between the stationary and non-stationary interpolation approaches. From the previous results, we can conclude

- We see clearly the effect of the number of data points N on the condition number of interpolation matrix (condition number is growing with N increasing), for all types of nodes.

- For instance, in general we see that Chebyshev nodes gives good results in a comparaison with Sobol and Halton nodes in 1D and 2D approximations in both approaches.

- We can see that, from the tests of stationary and non-stationary appraoches that the better way is to find at least optimal strategies of N and the shape paramater on both approaches.

- In both, stationary and non-stationary approaches, the accuracy is better when ϵ is toward zero.

Chapter 3

Application of GRBFs and CSRBFs for solving two-dimensional Volterra-Fredholm integral equations

Radial basis functions interpolation is a very well-established numerical technique for reconstructing multivariate functions from the so called scattered data. Recent works give a general description for RBF interpolation-see [13, 84]. Radial basis functions method is a very attractive method when it is applied to high scattered data interpolation problems, compared to alternative methods such as finite elements and finite difference methods. The most popular globally and compactly supported radial basis functions are given in [84]. Polynomials and trigonometric functions can also be used as basis functions-see [55].

In [34] a comparative study indicates that the most elegant interpolation RBF is the so called Hardy's multiquadric. After that a huge amount of studies have been carried out to show the reliability of multivariate scattered data interpolation. Radial basis functions are also used for solving partial differential equations. In 1990, Edward Kansa first introduced the collocation method by radial basis functions-see [49]. Many radial basis functions (RBFs) contain a free shape parameter ϵ. The choice of this parameter has a big effect on the accuracy of RBF interpolation [34, 36]. The method of cross validation has long been used in the statistics literature, and the special case of leave-one-out cross validation (LOOCV) represents the basis of many strategies to find both the optimal number of iterations and the optimal shape parameter. The globally RBFs interpolation problem using a large number of data points can cause serious stability problems embodied in that the matrix is dense and can be highly ill conditioned, so it was suggested that the difficulty can be overcome by the use of compactly supported positive definite RBFs (CS-PD-RBF) and it was demonstrated by [85] that for a given dimension d and smoothness C^{2k}, a positive definite radial basis function in the form of a univariate polynomial of minimal degree always exists, and it is unique within a constant factor. Results were given for $d = 1$, $d \leq 3$ and $d \leq 5$, for the current two- dimensional problems. The use of compactly supported radial basis functions can reduce the resultant full matrix to a sparse one, also the operation of the banded matrix system could reduce the ill-conditioning of the resultant coefficient matrix when using the global radial

Nonlinear phenomena play an important role in applied mathematics, physics and engineering. Many phenomena in applied science are modeled by nonlinear equations. Solutions of nonlinear evolution equations provide better understanding of the physical mecanism of many phenomena [63, 95]. In general finding exact solutions is very difficult; so it is necessary to use numerical techniques for finding approximate solutions.

The aim of this chapter ([24]) is to solve the mixed two-dimensional Volterra-Fredholm integral equation written in the form

$$u(s,t) = g(s,t) + \int_0^s \int_{\Omega'} U(s,t,x,y,u(x,y))dydx, \tag{3.1}$$

where Ω' is a closed subset of $\mathbb{R}$, The function $g(s,t)$ and $U(s,t,x,y,u)$ are defined respectively on $D = [0,T] \times \Omega'$, and $S = \left\{(s,t,x,y,u) : 0 \le x \le s \le T, t, y \in \Omega'\right\}$, and $u(s,t)$ is an unknown scalar valued function defined on D. We assume that $U(s,t,x,y,u) = k(s,t,x,y)[u(s,t)]^p$, p is a positive integer.

The equation (3.1) has a unique solution $u \in C(D)$, $D = [0,T] \times \Omega'$ under the conditions given in [9, 64].

In our work, we take $\Omega' = [0,a] \subset \mathbb{R}$, it becomes clear that all results can be readily extended to more general regions ($\Omega' \subset \mathbb{R}^n$).

The analytical solutions of Volterra-Fredholm equation (3.1) either do not exist or are hard to find, hence some numerical methods have been proposed for the general equations, such that collocation methods, projection methods, Adomian decomposition method, meshless method using radial basis functions, ...etc. Two-dimensional integral equations have significant applications in various fields of sciences and engineering such as, heat conduction problem [5], diffusion problem [4], plasma physics [27], diffraction theory [66], axisymmetric contact for bodies with complex rheology [59], ...etc. Instead of dealing with a multivariate function ϕ whose complexity increases proportionally to the space dimension, in our work, we have the same univariate function for all choices of dimensions. Multiquadric radial basis function (MQ-RBF) is related to a consistent solution of the harmonic potential problem and it has a physical foundation. RBFs also are related to prewavelets. There are many important problems in biology and finance markets, physics, and chemistry that involve PDEs in higher dimension which can be also converted to higher integral equations and can be solved by RBF interpolation, for examples

- Dirac relativity.
- The six dimensional Boltzmann's equation.
- The Fokker-Plank equation that describes the evolution of Probability density function.
- The Benjamin-Bona-Malhony equation which has many physical applications, it has been utilized in different scientific areas, such that a constic gravity waves in compressible fluids, analysis of surfaces waves of long wavelength in liquids.

Radial basis functions network is a neural network approach, by viewing the design as a curve fitting problem in a high dimensional space. Learning is equivalent to finding a multidimensional function that provides a benefit to training data with the criterian for benefit being measured in some statistical scheme-see [26, 96]. There are distinct benefits from the idea of using generalized multiquadric and compactly supported radial basis functions, that can be used for irregular regions, high accuracy, wide applicability, sparse matrix for the use of compactly radial basis functions, and a general convergence also can be obtained. In this study, we employ generalized multiquadric with compactly supported RBF interpolation to generate a numerical solution technique for solving Volterra-Fredholm two-dimensional integral equation. To do this, we require a suitable integration based on Legendre-Gauss-Lobatto quadrature formula. The series expansion of the solution by radial basis functions is used to convert our problem to nonlinear system of equations. This technique is simple and its precision is investigated on some test examples.

3.1 Radial basis and compactly radial basis functions

The generalized multiquadric radial basis function (GMQ) $\Phi(r) = (r^2 + \epsilon^2)^\beta$ has the exponent β and shape parameter ϵ that controls the shape of the RBF. In the cases of $\beta < 0$, and $0 < \beta < 1$, the GMQ interpolation problem can be easily shown to be invertible. The GMQ is conditionally positive definite of order $\lceil \beta \rceil$, when $\beta > 1$. So it is necessary to append low order polynomials to the RBF interpolant , to assure that system matrix is invertible [29].

Half-integer exponents have been formed to work well for values of β in the GMQ, but we are going to investigate non-half-integer values as well. According to [83] , $\beta = 1.03$ was an optimal value for the GMQ, in 2003, [93] found that $\beta = 1.99$ is an optimal value. Also [52] suggested that the GMQ has better outcomes when $\beta = \frac{5}{2}$. In our numerical experiments the GMQ radial basis function will be performed for better understanding the best choices of the integer β.

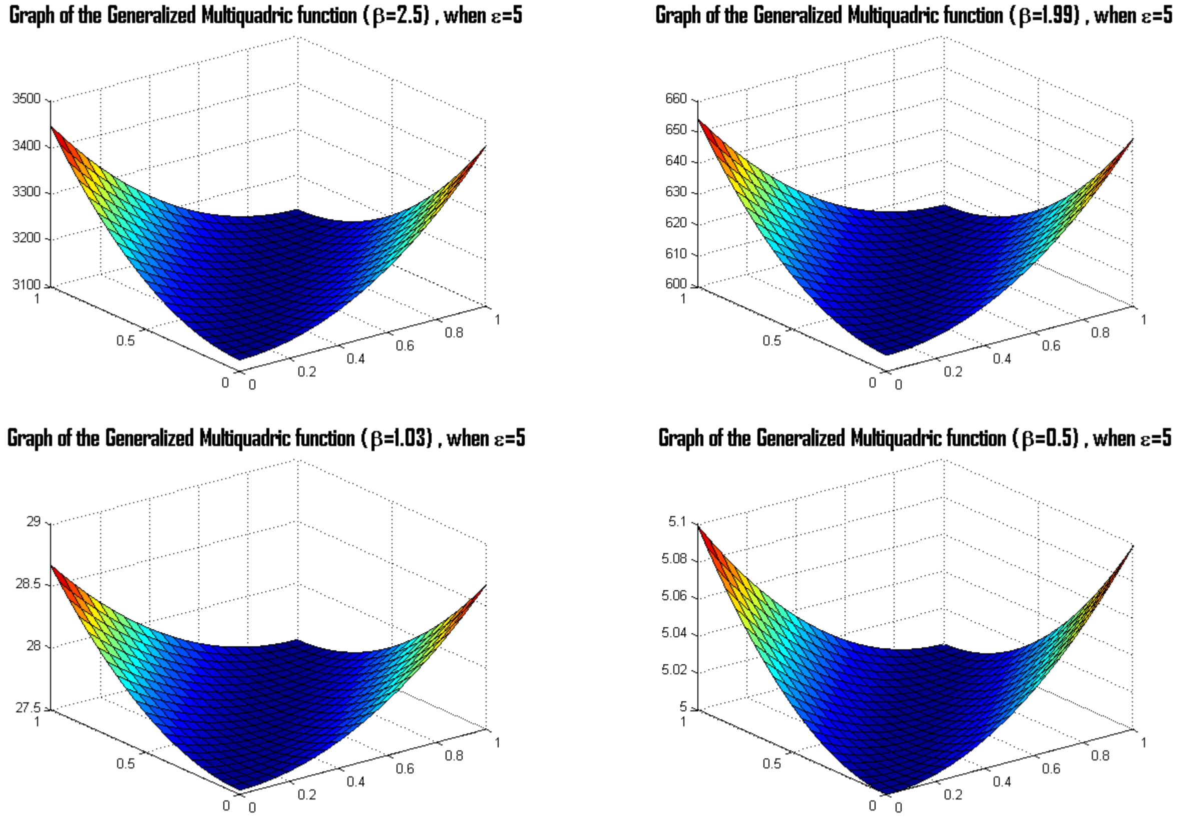

Figure 3.1: The GMQ function, with different values of β, when $\epsilon = 5$.

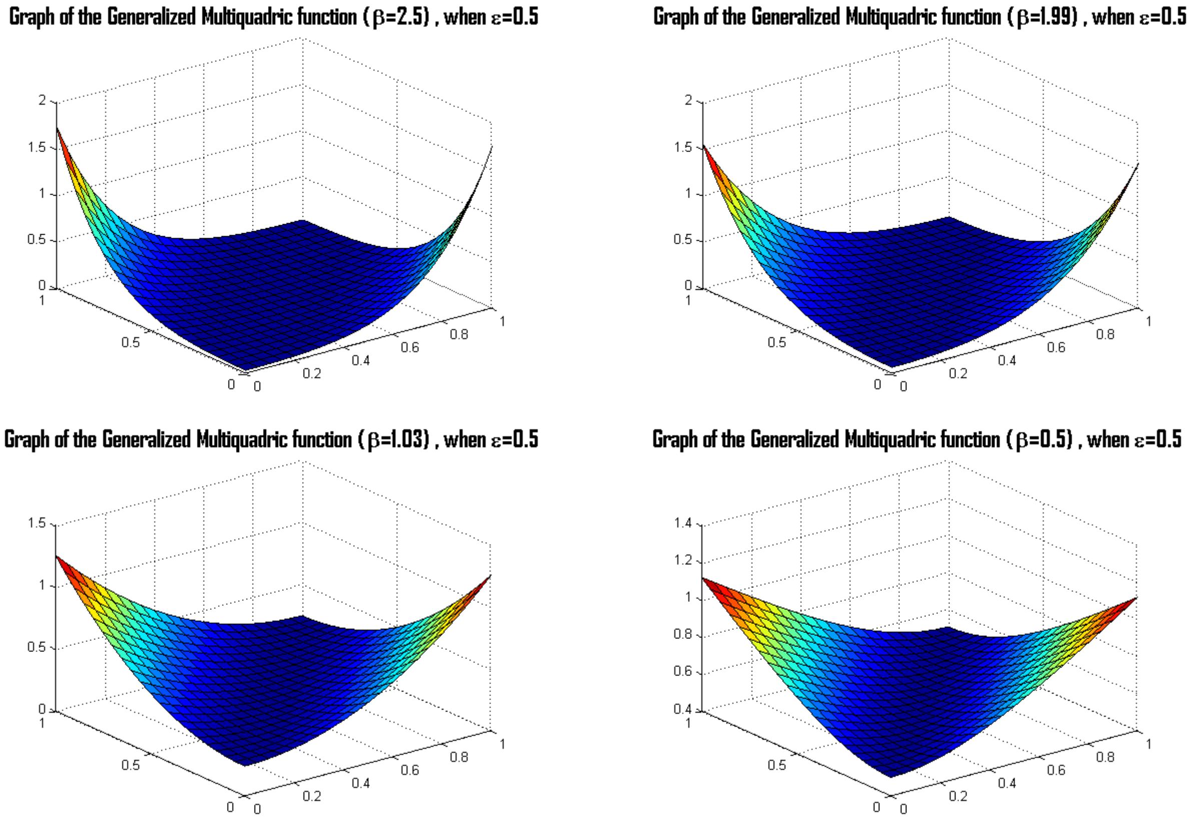

Figure 3.2: The GMQ function, with different values of β, when $\epsilon = 0.5$.

One of the most important RBFs are the compactly supported radial basis functions introduced in chapter 1, and the whole idea of their construction is based on the earlier work by Askey (1973) who observed by considering Fourier transforms that the truncated power function $\phi(r) = (1-r)_+^l$ gives rise to positive definite interpolation matrices for $l \geq [d/2]+1$.

Some of Wu's and Wendland's compactly supported radial basis functions are given in tables 3.1-3.2.

$\phi_{2,0}(r) = (1-r)_+^5(1+5r+9r^2+5r^3+r^4)$	$C^4 \cap SPD(\mathbb{R})$
$\phi_{2,1}(r) = D\phi_2(r) = (1-r)_+^4(4+16r+12r^2+3r^3)$	$C^2 \cap SPD(\mathbb{R}^3)$
$\phi_{2,2}(r) = D^2\phi_2(r) = (1-r)_+^3(8+9r+3r^2)$	$C^0 \cap SPD(\mathbb{R}^5)$

Table 3.1: Examples of Wu's CSRB functions.

$\phi_{3,0}(r) = (1-r)_+^3$	$C^0 \cap SPD(\mathbb{R}^5)$
$\phi_{3,1}(r) = (1-r)_+^4(1+4r)$	$C^2 \cap SPD(\mathbb{R}^3)$
$\phi_{3,2}(r) = (1-r)_+^6(3+18r+35r^2)$	$C^4 \cap SPD(\mathbb{R}^3)$

Table 3.2: Examples of Wendland's CSRB functions.

Furthermore, another class of CSRBFs constructed by Buhmann [12] is reminiscent of the popular thin plate splines. Three examples of these CSRBFs are given below

$\phi(x) = \Big(2r^4 log(r) + 7r^4/2 + 16r^3/3 + 2r^2 + 1/6\Big)_+$, $x \in R^3$;

$\phi(x) = \Big(112r^{9/2}/45 + 16r^{7/2}/3 + 7r^4 + 14r^2/15 + 1/9\Big)_+$, $x \in R^2$

$\phi(x) = \Big(1/18 - r^2 + 4r^3/9 + r^4/2 + 4r^3 log(r)/3\Big)_+, \, x \in R^2.$

For our study we use the Wendland's compactly supported radial basis function $\phi^{\sigma}_{4,2}$. Knowing that interpolation is always possible, it is necessary to look for the behaviour of the interpolation error $u - P_u$ (pointwise or in a given norm).

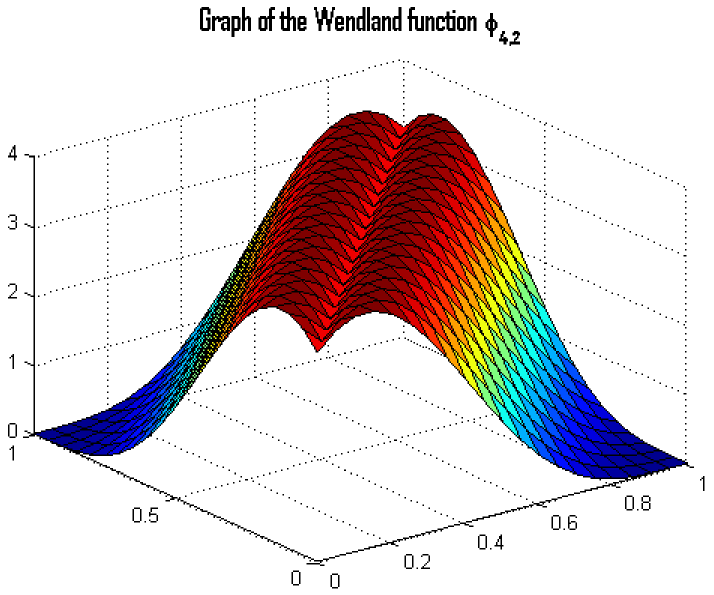

Figure 3.3: The Wendland compactly supported radial basis function $\phi_{4,2}$.

Theorem 3.1. *[85] Let $s = d/2 + k + 1/2$ and $k \geq 1$ for $d = 1, 2$. For every $u \in H^s(\mathbb{R}^d)$ and every compact $\Omega \subset \mathbb{R}^d$ satisfying a uniform interior cone condition, the interpolant $P_{u,X}$ on $X = \{x_1, x_2, \ldots, x_N\} \subset \Omega$ satisfying the estimate*

$$||u - P_{u,X}||_{L^\infty(\Omega)} \leq C||u||_{H^s(\mathbb{R}^d)} h^{k+1/2}, \tag{3.2}$$

with H is a sobolev space and $h = \sup_{x \in \Omega} \min_{1 \leq i \leq N} ||x - x_i||_2$ sufficiently small. Thus interpolation with $\Phi_{d,k}$ provides at least approximation of order $k + 1/2$.

3.2 Legendre–Gauss–Lobatto nodes and weights

The shifted Legendre–Gauss–Lobatto nodes are $x_0 = 0 < x_1 < \cdots < x_{N-1} < x_N = 1$, and x_m, $1 \leq m \leq N-1$ are the zeros of $L^{'}(x)$, where $L^{'}(x)$ is the derivative of Legendre polynomial of degree N, with respect to $x \in [0, 1]$. We approximate the integral of a function u on $[0, 1]$ as

$$\int_0^1 u(x)dx = \sum_{i=0}^{N} w_i u(x_i), \tag{3.3}$$

where x_i are the Legendre–Gauss–Lobatto nodes and the weights are w_i given in [16] as.

$$w_i = \frac{1}{N(N+1)} \cdot \frac{1}{(L_N(x_i)^2)}, \, i = 0, 1, \ldots, N. \tag{3.4}$$

The integration given in (3.3) is exact whenever $u(x)$ is a polynomial of degree $\leq 2N + 1$.

3.3 Description of the numerical technique

Let $\Phi(r)$ be the radial basis function, we approximate $u(s,t)$ with interpolation by the function $\Phi(r)$ as follow

$$u(s,t) \approx \sum_{i=0}^{N}\sum_{j=0}^{N} c_{ij}\Phi_{ij}(s,t) = C^T\Phi(s,t), \tag{3.5}$$

where $\Phi_{ij} = \Phi_{ij}(s,t) = \Phi(||(s,t)-(s_i,t_j)||)$,

$\Phi(s,t) = [\Phi_{00}, \Phi_{10}, \ldots, \Phi_{N0}, \Phi_{01}, \Phi_{11}, \ldots, \Phi_{N1}, \ldots, \Phi_{0N}, \Phi_{1N}, \ldots, \Phi_{NN}]^T$, and

$C^T = [c_{00}, c_{10}, \ldots, c_{N0}, c_{01}, c_{11}, \ldots, c_{N1}, \ldots, c_{0N}, c_{1N}, \ldots, c_{NN}]^T$.

Let given the two-dimensional Volterra-Fredholm integral equation (3.1) with $\Omega' = [0,1]$ and $s \in \Omega'$.

First, we collocate eq.(3.1) at shifted Legendre–Gauss–Lobatto nodes (s_i, t_j) for $i = 0, \ldots, N$, $j = 0, ..., N$. Hence, we have

$$u(s_i,t_j) = g(s_i,t_j) + \int_0^{s_i}\int_0^1 U(s_i,t_j,x,y,u(x,y))dydx. \tag{3.6}$$

Changing the variable $\tau = \frac{1}{s_i}x$, to transform the interval $[0,s_i]$ to $[0,1]$. Then, we have

$$u(s_i,t_j) = g(s_i,t_j) + s_i\int_0^1\int_0^1 U(s_i,t_j,\tau s_i,y,u(\tau s_i,y))dyd\tau. \tag{3.7}$$

By using strictly positive definite function $\Phi(r)$ or RBF approximation of $u(s,t)$ and applying numerical integration method given in (3.3), we can approximate the integral in equation (3.7) to obtain

$$C^T\Phi(s_i,t_j) = g(s_i,t_j) + s_i\sum_{r_1=0}^{N}\sum_{r_2=0}^{N}\omega_{r_1}\omega_{r_2}U(s_i,t_j,\tau_{r_1}s_i,y_{r_2},C^T\Phi(\tau_{r_1}s_i,y_{r_2})), \tag{3.8}$$

where w_{r_1} and w_{r_2}, r_1, $r_2 = 0, \ldots, N$ are given in equation (3.4). Then we obtain a nonlinear system of equations that can be solved via iteration methods such as Newton's iteration method to obtain the unknown vector C and then

$$\hat{u}(s,t) = \sum_{i=0}^{N}\sum_{j=0}^{N} c_{ij}\Phi_{ij}(s,t),$$

is the meshless discrete collocation solution based on the RBF approximation for the integral equation (3.1).

The following algorithm gives the numerical approximate solution of equation (3.1).
Algorithm of the proposed method

- Step 1. Input the given functions $g(s,t)$, $U(s,t,u(x,y))$.
- Step 2. Choose an RBF function $\Phi(r)$, we suggest here GMQ and CSRBF.
- Step 3. Choose the set of points (s_i,t_j), the shape parameter ϵ and the parameters β, σ.
- Step 4. Solve the $(N+1)\times(N+1)$ system of nonlinear equations (3.8) to determine the unknown vector C.
- Step 5. Substitute the obtained value of C in equation (3.5) to obtain the aprroximate solution $u(s,t)$.

3.4 Convergence analysis and error estimate

In the following, we investigate the error convergence analysis of the proposed method.

Theorem 3.2. *[84] (p. 139) Suppose $\Phi \in C(R^d) \cap L^1(\mathbb{R}^d)$ is a real-valued strictly positive definite function. Then the real native Hilbert space of Φ on $\mathbb{R}^d$ is given by*

$$\mathcal{N}_\Phi(\mathbb{R}^d) = \left\{ u \in C(R^d) \cap L^2(\mathbb{R}^d) : \frac{\hat{u}}{\sqrt{\hat{\Phi}}} \in L^2(\mathbb{R}^d) \right\}, \tag{3.9}$$

with inner product

$$\Big\langle u_1, u_2 \Big\rangle_{\mathcal{N}_\Phi(\mathbb{R}^d)} = \frac{1}{\sqrt{2\pi}} \Big\langle \frac{\hat{u_1}}{\sqrt{\hat{\Phi}}}, \frac{\hat{u_2}}{\sqrt{\hat{\Phi}}} \Big\rangle_{L^2(\mathbb{R}^d)} = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}^d} \frac{\hat{u_1}(y)\overline{\hat{u_2}(y)}}{\hat{\Phi}} dy, \tag{3.10}$$

where $\hat{u}$ denotes Fourier transform of u. Furthermore, every $u \in \mathcal{N}_\Phi(\mathbb{R}^d)$ has the representation

$$u(\chi) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}^d} \hat{u} e^{i\chi y} dy.$$

In particular, every $u \in \mathcal{N}_\Phi(\mathbb{R}^d)$ can be recoverd from its Fourier transform $\hat{u} \in L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$. The native spaces of translation invariant functions can be viewed as a generalization of standard Sobolev spaces. In other words, if the Fourier transform of strictly positive definite function Φ decays only algebraically, then the function Φ has a corresponding Sobolev space as its native space-see [29, 84].

Let $\Omega = [0,1] \times [0,1]$ and $C_N(\Omega) =$ span $\{\Phi(||x - x_1||), \ldots, \Phi(||x - x_N||)\}$. We define the collocation operator $\mathcal{P}_N$: $\mathcal{N}_\Phi(\Omega) \longrightarrow C_N(\Omega)$ as follows

$$\mathcal{P}_N u(x) = \sum_{i=1}^{N} c_i \phi(||x - x_i||, \, x \in \Omega. \tag{3.11}$$

Theorem 3.3. *[84](p.190) Let Ω be a cube in $\mathbb{R}^d$ Suppose that $\Phi = \phi(\| \ . \ \|_2)$ is a conditionally positive definite function such that $f := \phi(\sqrt{.})$ satisfies $| \ f^{(l)}(r) \ | \leq l! M^l$ for all integers $l \geq l_0$ and all $r \in [0, \infty)$, where $M > 0$ is a fixed constant. Then there exists a constant $c > 0$ such that the error between a function $f \in \mathcal{N}_\Phi(\Omega)$ and its interpolant sf, X can be bounded by*

$$||u - \mathcal{P}_N u||_{L^\infty(\Omega)} \leq e^{-C/h_{X,\Omega}} |u|_{\mathcal{N}_\Phi(\Omega)}, \tag{3.12}$$

provided $h_X \leq h_0(l)$.

for all data sites X with sufficiently small h_X. All radial basis and strictly positive definite functions give rise to reproducing kernels with respect to some Hilbert spaces which are named native Hilbert spaces.
Now, we represent the convergence analysis. Let given the two-dimensional integral equation (3.1), with $g(s,t) \in C([0,1] \times [0,1])$ and $U(s,t,u(x,y))$ is continuous for all $(s,t) \in [0,1] \times [0,1]$ and all x, y, satisfy the (uniform) Lipschitz conditions

$$|U(s,t,x,y,u(x,y) - U(s,t,x,y,u_1(x,y))| \leq L|u - u_1|. \tag{3.13}$$

We transform the equation (3.1) to operator form

$$Tu = u, \tag{3.14}$$

where the operator $T : C(\Omega) \longrightarrow C(\Omega)$, is defined as $Tu = g(s,t) + \int_0^s \int_{\Omega'} U(s,t,x,y,u(x,y))dydx$. Therefore, solving the integral equation (3.1) is equivalent to obtain the fixed points of operator T. Utilizing the Gauss–Legendre quadrature rule to approximate the integral in equation (3.1), a sequence of numerical integral operators $T_N, N \geq 1$ is introduced by

$$T_N u(s,t) = g(s,t) + s \sum_{r_1=0}^{N} \sum_{r_2=0}^{N} \omega_{r_1}\omega_{r_2} U(s,t,\tau_{r_1}s, y_{r_2}, u(\tau_{r_1}s, y_{r_2})),\ t,s \in [0,1]. \tag{3.15}$$

The discrete collocation method for solving equation (3.1) is defined by

$$z_N = \mathcal{P}_N T_N(z_N). \tag{3.16}$$

The iterated discrete collocation solution is defined by

$$\hat{z}_N = T_N(z_N). \tag{3.17}$$

Hence,

$$z_N = \mathcal{P}_N(\hat{z}_N), \tag{3.18}$$

and

$$\hat{z}_N = T_N \mathcal{P}_N(\hat{z}_N). \tag{3.19}$$

We suppose that

H1. T and $\{T_N, N \geq 1\}$, are completely continuous nonlinear operators on $C([0,1] \times [0,1])$.

H2. $\{T_N, N \geq 1\}$ is a collectively compact family on $C([0,1] \times [0,1])$.

H3. $\{T_N, N \geq 1\}$ is pointwise convergent to T on $C([0,1] \times [0,1])$, i.e., $T_N(u) \longrightarrow T(u)$, $u \in C([0,1] \times [0,1])$.

H4. At each point of $C([0,1] \times [0,1])$, $\{T_N\}$ is an equicontinuous family.

H5. T and $\{T_N, N \geq 1\}$ are twice Frechet differential on $B(u_0, r) = \{u : ||u - u_0|| \leq r, r > 0\}$ and moreover $||T''u||, ||T''_N u|| \leq M < \infty$, $N \geq 1$, $u \in B(u_0, r)$.

The previous hypotheses on T and $T_N, N \geq 1$ are listed and labeled in the following [1, 2, 3]. Theorem 4 in [3] affirms that under the previous conditions on $B(u_0, r)$, and let u_0 be a fixed point of the operator T and also assume that 1 is not an eigenvalue of $T'(u_0)$. If H5 is satisfied on $B(u_0, r) \subset C([0,1] \times [0,1])$. Then u_0 is an isolated fixed point of T, of nonzero index, Moreover, there are $0 < \epsilon \leq r$ and $N_1 > 0$ such that for every $N \geq N_1$, $T_N P_N$ has a unique fixed point $\hat{z}_N$ in $B(u_0, \epsilon)$. Also, there exist $\gamma > 0$ such that

$$||\hat{z}_N - u_0||_{L^\infty(\Omega)} \leq \gamma ||Tu_0 - T_N P_N u_0||_{L^\infty(\Omega)},\ N \geq N_1, \tag{3.20}$$

this gives a bound on the rate of convergence of $\hat{z}_N$ and u_0. The proof of the previous results is based on the fact that the operators $\{T_N P_N\}$ also satisfies the hypotheses H1-H4 and the remainder of the proof follows from Atkinson [1]. For the convergence of z_N to u, we use the

results given by [98] and also the works of [1, 2, 3] as follow
Let $e_N = u - z_N$ the error estimation of the proposed technique, we can write e_N in the form

$$e_N = u - \mathcal{P}_{\mathcal{N}}u + \mathcal{P}_{\mathcal{N}}u - z_N, \tag{3.21}$$

using equation (3.14), we have $\mathcal{P}_N Tu = \mathcal{P}_N u$. Applying equation (3.16), we obtain

$$\mathcal{P}_N u - z_N = \mathcal{P}_N Tu - \mathcal{P}_N T_N z_N = \mathcal{P}_N[Tu - T_N z_N] = \mathcal{P}_N\Bigg([Tu - T_N u] + [T_N u - T_N z_N]\Bigg). \tag{3.22}$$

We use mean-Value theorem to the kernel function $U(s,t,x,y,u(x,y))$ implies that there exists a function $\theta_N(s,t)$, whose value is between $u(s,t)$ and $z_N(s,t)$, such that

$$U(s,t,x,y,u(x,y)) - U(s,t,x,y,z_N) = U(s,t,x,y,\theta_N(x,y))e_N(x,y). \tag{3.23}$$

From equations (3.15) and (3.23), we have

$$\begin{aligned} T_N u - T_N z_N &= T_N(u - z_N) = s\sum_{r_1=0}^{N}\sum_{r_2=0}^{N}\omega_{r_1}\omega_{r_2}U[s,t,\tau_{r_1}s,y_{r_2},u(\tau_{r_1}s,y_{r_2})] \qquad (3.24)\\ &- s\sum_{r_1=0}^{N}\sum_{r_2=0}^{N}\omega_{r_1}\omega_{r_2}U[s,t,\tau_{r_1}s,y_{r_2},z_N(\tau_{r_1}s,y_{r_2}]\\ &= s\sum_{r_1=0}^{N}\sum_{r_2=0}^{N}\omega_{r_1}\omega_{r_2}U[s,t,\tau_{r_1}s,y_{r_2},\theta_N(\tau_{r_1}s,y_{r_2})]e_N(\tau_{r_1}s,y_{r_2}). \end{aligned}$$

Let

$$T_N u - T_N z_N = G_1(u - z_N), \tag{3.25}$$

where

$$(G_1\psi)(s,t) = s\sum_{r_1=0}^{N}\sum_{r_2=0}^{N}\omega_{r_1}\omega_{r_2}U[s,t,\tau_{r_1}s,y_{r_2},\theta_N(\tau_{r_1}s,y_{r_2})]\psi(\tau_{r_1}s,y_{r_2}). \tag{3.26}$$

Then, by using equation (3.22), we obtain

$$e_N = u - \mathcal{P}_{\mathcal{N}}u + \mathcal{P}_{\mathcal{N}}u - z_N = u - \mathcal{P}_N u + \mathcal{P}_N\Bigg([Tu - T_N u] + G_1([u - z_N])\Bigg), \tag{3.27}$$

hence,

$$[I - \mathcal{P}_N G_1](u - z_N) = u - \mathcal{P}_N u + \mathcal{P}_N([Tu - T_N u]), \tag{3.28}$$

this is equivalent to

$$[I - \mathcal{P}_N G_1]e_N = u - \mathcal{P}_N u + \mathcal{P}_N([Tu - T_N u]), \tag{3.29}$$

Now, we show the existence of the inverse $[I - \mathcal{P}_N G_1]$. Let

$$(G_2\psi)(s,t) = s\sum_{r_1=0}^{N}\sum_{r_2=0}^{N}\omega_{r_1}\omega_{r_2}U_u[s,t,\tau_{r_1}s,y_{r_2},u(\tau_{r_1}s,y_{r_2})]\psi(\tau_{r_1}s,y_{r_2}), \tag{3.30}$$

then, we have,

$$\begin{aligned} I - \mathcal{P}_N G_1 &= I - \mathcal{P}_N G_2 + \mathcal{P}_N(G_2 - G_1) \qquad (3.31)\\ &= (I - G_2) + (I - \mathcal{P}_N)G_2 + \mathcal{P}_N(G_2 - G_1)\\ &= (I - G_2)\Bigg[I + (I - G_2)^{-1}[(I - \mathcal{P}_N)G_2 + \mathcal{P}_N(G_2 - G_1)]\Bigg]. \end{aligned}$$

The existence and uniform boundedness of $(I-G_2)^{-1}$ and using that $||G_2-G_1|| \longrightarrow 0$ as $N \longrightarrow \infty$, $||(I-\mathcal{P}_N)G_2|| \longrightarrow 0$ as $N \longrightarrow \infty$ [43, 98]. We obtain that the operator $(I-\mathcal{P}_N G_1)^{-1}$ exists and is uniformly bounded for sufficiently large N. From the uniformly boundedness of the family $\{P_N\}$ we get

$$C_1 = \sup_{N\geq 1} ||\mathcal{P}_N||.$$

We have

$$||e_N|| \leq ||(I-\mathcal{P}_N G_1)^{-1}||\, ||u-\mathcal{P}_N u + \mathcal{P}_N([Tu-T_N u])||, \tag{3.32}$$

$$||e_N|| \leq C_2\left(C_l h_X^l |u|_{\mathcal{N}_\Phi(\Omega)}\right) + C_2 C_1 ||[Tu-T_N u]||, \tag{3.33}$$

$$||e_N|| \leq C_2 C_l h_X^l |u|_{\mathcal{N}_\Phi(\Omega)} + C_2 C_1 ||[Tu-T_N u]||, \tag{3.34}$$

where

$$||(I-\mathcal{P}_N G_1)^{-1}|| \leq C_2.$$

Since, $h_X^l \longrightarrow 0$ as $N \longrightarrow \infty$ (Justified by the quasi-uniform condition on X), and $\lim_{N\longrightarrow\infty} ||[Tu-T_N u]|| = 0$ (The error caused by numerical integration scheme) [3], yields to $z_N \longrightarrow u$. This completes the proof.

For the error estimation by using the compactly supported radial basis function, we just use the estimation in equation (3.2).

Remark 3.1. If z_N is a fixed point of equation (3.16), and consequently, a solution for the presented method, we consider the decomposition

$$u - z_N = u - P_N \hat{z}_N = \Big(u - P_N u\Big) + P_N\Big(u - \hat{z}_N\Big),$$

which gives

$$||u-z_N||_\infty \leq ||u-P_N u||_\infty + ||P_N||\, ||u-\hat{z}_N||_\infty,$$

by using (3.20), we get

$$\begin{aligned}
||u-z_N||_\infty &\leq ||u-P_N u||_\infty + \gamma ||P_N||\, ||Tu - T_N P_N u||_\infty \qquad (3.35)\\
&\leq ||u-P_N u||_\infty + \gamma ||P_N|| \Big[||Tu-T_N u||_\infty + ||T_N||\, ||u-P_N u||_\infty \Big]\\
&\leq ||u-P_N u||_\infty + \gamma ||P_N|| \Big[||Tu-T_N u||_\infty + \gamma_1 ||u-P_N u||_\infty \Big]\\
&\leq \Big(1+\gamma\gamma_1 ||P_N||\Big) ||u-P_N u||_\infty + \gamma ||P_N|| \Big[||Tu-T_N u||_\infty \Big]\\
&\leq \Big(1+\gamma\gamma_1 C_1\Big) ||u-P_N u||_\infty + \gamma C_1 \Big[||Tu-T_N u||_\infty \Big],
\end{aligned}$$

where $||T_N|| \leq \gamma_1$. Hence, the error for the presented method is mainly based on the error of the RBF interpolation and the error of the integration rule.

3.5 Analysis of numerical examples

In this section some comparative examples are provided to show the strength of the two proposed numerical methods in approximating the solution of two-dimensional Volterra-Fredholm integral equations. The aim here is to show that the choice of generalized multiquadric radial basis functions with the optimal exponent β gives a good accuracy for the approximation of the exact solution. From the test examples, we confirm that in different cases, the optimal values of β give a good accuracy compared with the results obtained by the compactly radial basis functions.

To analyse the efficiency and accuracy of the technique mentioned in the previous section, it was applied to some linear and nonlinear Volterra-Fredholm integral equations. The values of absolute error at some different points are given by the following formula

$$e(s,t) = |u(s,t) - u_N(s,t)|, \qquad 0 \leq s,t \leq 1.$$

The errors are represented in the following tables using different values of $N = 3$, 5 and 7, and some values of the parameters ϵ and σ. All the numerical computations have been done using MATLAB 2011.

Example 3.1. Consider the following linear Volterra–Fredholm integral equation [58]

$$U(s,t) = \sin(s) + t - \frac{e^s}{6} + 1/6 + \int_0^s \int_0^1 (2y-1)e^x U(x,y)dydx,\ 0 \leq s < 1, \tag{3.36}$$

in which $(s,t) \in ([0,1] \times [0,1])$, with the exact solution $U(s,t) = \sin(s) + t$. The numerical results using CSRBFs method are represented in table 3.3 and figure 3.4, for the GMQ, they are represented in tables 3.6 − 3.9 and figure 3.7.

Example 3.2. Let given the nonlinear Volterra-Fredholm integral equation

$$U(s,t) = \frac{1}{(1+s+t)^2} - \frac{s}{6(1+t)} + \int_0^1 \int_0^1 \frac{s}{1+t}(1+x+y)U^2(x,y)dydx, \tag{3.37}$$

with the exact solution $U(s,t) = \dfrac{1}{(1+s+t)^2}$. The numerical results using CSRBFs method are represented in table 3.4 and figure 3.5, for the GMQ, they are represented in tables 3.10 − 3.13 and figure 3.8.

Example 3.3. Consider the linear Volterra-Fredholm integral equation

$$U(s,t) = g(s,t) - \int_0^s \int_0^1 t^2 e^{-y} U(x,y)dydx, \tag{3.38}$$

where $g(s,t) = \frac{1}{3}s^2(3e^t + st^2)$. The exact solution is given by $U(s,t) = s^2e^t$. The numerical results using CSRBFs method are represented in table (3.5) and figure 3.6, for the GMQ, they are represented in tables 3.14 − 3.17 and figure 3.9.

Table 3.3: Computed errors using CSRBFs method for Example (3.1) , with $\sigma = 2$.

(s,t)	$N=3$	$N=5$	$N=7$	(s,t)	$N=3$	$N=5$	$N=7$
$(0,0)$	0.0000	0.0000	0.0000	$(0,0)$	0.0000	0.0000	0.0000
$(0.1,0.1)$	$4.0564E-2$	$3.5505E-2$	$3.4792E-1$	$(0.01,0.01)$	$5.6989E-3$	$4.1310E-3$	$3.5989E-3$
$(0.2,0.2)$	$6.6561E-1$	$7.2976E-2$	$7.3776E-1$	$(0.02,0.02)$	$1.1012E-3$	$8.0203E-3$	$7.0728E-3$
$(0.3,0.3)$	$1.2568E-1$	$1.1649E-1$	$1.1673E-1$	$(0.03,0.03)$	$1.5966E-2$	$1.1714E-2$	$1.0476E-2$
$(0.4,0.4)$	$2.1892E-1$	$1.6345E-1$	$1.6396E-1$	$(0.04,0.04)$	$2.0591E-2$	$1.5258E-2$	$1.3854E-2$
$(0.5,0.5)$	$3.0864E-1$	$2.1423E-1$	$2.1659E-1$	$(0.05,0.05)$	$2.4917E-2$	$1.8696E-2$	$1.7240E-2$
$(0.6,0.6)$	$3.5785E-1$	$2.7271E-1$	$2.7407E-1$	$(0.06,0.06)$	$2.8977E-2$	$2.2069E-2$	$2.0656E-2$
$(0.7,0.7)$	$3.5531E-1$	$3.4045E-1$	$3.3712E-1$	$(0.07,0.07)$	$3.2803E-2$	$3.5410E-2$	$2.4114E-2$
$(0.8,0.8)$	$3.3389E-1$	$4.1244E-1$	$4.0859E-1$	$(0.08,0.08)$	$3.6429E-2$	$2.8748E-2$	$2.7620E-2$
$(0.9,0.9)$	$3.7546E-1$	$4.8529E-1$	$4.8720E-1$	$(0.09,0.09)$	$3.9887E-2$	$3.2111E-2$	$3.1179E-2$

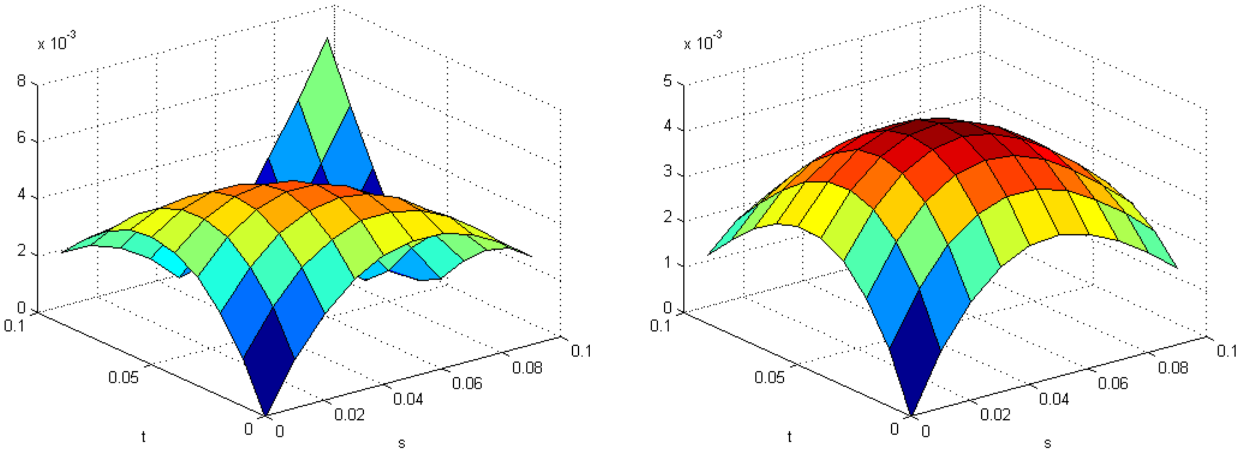

Figure 3.4: The Absolute errors, for $N = 3$ (left), $N = 5$ (right), when $\sigma = 1$.

Table 3.4: Computed errors using CSRBFs method for Example (3.2) , with $\sigma = 4$.

(s,t)	$N=3$	$N=5$	$N=7$	(s,t)	$N=3$	$N=5$	$N=7$
$(0,0)$	0.0000	$0.5395E-14$	$1.5530E-13$	$(0,0)$	0.0000	$0.5395E-14$	$1.5330E-13$
$(0.1,0.1)$	$2.9743E-2$	$7.6968E-4$	$2.2225E-4$	$(0.01,0.01)$	$9.2435E-3$	$2.5670E-3$	$7.9981E-4$
$(0.2,0.2)$	$8.9163E-3$	$4.7855E-4$	$4.4320E-6$	$(0.02,0.02)$	$1.6567E-2$	$4.0243E-3$	$9.9977E-4$
$(0.3,0.3)$	$1.0070E-3$	$2.2689E-5$	$6.6375E-5$	$(0.03,0.03)$	$2.2192E-2$	$4.6332E-3$	$8.6230E-4$
$(0.4,0.4)$	$1.5321E-3$	$5.4130E-5$	$2.2830E-6$	$(0.04,0.04)$	$2.6327E-2$	$4.6291E-3$	$5.8772E-4$
$(0.5,0.5)$	$9.0126E-4$	$7.65860E-6$	$9.3241E-6$	$(0.05,0.05)$	$2.9164E-2$	$4.2168E-3$	$3.0433E-4$
$(0.6,0.6)$	$7.1433E-4$	$2.5748E-5$	$6.6099E-7$	$(0.06,0.06)$	$3.0885E-2$	$3.5672E-3$	$7.4510E-5$
$(0.7,0.7)$	$1.1294E-4$	$3.6335E-4$	$7.0214E-6$	$(0.07,0.07)$	$3.1655E-2$	$2.8150E-3$	$8.5540E-5$
$(0.8,0.8)$	$8.2513E-4$	$4.3246E-5$	$2.3258E-7$	$(0.08,0.08)$	$3.1629E-2$	$2.0590E-3$	$1.8060E-4$
$(0.9,0.9)$	$1.1044E-3$	$8.9130E-6$	$1.3192E-6$	$(0.09,0.09)$	$3.0949E-2$	$1.3650E-3$	$2.2177E-4$

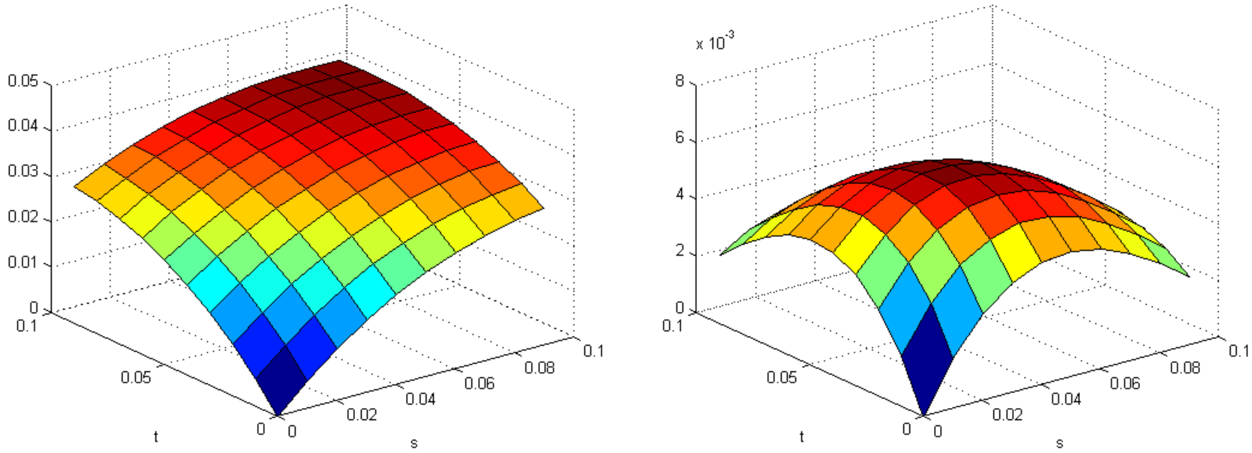

Figure 3.5: The Absolute errors, for $N = 3$ (left), $N = 5$ (right), when $\sigma = 3$.

Table 3.5: Computed errors using CSRBFs method for Example (3.3) , with $\sigma = 6$.

(s,t)	$N=3$	$N=5$	$N=7$	(s,t)	$N=3$	$N=5$	$N=7$
$(0,0)$	0.0000	$2.8710E-10$	$3.6988E-13$	$(0,0)$	0.0000	$2.8710E-10$	$3.9688E-13$
$(0.1,0.1)$	$6.5254E-3$	$4.7116E-3$	$4.5910E-3$	$(0.01,0.01)$	$4.1075E-4$	$1.4104E-4$	$7.1890E-5$
$(0.2,0.2)$	$1.8858E-2$	$1.7066E-2$	$1.7441E-2$	$(0.02,0.02)$	$8.7134E-4$	$3.5032E-4$	$2.2648E-4$
$(0.3,0.3)$	$3.6020E-2$	$3.6260E-2$	$3.6725E-2$	$(0.03,0.03)$	$1.3833E-3$	$6.3022E-4$	$4.6622E-4$
$(0.4,0.4)$	$5.6531E-2$	$6.0044E-2$	$5.9689E-2$	$(0.04,0.04)$	$1.9482E-3$	$9.8294E-4$	$7.9293E-4$
$(0.5,0.5)$	$7.9698E-2$	$8.4014E-2$	$8.3034E-2$	$(0.05,0.05)$	$2.5676E-3$	$1.4104E-3$	$1.2076E-3$
$(0.6,0.6)$	$1.0154E-1$	$1.0249E-1$	$1.0213E-2$	$(0.06,0.06)$	$3.2429E-3$	$1.9114E-3$	$1.7104E-3$
$(0.7,0.7)$	$1.1059E-1$	$1.0944E-1$	$1.0985E-1$	$(0.07,0.07)$	$3.9753E-3$	$2.4959E-3$	$2.3010E-3$
$(0.8,0.8)$	$8.8154E-2$	$9.7181E-2$	$9.8780E-2$	$(0.08,0.08)$	$4.7659E-3$	$3.1558E-3$	$2.9786E-3$
$(0.9,0.9)$	$2.2374E-2$	$5.0871E-2$	$8.1713E-2$	$(0.09,0.09)$	$5.6157E-3$	$3.8944E-3$	$3.7423E-3$

Table 3.6: Computed errors using GMQ method for $\epsilon = 2$, with $\beta = \frac{5}{2}$. for Example (3.1).

(t,s)	$N=3$	$N=5$	$N=7$	(t,s)	$N=3$	$N=5$	$N=7$
$(0,0)$	$3.9790E-13$	$4.3656E-11$	$1.3970E-8$	$(0,0)$	$3.9790E-13$	$4.3656E-11$	$1.3970E-8$
$(0.1,0.1)$	$7.4020E-4$	$1.7251E-7$	$8.7544E-8$	$(0.01,0.01)$	$1.8378E-4$	$4.3282E-7$	$1.7229E-8$
$(0.2,0.2)$	$3.4535E-4$	$3.9923E-8$	$4.7497E-8$	$(0.02,0.02)$	$3.3563E-4$	$6.8681E-7$	$4.3259E-8$
$(0.3,0.3)$	$8.1556E-5$	$2.8016E-7$	$6.5193E-9$	$(0.03,0.03)$	$4.5872E-4$	$8.0477E-7$	$6.1002E-8$
$(0.4,0.4)$	$2.7082E-4$	$4.0188E-7$	$2.5611E-9$	$(0.04,0.04)$	$5.5601E-4$	$8.2375E-7$	$4.1910E-8$
$(0.5,0.5)$	$2.8697E-4$	$1.2181E-6$	$5.3318E-8$	$(0.05,0.05)$	$6.3028E-4$	$7.7295E-7$	$4.3772E-8$
$(0.6,0.6)$	$2.1627E-4$	$6.7616E-7$	$2.0955E-9$	$(0.06,0.06)$	$6.8409E-4$	$6.7743E-7$	$3.0734E-8$
$(0.7,0.7)$	$5.4858E-5$	$9.8666E-7$	$1.8626E-9$	$(0.07,0.07)$	$7.1985E-4$	$5.5623E-7$	$1.9558E-8$
$(0.8,0.8)$	$2.2223E-4$	$1.4628E-6$	$4.5635E-9$	$(0.08,0.08)$	$7.3978E-4$	$4.2424E-7$	$8.3819E-9$
$(0.9,0.9)$	$4.5426E-4$	$3.8013E-7$	$3.8650E-9$	$(0.09,0.09)$	$7.4593E-4$	$2.9369E-7$	$2.1420E-8$

Table 3.7: Computed errors using GMQ method for $\epsilon = 2$, with $\beta = 1.99$. for Example (3.1).

(t, s)	$N = 3$	$N = 5$	$N = 7$	(t, s)	$N = 3$	$N = 5$	$N = 7$
$(0, 0)$	$9.0950E - 13$	$4.6566E - 10$	$1.9511E - 7$	$(0, 0)$	$9.0950E - 13$	$4.6566E - 10$	$1.9511E - 7$
$(0.1, 0.1)$	$1.7454E - 4$	$5.3691E - 7$	$1.2573E - 8$	$(0.01, 0.01)$	$3.1572E - 5$	$5.4308E - 7$	$6.1467E - 8$
$(0.2, 0.2)$	$1.1876E - 4$	$2.2643E - 6$	$3.5390E - 8$	$(0.02, 0.02)$	$5.9656E - 5$	$9.4087E - 7$	$4.6100E - 8$
$(0.3, 0.3)$	$4.0928E - 5$	$1.7799E - 6$	$2.1886E - 8$	$(0.03, 0.03)$	$8.4382E - 5$	$1.2056E - 6$	$6.2864E - 8$
$(0.4, 0.4)$	$1.9590E - 4$	$1.5063E - 6$	$1.2573E - 8$	$(0.04, 0.04)$	$1.0588E - 4$	$1.3527E - 6$	$2.9337E - 8$
$(0.5, 0.5)$	$2.6710E - 4$	$3.5699E - 6$	$1.0664E - 7$	$(0.05, 0.05)$	$1.2429E - 4$	$1.3961E - 6$	$4.1910E - 9$
$(0.6, 0.6)$	$2.1496E - 4$	$1.7164E - 6$	$3.4946E - 8$	$(0.06, 0.06)$	$1.3973E - 4$	$1.3505E - 6$	$6.0070E - 8$
$(0.7, 0.7)$	$4.9315E - 5$	$2.3142E - 6$	$2.7940E - 8$	$(0.07, 0.07)$	$1.5235E - 4$	$1.2295E - 6$	$4.9360E - 8$
$(0.8, 0.8)$	$1.5833E - 4$	$3.3884E - 6$	$2.7940E - 9$	$(0.08, 0.08)$	$1.6226E - 4$	$1.0449E - 6$	$4.8894E - 8$
$(0.9, 0.9)$	$2.5796E - 4$	$9.3837E - 7$	$9.8720E - 8$	$(0.09, 0.09)$	$1.6962E - 4$	$8.1037E - 6$	$3.7719E - 8$

Table 3.8: Computed errors using GMQ method for $\epsilon = 4$, with $\beta = 1.03$. for Example (3.1).

(t, s)	$N = 3$	$N = 5$	$N = 7$	(t, s)	$N = 3$	$N = 5$	$N = 7$
$(0, 0)$	$1.1642E - 10$	$4.7684E - 6$	$1.1921E - 6$	$(0, 0)$	$1.1642E - 10$	$4.7684E - 6$	$1.1921E - 6$
$(0.1, 0.1)$	$1.8864E - 4$	$5.7222E - 6$	$7.1526E - 7$	$(0.01, 0.01)$	$3.5496E - 5$	$3.8147E - 6$	$9.5367E - 7$
$(0.2, 0.2)$	$1.2361E - 4$	$4.7684E - 6$	$1.4305E - 6$	$(0.02, 0.02)$	$6.6787E - 5$	$1.9073E - 6$	$1.6689E - 6$
$(0.3, 0.3)$	$4.1201E - 5$	$3.8147E - 6$	$2.3842E - 6$	$(0.03, 0.03)$	$9.3993E - 5$	$2.8610E - 6$	$2.3842E - 6$
$(0.4, 0.4)$	$1.9141E - 4$	$8.5831E - 6$	$2.1458E - 6$	$(0.04, 0.04)$	$1.1741E - 4$	$8.5831E - 6$	0.0000
$(0.5, 0.5)$	$2.5335E - 4$	$6.6757E - 6$	$2.8610E - 6$	$(0.05, 0.05)$	$1.3721E - 4$	$1.9073E - 6$	$2.8610E - 6$
$(0.6, 0.6)$	$1.9764E - 4$	$3.8147E - 6$	$9.5367E - 7$	$(0.06, 0.06)$	$1.5358E - 4$	$1.2398E - 5$	$2.3842E - 7$
$(0.7, 0.7)$	$4.3824E - 5$	$5.7220E - 6$	$1.1921E - 6$	$(0.07, 0.07)$	$1.6673E - 4$	$3.8147E - 6$	$1.1921E - 6$
$(0.8, 0.8)$	$1.3531E - 6$	$2.8610E - 6$	$1.1921E - 6$	$(0.08, 0.08)$	$1.7683E - 4$	$1.4444E - 5$	$1.6689E - 6$
$(0.9, 0.9)$	$2.1123E - 4$	$1.9073E - 6$	0.0000	$(0.09, 0.09)$	$1.8407E - 4$	$3.8147E - 6$	$7.1526E - 6$

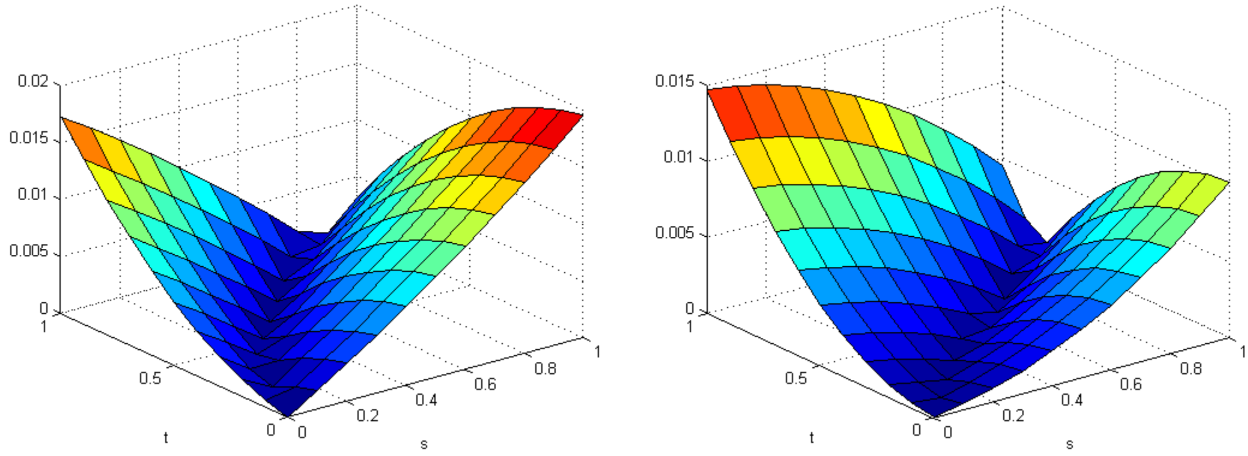

Figure 3.6: The Absolute errors, for $N = 3$ (left), $N = 5$ (right), when $\sigma = 7.5$.

Table 3.9: Computed errors using GMQ method for $\epsilon = 3$, with $\beta = \frac{1}{2}$. for Example (3.1).

(t,s)	$N=3$	$N=5$	$N=7$	(t,s)	$N=3$	$N=5$	$N=7$
$(0,0)$	$4.5475E-13$	$4.6566E-10$	$2.8902E-8$	$(0,0)$	$4.5475E-13$	$4.6566E-10$	$2.9802E-8$
$(0.1,0.1)$	$1.3838E-5$	$3.2270E-7$	$1.3039E-8$	$(0.01,0.01)$	$8.7356E-7$	$3.3644E-7$	$5.5979E-9$
$(0.2,0.2)$	$1.6107E-5$	$1.3169E-6$	$1.5832E-8$	$(0.02,0.02)$	$1.9968E-6$	$5.7952E-7$	$8.3819E-9$
$(0.3,0.3)$	$6.2842E-6$	$1.0228E-6$	$4.7497E-8$	$(0.03,0.03)$	$3.3129E-6$	$7.3924E-7$	$9.3132E-9$
$(0.4,0.4)$	$4.1039E-5$	$8.5006E-7$	$2.7940E-9$	$(0.04,0.04)$	$4.7682E-6$	$8.2608E-7$	$3.4459E-9$
$(0.5,0.5)$	$6.4107E-5$	$2.0091E-6$	$5.8673E-8$	$(0.05,0.05)$	$6.3125E-6$	$8.4867E-7$	$2.4214E-8$
$(0.6,0.6)$	$5.5851E-5$	$9.6601E-7$	$1.0245E-8$	$(0.06,0.06)$	$7.8986E-6$	$8.1770E-7$	$1.8626E-8$
$(0.7,0.7)$	$1.4412E-5$	$1.2862E-6$	$1.8626E-9$	$(0.07,0.07)$	$9.4829E-6$	$7.4226E-6$	$8.3819E-6$
$(0.8,0.8)$	$3.8213E-5$	$1.8808E-6$	$7.3574E-8$	$(0.08,0.08)$	$1.1024E-5$	$6.2818E-7$	$5.5879E-9$
$(0.9,0.9)$	$5.9713E-5$	$5.1130E-7$	$1.2107E-8$	$(0.09,0.09)$	$1.2488E-5$	$4.8708E-7$	$5.5879E-9$

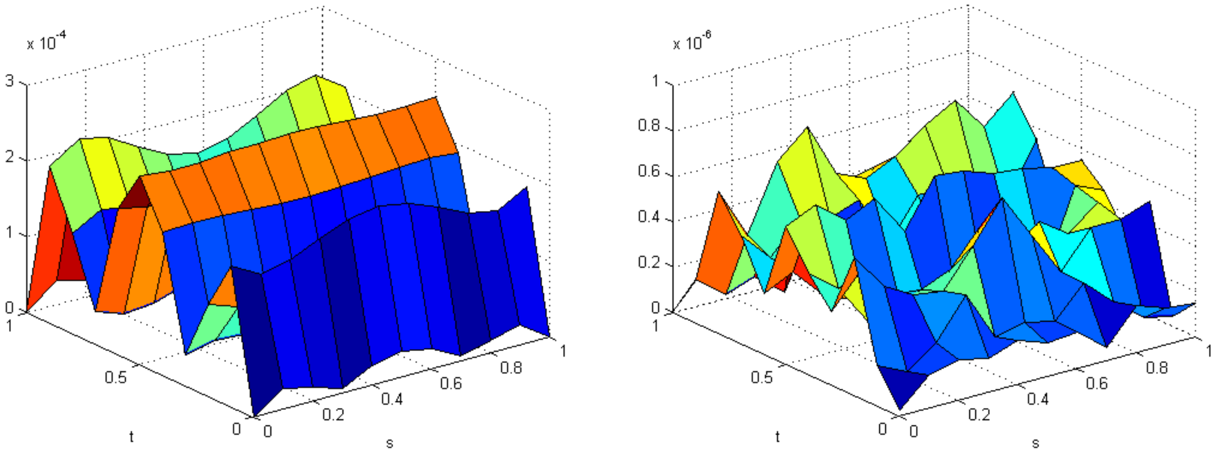

Figure 3.7: The Absolute errors, for $N = 3$ (left), $N = 5$ (right) with $\beta = 1.99$, when $\epsilon = 3.5$.

Table 3.10: Computed errors using GMQ method for $\epsilon = 2$, with $\beta = \frac{5}{2}$. for Example (3.2).

(t,s)	$N=3$	$N=5$	$N=7$	(t,s)	$N=3$	$N=5$	$N=7$
$(0,0)$	$0.7276E-11$	$4.4703E-8$	$4.8828E-4$	$(0,0)$	$0.7276E-11$	$4.4703E-8$	$4.8828E-4$
$(0.1,0.1)$	$4.7513E-3$	$3.4547E-4$	$1.0376E-3$	$(0.01,0.01)$	$1.5134E-5$	$3.7628E-4$	$1.3428E-3$
$(0.2,0.2)$	$4.5234E-3$	$1.1547E-3$	$1.2207E-4$	$(0.02,0.02)$	$2.2982E-4$	$6.5345E-4$	$1.0986E-3$
$(0.3,0.3)$	$1.6590E-3$	$6.7922E-4$	$6.1035E-4$	$(0.03,0.03)$	$6.5942E-4$	$8.3712E-4$	$1.2207E-4$
$(0.4,0.4)$	$6.9041E-3$	$4.2710E-4$	$6.1035E-4$	$(0.04,0.04)$	$1.2107E-3$	$9.3625E-4$	$1.4038E-3$
$(0.5,0.5)$	$7.7879E-3$	$7.6804E-4$	$1.4038E-3$	$(0.05,0.05)$	$1.8316E-3$	$9.5965E-4$	$2.9907E-3$
$(0.6,0.6)$	$4.9804E-3$	$2.8787E-4$	$3.0518E-4$	$(0.06,0.06)$	$2.4795E-3$	$9.1955E-4$	$2.4414E-3$
$(0.7,0.7)$	$9.6601E-3$	$3.0913E-4$	$2.1362E-3$	$(0.07,0.07)$	$3.1198E-3$	$8.2801E-4$	$1.4038E-3$
$(0.8,0.8)$	$1.9807E-3$	$3.6240E-4$	$1.7700E-3$	$(0.08,0.08)$	$3.7252E-3$	$6.9363E-4$	$1.8311E-4$
$(0.9,0.9)$	$2.2527E-3$	$8.1174E-5$	$1.6479E-3$	$(0.09,0.09)$	$4.2743E-3$	$5.3024E-4$	$8.5449E-4$

Table 3.11: Computed errors using GMQ method for $\epsilon = 2$, with $\beta = 1.99$. for Example (3.2).

(t,s)	$N=3$	$N=5$	$N=7$	(t,s)	$N=3$	$N=5$	$N=7$
$(0,0)$	$2.3283E-10$	0.0000	$1.3733E-4$	$(0,0)$	$2.3283E-11$	0.0000	$1.3733E-4$
$(0.1,0.1)$	$3.5544E-3$	$3.1471E-4$	$1.0681E-4$	$(0.01,0.01)$	$4.3764E-4$	$3.0613E-4$	$3.2043E-4$
$(0.2,0.2)$	$4.2906E-3$	$1.1196E-3$	$1.2970E-4$	$(0.02,0.02)$	$5.1854E-4$	$5.4407E-4$	$2.2888E-4$
$(0.3,0.3)$	$1.7012E-3$	$6.6805E-4$	0.0000	$(0.03,0.03)$	$3.3092E-4$	$7.1859E-4$	$1.2207E-4$
$(0.4,0.4)$	$7.4346E-3$	$4.3082E-4$	$2.2888E-5$	$(0.04,0.04)$	$5.0562E-5$	$8.0490E-4$	$3.0518E-5$
$(0.5,0.5)$	$8.7508E-3$	$7.9393E-4$	$6.1035E-5$	$(0.05,0.05)$	$5.6313E-4$	$8.4400E-4$	$4.5776E-5$
$(0.6,0.6)$	$5.8734E-3$	$3.0470E-4$	$2.4414E-4$	$(0.06,0.06)$	$1.1544E-3$	$8.2827E-4$	$9.1553E-5$
$(0.7,0.7)$	$1.1955E-3$	$3.2568E-4$	$7.6294E-6$	$(0.07,0.07)$	$1.7811E-3$	$7.3528E-4$	$1.5259E-4$
$(0.8,0.8)$	$2.8092E-3$	$3.9115E-4$	$3.5858E-4$	$(0.08,0.08)$	$2.4079E-3$	$6.3801E-4$	$2.4414E-4$
$(0.9,0.9)$	$4.0571E-3$	$8.5831E-5$	$3.8147E-5$	$(0.09,0.09)$	$3.0065E-3$	$4.8161E-4$	$1.8311E-4$

Table 3.12: Computed errors using GMQ method for $\epsilon = 1.8$, with $\beta = 1.03$. for Example (3.2).

(t,s)	$N=3$	$N=5$	$N=7$	(t,s)	$N=3$	$N=5$	$N=7$
$(0,0)$	$1.4552E-11$	$2.3842E-7$	$3.9673E-4$	$(0,0)$	$1.4552E-11$	$2.3842E-7$	$3.9673E-4$
$(0.1,0.1)$	$4.3032E-3$	$1.9247E-4$	$8.882E-4$	$(0.01,0.01)$	$2.4396E-3$	$7.4796E-5$	$8.1253E-4$
$(0.2,0.2)$	$7.1513E-3$	$8.2710E-4$	$4.9621E-4$	$(0.02,0.02)$	$4.1650E-3$	$1.6911E-4$	$2.6703E-5$
$(0.3,0.3)$	$8.7069E-4$	$5.2533E-4$	$2.3270E-4$	$(0.03,0.03)$	$5.3012E-3$	$2.6092E-4$	$1.9073E-5$
$(0.4,0.4)$	$4.5922E-3$	$3.3735E-4$	$3.4332E-4$	$(0.04,0.04)$	$5.9570E-3$	$3.3626E-4$	$1.0681E-4$
$(0.5,0.5)$	$5.7003E-3$	$6.1741E-4$	$2.1744E-4$	$(0.05,0.05)$	$6.2274E-3$	$3.8509E-4$	$3.9673E-4$
$(0.6,0.6)$	$3.7566E-3$	$2.3790E-4$	$1.7166E-4$	$(0.06,0.06)$	$6.1945E-3$	$4.0373E-4$	$2.0218E-4$
$(0.7,0.7)$	$7.5440E-4$	$2.5992E-4$	$1.4687E-4$	$(0.07,0.07)$	$5.9292E-3$	$3.9151E-4$	$3.1281E-4$
$(0.8,0.8)$	$1.5053E-3$	$3.0144E-4$	$1.4496E-4$	$(0.08,0.08)$	$5.4923E-3$	$3.4979E-4$	$4.8065E-4$
$(0.9,0.9)$	$2.0854E-3$	$6.4575E-4$	$6.1035E-5$	$(0.09,0.09)$	$4.9356E-3$	$2.8203E-4$	$6.4468E-4$

Table 3.13: Computed errors using GMQ method for $\epsilon = 0.6$, with $\beta = \frac{1}{2}$. for Example (3.2).

(t,s)	$N=3$	$N=5$	$N=7$	(t,s)	$N=3$	$N=5$	$N=7$
$(0,0)$	$0.0001E-11$	$1.0170E-13$	$2.0464E-12$	$(0,0)$	$0.0001E-11$	$1.0170E-13$	$2.0464E-12$
$(0.1,0.1)$	$3.2044E-2$	$7.1103E-4$	$6.9573E-5$	$(0.01,0.01)$	$9.9507E-3$	$2.1458E-3$	$3.3052E-4$
$(0.2,0.2)$	$8.6903E-3$	$3.3313E-4$	$1.3704E-5$	$(0.02,0.02)$	$1.7851E-2$	$3.3873E-3$	$4.2113E-4$
$(0.3,0.3)$	$2.2576E-4$	$7.1978E-4$	$2.2836E-4$	$(0.03,0.03)$	$2.3932E-2$	$3.9365E-3$	$3.7471E-4$
$(0.4,0.4)$	$3.8850E-3$	$4.5121E-4$	$2.4781E-5$	$(0.04,0.04)$	$2.8411E-2$	$3.9876E-3$	$2.6513E-4$
$(0.5,0.5)$	$6.3386E-3$	$9.8913E-4$	$3.4746E-4$	$(0.05,0.05)$	$3.1491E-2$	$3.6885E-3$	$1.4197E-4$
$(0.6,0.6)$	$4.4749E-3$	$4.8454E-4$	$1.7459E-5$	$(0.06,0.06)$	$3.3359E-2$	$3.1725E-3$	$3.5330E-5$
$(0.7,0.7)$	$9.8344E-4$	$5.5387E-4$	$1.7688E-4$	$(0.07,0.07)$	$3.4192E-2$	$2.5447E-3$	$3.9877E-5$
$(0.8,0.8)$	$2.5198E-3$	$6.3404E-4$	$9.2614E-6$	$(0.08,0.08)$	$3.4152E-2$	$1.8873E-3$	$7.9523E-5$
$(0.9,0.9)$	$3.9825E-3$	$1.5049E-4$	$6.1714E-5$	$(0.09,0.09)$	$3.3390E-2$	$1.2619E-3$	$8.6915E-5$

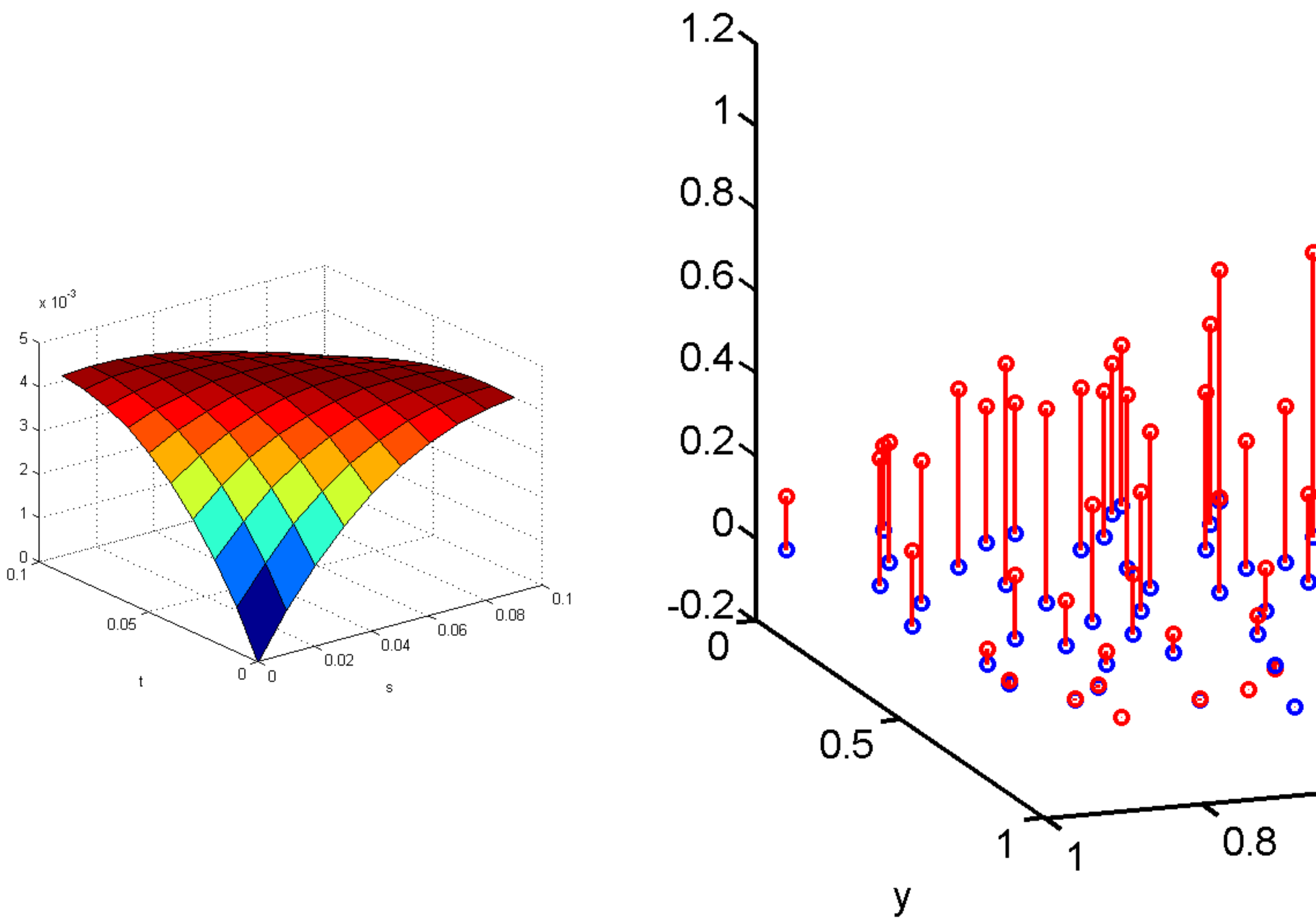

Figure 3.8: The Absolute errors for $N = 3$ (left), $N = 5$ (right) with $\beta = 1.03$, when $\epsilon = 2.5$.

Table 3.14: Computed errors using GMQ method for $\epsilon = 1.8$, with $\beta = \frac{5}{2}$. for Example (3.3).

(t,s)	$N=3$	$N=5$	$N=7$	(t,s)	$N=3$	$N=5$	$N=7$
$(0,0)$	$0.0142E-11$	$2.0373E-10$	$5.2452E-5$	$(0,0)$	$0.0142E-11$	$2.0373E-10$	$5.2452E-5$
$(0.1,0.1)$	$6.5427E-3$	$2.8740E-5$	$1.0490E-5$	$(0.01,0.01)$	$1.2178E-3$	$3.1728E-5$	$3.0518E-5$
$(0.2,0.2)$	$4.3275E-3$	$1.1390E-4$	$3.3379E-5$	$(0.02,0.02)$	$2.2931E-3$	$5.4267E-5$	$6.4850E-5$
$(0.3,0.3)$	$1.5107E-3$	$8.5512E-5$	$5.9128E-5$	$(0.03,0.03)$	$3.2326E-3$	$6.8786E-5$	$6.9618E-5$
$(0.4,0.4)$	$6.9533E-3$	$7.1149E-5$	$9.5367E-7$	$(0.04,0.04)$	$4.0429E-3$	$7.6382E-5$	$1.8120E-5$
$(0.5,0.5)$	$9.2423E-3$	$1.6350E-4$	$2.6703E-5$	$(0.05,0.05)$	$4.3703E-3$	$7.8073E-5$	$2.8610E-5$
$(0.6,0.6)$	$7.1619E-3$	$7.5883E-5$	$3.8147E-6$	$(0.06,0.06)$	$5.3013E-3$	$7.4808E-5$	$4.4823E-5$
$(0.7,0.7)$	$1.3627E-3$	$1.0452E-4$	0.0000	$(0.07,0.07)$	$5.7620E-3$	$6.7465E-5$	$2.8610E-5$
$(0.8,0.8)$	$5.3907E-3$	$1.4822E-4$	$4.6730E-5$	$(0.08,0.08)$	$6.1186E-3$	$5.6853E-5$	$3.9101E-5$
$(0.9,0.9)$	$8.1578E-3$	$4.2256E-5$	$7.0572E-5$	$(0.09,0.09)$	$6.3769E-3$	$4.3718E-5$	$4.5776E-5$

Table 3.15: Computed errors using GMQ method for $\epsilon = 2$, with $\beta = 1.99$. for Example (3.3).

(t,s)	$N=3$	$N=5$	$N=7$	(t,s)	$N=3$	$N=5$	$N=7$
$(0,0)$	$9.4587E-11$	$2.9802E-8$	$8.3447E-6$	$(0,0)$	$9.4587E-11$	$2.9802E-8$	$8.7738E-6$
$(0.1,0.1)$	$7.8363E-3$	$3.3665E-5$	$5.9605E-7$	$(0.01,0.01)$	$1.4912E-3$	$3.7260E-5$	$1.2279E-5$
$(0.2,0.2)$	$5.0778E-3$	$1.3251E-4$	$1.1921E-7$	$(0.02,0.02)$	$2.8007E-3$	$6.3568E-5$	$3.6955E-6$
$(0.3,0.3)$	$1.7276E-3$	$9.8798E-5$	$1.0729E-5$	$(0.03,0.03)$	$3.9381E-3$	$8.0518E-5$	$1.3113E-6$
$(0.4,0.4)$	$7.8789E-3$	$8.1595E-5$	$2.6226E-6$	$(0.04,0.04)$	$4.9128E-3$	$8.9295E-5$	$1.4303E-6$
$(0.5,0.5)$	$1.0361E-2$	$1.8596E-4$	$7.2718E-6$	$(0.05,0.05)$	$5.7338E-3$	$9.1351E-5$	$1.2636E-5$
$(0.6,0.6)$	$7.9579E-3$	$8.5540E-5$	$8.1062E-6$	$(0.06,0.06)$	$6.4101E-3$	$8.7589E-5$	$1.1086E-5$
$(0.7,0.7)$	$1.5321E-3$	$1.1674E-4$	$2.3842E-6$	$(0.07,0.07)$	$6.9503E-3$	$7.8894E-5$	$7.7486E-5$
$(0.8,0.8)$	$5.7535E-3$	$1.6326E-4$	$2.2650E-6$	$(0.08,0.08)$	$7.3626E-3$	$6.6661E-5$	$4.7684E-6$
$(0.9,0.9)$	$8.5560E-3$	$4.5802E-5$	$4.4107E-6$	$(0.09,0.09)$	$7.6553E-3$	$5.1074E-5$	$3.8147E-6$

Table 3.16: Computed errors using GMQ method for $\epsilon = 2$, with $\beta = 1.03$. for Example (3.3).

(t,s)	$N=3$	$N=5$	$N=7$	(t,s)	$N=3$	$N=5$	$N=7$
$(0,0)$	$2.9104E-11$	$2.8902E-8$	$3.2425E-5$	$(0,0)$	$2.9104E-11$	$2.9802E-8$	$3.2425E-5$
$(0.1,0.1)$	$5.3639E-3$	$4.2498E-5$	$1.9073E-6$	$(0.01,0.01)$	$9.8822E-4$	$4.6074E-5$	$1.9073E-6$
$(0.2,0.2)$	$3.5846E-3$	$1.6980E-4$	$1.7166E-5$	$(0.02,0.02)$	$1.8630E-3$	$7.8980E-5$	$1.1444E-5$
$(0.3,0.3)$	$1.2194E-3$	$1.2676E-4$	$3.8147E-6$	$(0.03,0.03)$	$2.6294E-3$	$1.0033E-4$	$2.5610E-5$
$(0.4,0.4)$	$5.6054E-3$	$1.0302E-4$	$9.5367E-6$	$(0.04,0.04)$	$3.2923E-3$	$1.1168E-4$	0.0000
$(0.5,0.5)$	$7.1552E-3$	$2.2967E-4$	$3.8147E-6$	$(0.05,0.05)$	$3.8562E-3$	$1.1444E-4$	$1.9073E-5$
$(0.6,0.6)$	$5.1188E-3$	$1.0233E-4$	$2.4796E-5$	$(0.06,0.06)$	$4.3271E-3$	$1.0988E-4$	$1.5259E-5$
$(0.7,0.7)$	$8.9421E-4$	$1.3201E-4$	$9.5367E-6$	$(0.07,0.07)$	$4.7084E-3$	$9.9283E-5$	$5.7220E-6$
$(0.8,0.8)$	$2.4403E-3$	$1.7087E-4$	$2.2888E-5$	$(0.08,0.08)$	$5.0053E-3$	$8.3860E-5$	$1.9073E-6$
$(0.9,0.9)$	$2.2223E-3$	$4.1999E-5$	$2.2888E-5$	$(0.09,0.09)$	$5.2223E-3$	$6.4623E-5$	$2.4796E-5$

Table 3.17: Computed errors using GMQ method for $\epsilon = 3$, with $\beta = \frac{1}{2}$. for Example (3.3).

(t,s)	$N=3$	$N=5$	$N=7$	(t,s)	$N=3$	$N=5$	$N=7$
$(0,0)$	$1.6007E-11$	$1.4901E-7$	$1.3828E-5$	$(0,0)$	$1.6007E-11$	$1.4901E-7$	$1.3828E-5$
$(0.1,0.1)$	$2.8338E-3$	$1.9103E-5$	$7.6259E-6$	$(0.01,0.01)$	$5.0513E-4$	$2.0802E-5$	$2.8610E-6$
$(0.2,0.2)$	$1.9371E-3$	$7.5012E-5$	$7.6294E-6$	$(0.02,0.02)$	$9.5631E-4$	$3.5524E-5$	$6.1989E-6$
$(0.3,0.3)$	$6.5883E-4$	$5.4508E-5$	$8.1062E-6$	$(0.03,0.03)$	$1.3552E-3$	$4.5478E-5$	$4.7684E-7$
$(0.4,0.4)$	$3.0039E-3$	$4.3601E-5$	$8.1062E-6$	$(0.04,0.04)$	$1.7036E-3$	$5.0515E-5$	$9.5367E-7$
$(0.5,0.5)$	$3.7284E-3$	$9.6530E-5$	$6.1989E-6$	$(0.05,0.05)$	$2.0032E-3$	$5.1588E-5$	$9.5367E-7$
$(0.6,0.6)$	$2.5425E-3$	$4.2766E-5$	$1.3828E-5$	$(0.06,0.06)$	$2.2559E-3$	$4.9502E-5$	$8.5831E-6$
$(0.7,0.7)$	$4.1992E-4$	$5.4866E-5$	$1.3828E-5$	$(0.07,0.07)$	$2.4634E-3$	$4.4584E-5$	$4.7684E-7$
$(0.8,0.8)$	$8.9515E-4$	$7.1347E-5$	$1.9073E-5$	$(0.08,0.08)$	$2.6276E-3$	$3.7581E-5$	$1.2875E-7$
$(0.9,0.9)$	$3.3706E-4$	$1.7911E-5$	$4.7684E-6$	$(0.09,0.09)$	$2.7505E-3$	$2.9057E-5$	$5.2452E-6$

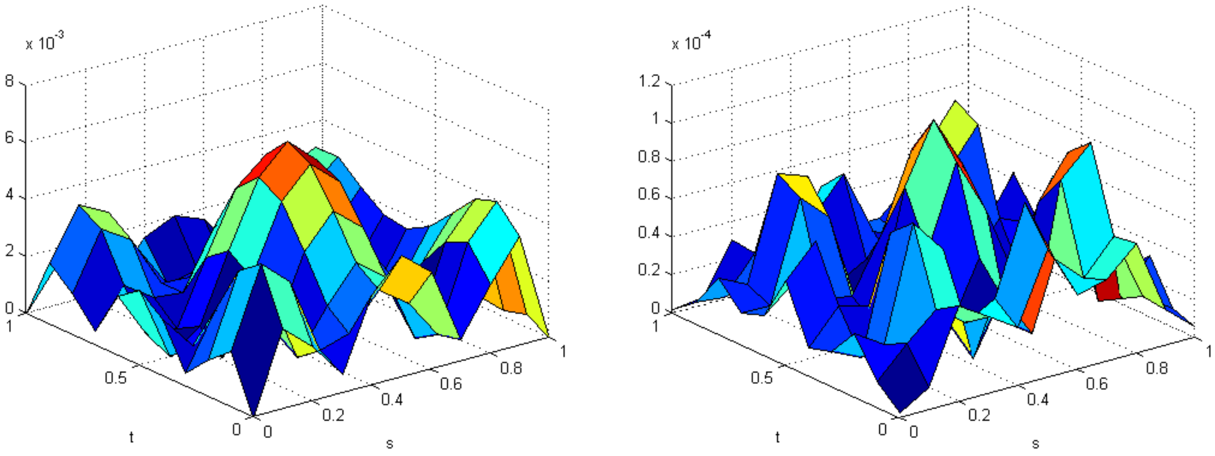

Figure 3.9: The Absolute errors for $N = 3$ (left), $N = 5$ (right) with $\beta = 1.03$, when $\epsilon = 3$.

3.6 Analysis of Results

In this chapter, an effective method for solving a class of two-dimensional mixed nonlinear Volterra–Fredholm integral equations is presented. This method compares interpolation using global and compactly supported radial basis functions. The shifted Gauss-Legendre quadrature formula is applied for numerical integration. Error estimate is provided for suffficiently smooth kernel and source functions. The convergence accuracy of the new scheme has been examined for several examples. From our test examples, we remark that in examples 1, 2 and 3 the optimal values of the exponent β give better results compared with that obtained by CSRBF (the choice of σ is obtained after several tests). Based on those findings, the two methods provide remarkable accurate solutions. The numerical results reported in tables and figures show that only a small number of collocation points is required to obtain a good approximation to the exact solution. An improtant property of the use of RBFs (Wendland's functions) is that they are positive definite so the problem is well posed. The use of optimal exponent β of the generalized multiquadric function is a good choice to performe the accuracy of the approximation. As a conclusion, from our experiments one can say that the optimal shape parameter and the condition number of interpolation matrix can also be taken in consideration for comparaison. By increasing the number of unknowns in a nonlinear system or by using very efficient solvers for a nonlinear system, the accuracy of the solution can be increased. The proposed method is simple because it does not need any background interpolation or approximation cells and so it is a meshless method. The test experiments are in particular encouraging to solve a large physical problems of time dependent and other problems based on solution representation. Generalizing the proposed method to local fractional equations is a good idea for future work. The method can also be easily adapted to solve higher dimensional integral equations. Another future application concerns the ability of the proposed method for solving stochastic problems by using chaotic radial basis functions.

Conclusion

In this work, we have been concerned with the study of the resolution of some linear and nonlinear integral and partial differential equations by using globaly and compactly supported radial basis functions, where we have provided a comprehensive theory based on some optimal strategies for choosing the shape parameter and centers of the RBFs. By using RBFs interpolation problem, the original problem is converted to a system of algebraic equations which can be solved by using numerical techniques. When using global radial basis functions, the resulting interpolation matrix is fully populated, the system may also become ill conditioned if very smooth radial basis functions are used on large number of points, both CPU time and ill conditioning increase with increasing the number of collocation points. This limits the use of global radial basis functions to a maximum number of collocation points which depend on the power of the computational platform. Other alternatives are used like domain decomposition and compactly supported RBFs. This results a sparse matrix coefficients and pass the above mentioned problems related to global RBF. Also, we have presented a complete numerical study of application of both golbaly and compactly supported for solving some integral and partial differential equations. A comparative study between stationary and non stationary approaches was discussed. Quadrature rules are also used on the approximation of the definite integrals that appear in our integral equation by a finite sum to seek an approximation solution in a finite number of points. An adaptive strategy based on generalized multiquadric RBFs with some optimal values of the exponent β is used for solving two-dimensional Volterra-Fredholm integral equations, and is compared with Wendland's CSRBF through our work, we have explained that the adaptive technique is suitable and very accurate for solving nonlinear integral equations and it has an exponential convergence order. We can also examin a stabilized RBF with a hybrid kernel generated, through a hybridization of gaussian and multiquadric RBF. This is used to improve the condition number of the interpolation matrix. Finally, we look forward to consider some strategies for improving the condition number, first by examining the relation between the condition number and the shape parameter and if this depend on the number N of interpolation points, secondly, presenting some high dimensional PDE problems survery what has been done with mesh based on stochastics methods. Finally to find a stable relation between the shape parameter and the accuracy of the RBFs approximation.

Some of perspectives of future work are

- Working with chaotic radial basis functions and looking if they can improve the accuracy and ill-conditioned of interpolation problems.
- Find the best radial basis function for solving integral equations in one and higher dimensional, by taking into account all the previous interpolation problems by RBFs.
- Also, finding some optimal strategies for Kansa's method for solving PDEs.

Bibliography

[1] Atkinson, K. E., *"The numerical evaluation of fixed points for completely continuous operators"*, **SIAM. J. Numer. Anal**. Vol. 10, No. 5, pp. 799–807, 1973.

[2] Atkinson, K. E., *"The Numerical Solution of Integral Equations of the Second Kind"*, **Cambridge University Press, Cambridge**, 1997.

[3] Atkinson, K. E., Flores, J., *" The discrete collocation method for nonlinear integral equations"*, **IMA. J. Numer. Anal**. Vol. 13, pp. 195–213, 1993.

[4] Baratella. P., *" A Nystrom interpolant for some weakly singular linear Volterra integral equations"*, **Comput. Appl. Math**. Vol. 231(2), pp. 725–734, 2009.

[5] Bartoshevich, M. A., *" A heat conduction problem "*, **J. Eng. Phys**. Vol. 28(2), pp. 240–244, 1975.

[6] Bochner, S., *" Monotone funktionen, stieltjessche integrale and harmonische analyse"*, **Math. Ann**. Vol 108, pp. 378–410, 1933.

[7] Bozzini, M. , Lenarduzzi, L. , Rossini, M. and Schaback, R., *" Interpolation by basis functions of different scales and shapes"*, **Calcolo**. Vol. 41(2), pp. 77–87, 2004.

[8] Brunner, H., *"Collocation Methods for Volterra Integral and Related Functional Equations"*, **Cambridge University Press**, 2004.

[9] Brunner, H., *" On the numerical solution of nonlinear Volterra-Fredholm integral equations by collocation methods"*, **SIAM J. Numer. Anal**, Vol. 27(4), pp. 987-1000, 1990.

[10] Brunner, H. and Kauthen, J. P., *" The numerical solution of two-dimensional Volterra integral equations by collocation and iterated collocation"*, **IMA J. Numer. Anal**. Vol. 9 (1) , pp. 47–59, 1989.

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

[12] Buhmann, M. D., *"Radial functions on compact support"*, **Proc. Edin. Math. Soc. II**. Vol. 41 , pp. 33–46, 1998.

[13] Buhmann, M. D., *"Radial basis functions "*, **in Acta Numerica Cambridge University Press**. Vol. 9, pp. 1–38, 2000.

[14] Buhmann, M. D. and Michelli, C.A., *" Multivariate interpolation in odd-dimensional Euclidean spaces using multiquadrics"*, **Const. Approx**. Vol. 6 (12) , pp. 21–34, 1990.

[15] BUHMANN, M. D. AND MICHELLI, C.A., *" Multiquadric interpolation improved"*, **Computers Math. Applic**. Vol. 24 (12) , pp. 21–26, 1992.

[16] CANUTO, C., HUSSAINI, M.Y., QUARTERONI, A. AND ZANG, TH.A., *" Spectral Methods: Fundamentals in Single Domains"*, **Springer-Verlag, Berlin, Heidelberg**, 2006.

[17] CAVERTTO, R., DE ROSSI, A. , MUKHAMETZHANOV, M. S. AND SERGEYEV, YA. D., *"On the search of the shape parameter in radial basis functions using univariate global optimization methods"*, **Journal of Global Optimization**. Vol. 79, pp. 305-327, 2019.

[18] CECIL T, QIAN J, OSHER S, *" Numerical methods for high dimensional Hamilton-Jacobi equations using radial basis functions "*, **Journal of Computational Physics** . Vol. 196(1), pp. 327–347, 2004.

[19] CHEN, W. , FU, Z.J. AND CHEN, C.S., *" Recent Advances in Radial Basis Function Collocation Methods "*, **Springer Brief in Applied Sciences and Technology**, Springer-Verlag, Berlin, Heilelberg, 2014.

[20] CHEN, W., HON ,Y.C. , LIN, J., *"The sample solution approach for determination of the optimal shape parameter in the Multiquadric function of the Kansa method"*, **Comput. Math. Appl**. Vol. 75, pp. 2942–2954, 2018.

[21] CHEN C.S., HON Y.C. AND SHABACK R. A., *"Scientific Computing with Radial Basis Functions"*, http://www.cityu.edu.hk/ma/staff/ychon/book/SCWRBF.pdf

[22] CHIN CHEN, CHANG GUO, DNG SU, CHIA CHENLIN., *" An improved Sudiku based data hidings Scheme using greedy method"*, **International Journal of Computational Science and Engineering**. Vol. 23, No.1, pp. 10–21, 2020.

[23] CHUI, C.K. , STOECKLER, J. AND WARD, J. D., *"Analytic wavelets generated by radial functions"*, **Adv. Comput. Math**. Vol. 5 , pp. 95–123, 1996.

[24] DALILA TAKOUK, REBIHA ZEGHDANE, BELKACEM LAKEHALI ., *" A new approach based on generalised multiquadric and compactly supported radial basis functions for solving two-dimensional Volterra-Fredholm integral equations"*, **International Journal of Computational Science and Engineering**. Vol. 25, No.5, pp. 532–547, 2022.

[25] DEHGHAN, M. AND SHOKRI, A., *" Numerical solution of nonlinear Klein-Gordon equation using radial basis functions "*, **J. Comput. Appl. Math**. Vol. 230, pp. 400-410, 2009.

[26] DONGDONG CHEN, TIANMIAO WANG, PEIJANG YUAN, NING SUN, HAIYANG TANG., *" A positional error compensation method for industrial robots combining error similarity and radial basis neural network"*, **Measurement Science and Technology**. Vol. 30. No. 12. 125010,2019. doi.org/10.1088/1361-6501/ab3311, 2019.

[27] FARENGO, .R, LEE, Y. C. , GUZDAR, P. N. , *" An electromagnetic integral equations: Application microtearing modes "*, **Phys. Fluid**. Vol. 26(12), pp. 3515–3523, 1983.

[28] FASSHAUER, G.E., *" Surface fitting and multiresolution methods"*, **edited by A. Le Méhauté, C. Rabut, L.L. Schumaker (Vanderbilt University Press**, pp. 131–138, 1997.

[29] Fasshauer, G. E., " *Meshfree Approximation Methods with Matlab*", **World Scientific Publishing, Interdisciplinary Mathematical Sciences** . Vol. 6, 520 pages, 2007.

[30] Fasshauer, G. E. and McCourt, M. J., "*Stable evaluation of gaussian radial basis function interpolants*", **SIAM Journal on Scientific Computing**. Vol. 34(2), pp. 737–762, 2012.

Fasshauer, G. E. and Zhang, J., " *Preconditioning of Radial Basis Function Interpolation Systems via Accelerated Iterated Approximate Moving Least Squares Approximation*", **volume 11 of Computational Methods in Applied Sciences. Springer Netherlands**, 2009.

[31] Floater M.S. and Iske. A, " *Thining algorithms for scattered data interpolation*", **BIT Numerical Mathematics**. Vol. 38(4), pp. 705–720, 1998.

[32] Foley, T. A., " *Near optimal parameter selection for multiquadratic interpolation*", **Journal of Applied Science and Computation**. Vol. 1, pp. 54–69, 1994.

[33] Fornberg, B. and Wright, G., " *Stable computation of multiquadric interpolants for all values of the shape parameter*", **Comput. Math. Appl**. Vol. 48(5-6), pp. 853–867, 2004.

[34] Franke, R., " *Scattered data interpolation: tests of some methods*", **Mathematics of computation**. Vol. 48, pp. 181–200, 1982.

[35] Franke, C. and Schaback, R., " *Solving partial differential equations by collocation using radial basis functions*", **Appl. Math. Comput**. Vol. 93, pp. 73–82, 1998.

[36] Hardy, R. L.," *Multiquadric equations of topography and other irregular surfaces*", **J. Geophys. Res**. Vol. 76 (8), pp. 1905–1915, 1971.

[37] Hardy, R. L.," *Theory and applications of the multiquadric-biharmonic method*", 20 years of discovery 1968-1988, **Computers Math. Applic**. Vol. 19 (8/9), pp. 163–208, 1990.

[38] Hendi, F.A and Albugami, A.M., "*Numerical solution for Fredholm-Volterra integral equation of the second kind by using collocation and Galerkin methods*", **Journal of King Saud University (Science)**. Vol. 22, pp, 37–40, 2010.

[39] Hon, Y.C. and Mao, X.Z., "*An efficient numerical scheme for Burger's equation* ", **Appl. Math. Comput**. Vol. 95 (1), pp. 37-50, 1998.

[40] Hon, Y.C., K.F. Cheung, K.F., Mao, X. and Kansa, E.J., " *Multiquadric solution for shallow water equations* ",**ASCE J. Hydraulic. Eng**. Vol. 125, pp. 524–533 , 1999.

[41] Hon, Y.C. and Mao, X.Z., " *A radial basis function method for solving options pricing model* ", **Financial. Eng**. Vol. 8, pp. 31–49, 1999.

[42] Hon, Y. C., "*An efficient numerical scheme for Burgers'equation* ", **Appl. Math. Comput**. Vol. 95 , pp. 37–50, 1998.

[43] Hu, Q. ," *Stieltjes derivatives and polynomial spline collocation for Volterra integro-differential equations with singularities*", **SIAM J. Numer. Anal. S3**. Vol. 1, pp. 208–220, 1996.

[44] Huang, C. S., Lee, C. F. and Cheng, A. D., *" Error estimate, optimal shape factor, and high precision computation of multiquadric collocation method"*, **Engineering Analysis with Boundary Elements**. Vol. 31, pp. 614–623, 2007.

[45] Huaiqing Zhang, Yu Chen, and Xin Nie., *"Solving the Linear Integral Equations Based on Radial Basis Function Interpolation"*, **Journal of Applied Mathematics**. Vol. 2014, pp. 1-8, 2014.

[46] Iske. A, *" Optimal distribution of centers for radial basis function methods "*, **Technical Report TUM-M 0004, Technische UniversitÄat MÄunchen** , 2000.

[47] Johnson Claes , *" Numerical Solution of Partial Differentiel Equations by the Finite Element Method- Dover Books on Mathematics "*, **Publisher: Dover Publication Inc**, 278 pages, 2009.

[48] Kadalbajoo, M., Kumar, A. and Tripathi, L., *" Application of local radial basis function based finite difference method for American option problem "*, **Int. J. Comput. Math**. Vol. 8, pp. 1608–1624, 2016.

[49] Kansa, E. J., *" Multiquadrics - a scattered data approximation scheme with applications to computational fluid-dynamics- surface approximations and partial derivative estimates "*, **I. Comput. Math. Applic**. Vol. 19, No. 8, pp, 127–145, 1990.

[50] Kansa, E. J., Wertz, J. and Ling, L., *"The role of the multiquadratic shape parameter in solving elliptic partial differential equations"*, **Computers and Mathematics with Applications**. Vol. 51(8), pp. 1335–1348, 2006.

[51] Kansa, E. J., *" Multiquadrics - a scattered data approximation scheme with applications to computational fluid-dynamics-solutions to parabolic, hyperbolic and elliptic partial differential equations "*, **I. Comput. Math. Applic**. Vol. 19, No. 8, pp, 147–161, 1990.

[52] Kansa, E. J. , *"Numerical simulation of two-dimensional combustion using mesh-free methods"*, **Engineering Analysis with Boundary Elements**. Vol. 33, pp. 940–950, 2009.

[53] Kansa, E. J. and Hon, Y., *"Circumventing the ill-conditioning problem with multiquadric radial basis functions: Applications to elliptic partial differential equations"*, **Computers and Mathematics with Applications**. Vol. 39(78), pp. 123–137, 2000.

[54] Kansa E. J ., *" Multiquadrics – a scattered data approximation scheme with applications to fluid dynamics – II: Solutions to parabolic, hyperbolic and elliptic partial differential equation"*, **Comput. Math. Appl**. Vol. 19, pp. 147–161, 1990.

[55] Li, X. and Chen, C. S. , *" mesh-free method using hyperinterpolation and fast fourier transform for solving differential equation"*, **Engineering Analysis with Boundary Elements**. Vol. 28, pp. 1253–1260, 2004.

[56] Lizka, T. and Orkitz J., *"The finite difference method at arbitrary irregular grids and its application in applied mechanics"*, **Comput. Struct**. Vol. 11, pp. 83–95, 1980.

[57] Mairhuber, J. C. , *"On Haar's theorem concerning Chebyshev approximation problems having unique solutions"*, **Proc. Am. Math. Soc**. Vol. (7) ,pp, 609-615, 1956.

[58] Maleknejad, K. and Jafari Behbahani, Z.," *Applications of two-dimensional triangular functions for solving nonlinear class of mixed Volterra–Fredholm integral equations*", **Math. Comput. Model**. Vol. 55, pp.1833–1844, 2012.

[59] Manzhirov, A. V., " *On a method of solving two-dimensional integral equations of axisymmetric contact problems for bodies with complex rheology* ", **J. Appl. Math. Mech**. Vol. 49(6), pp. 777–782, 1985.

[60] Marchi S.D., "*On optimal center locations for radial basis function interpolation: Computational aspects*", **Rend. Sem. Mat. Univ. Pol. Torino (Splines and Radial Functions)**. Vol. 61(3), pp, 343–358, 2003.

[61] Marchi S.D., Schaback R. and Wendland H., " *Nearoptimal data-independent point locations for radial basis function interpolation*", **Advances in Computational Mathematics**. Vol. 23, pp. 317–330, 2005.

[62] Meinguet, J., ,"*Multivariate interpolation at arbitrary points made simple*", **J. Appl. Math. Phys**. Vol. 30, pp.292– 304, 1979.

[63] Noor, M. A. , Noor, K. I., Waheed, A. and Al-Said, E. A., "*Some new solitonary solutions of the modified Benjamin-Bona-Mahony equation*", **Computer and Mathematics with Applications**. Vol. 62, pp. 2126–2131, 2011.

[64] Pathpatte, B. G., "*On mixed Volterra–Fredholm integral equations*", **Indian J. Pure. Appl. Math**. Vol. 17, pp. 488–496, 1986.

[65] Perrone, N. and Kao, R., " *A general finite difference method for arbitrary meshes*", **Comput. Struct.**. Vol. 5, pp. 45–58, 1975.

[66] Radlow, J. ," *A two-dimensional singular integral equations of diffraction theory* ", **Bull. Am. Math. Soc**. Vol. 70(4), pp. 596–599, 1964.

[67] Reza Firouzdor, Shukooh Sadat Asari and Majid Amirfakhrian.. " *Application of radial basis function to approximate functional integral equations*", **Journal of Interpolation and Approximation in Scientific Computing**. Vol. 2, pp. 77-86, 2016.

[68] Robert Schaback., " *Error estimates and condition numbers for radial basis function interpolation*", **Adv. Comput. Math.**. Vol. 3(3), pp. 251–264, 1995.

[69] Roque, C. M. C. and Ferreira, A. J. M., " *Numerical experiments on optimal shape parameter for radial basis functions*", **Numerical methods for Partial Differential Equations"**. Vol. 26, Issue 3, pp. 675–689, 2009.

[70] Saberi-Nadjafi, S., Mehrabienzhad, M. and Akbari, H. ," *'Solving Volterra integral equations of the second kind by wavelet-Galerkin sheme*" , **Computers and Mathematics with application**. Vol. 63(11), pp. 1536–15447, 2012.

[71] Sarra S.A., " *Adaptive radial basis function methods for time dependent partial differential equations*", **Applied Numerical Mathematics**. Vol. 54, pp. 79–94, 2005.

[72] Schoenberg, I. J., "*Metric spaces and completely monotone functions*", **Ann. of Math**. Vol. 39, pp, 811–841, 1938.

[73] Schaback, R. and Wu, Z. ,*"Operators on radial functions "*, **J. Comput. Appl. Math**. Vol. 73, pp. 257–270, 1996.

[74] Shivanian, E. , *"Analysis of meshless local radial point interpolation (MLRPI) on a nonlinear partial integro-differential equation arising in population dynamics "*, **Eng. Anal. Bound. Elem**. Vol. 37, pp. 1693–1702, 2013.

[75] Shokri, A. and Dehghan, M., , *" A Not-a-Knot Meshless Method Using Radial Basis Functions and Predictor-Corrector Scheme to the Numerical Solution of Improved Boussinesq Equation "*, **Comput. Phys. Commun** . Vol. 181, pp. 1990–2000 ,2010.

[76] Shmuel Rippa. ,*" An algorithm for selecting a good value for the parameter c in radial basis functions interpolation"*, **Advances in Computational Mathematics**. Vol. 11, pp. 193–210, 1990.

[77] Tatari, M. and Dehghan, M., *" A method for solving partial differential equations via radial basis functions: Application to the heat equation "*, **Eng. Anal. Bound. Elem**. Vol. 34(3), pp. 206–212, 2010.

[78] Tatari, M. and Dehghan, M., *"On the solution of the non-local parabolic partial differential equations via radial basis functions "*, **Appl. Math. Model**. Vol. 33(3), pp. 1729–1738 ,2009.

[79] Trahan, C. J. and Wyatt, R. E.,*" Radial bais function interpolation in the quantum trajectory method: Optimization of the multiquadratic shape parameter"*, **Journal of Computational Physics**. Vol. 185, pp. 27–49, 2003.

[80] Trefethen, L. N. and Bau, D., *" Numerical Linear Algebra"*, **SIAM, Philadelphia, 1st edition**, January 1997.

[81] Tsai, C., Kolibal, J. and Li, M., *" The golden section search algorithm for finding a good shape parameter for meshless collocation methods"*, **Eng. Anal. Boundary Elem**. Vol. 34(8), pp. 738–746, 2010.

[82] Wahba, G., *"Convergence rates of "thin plate" smoothing splines wihen the data are noisy, in Smoothing Techniques for Curve Estimation"*, **edited by T. Gasser, M. Rosenblatt, Lect. Notes Math**, Springer, Berlin, Heidelberg. Vol. 757 pp. 233–245, 1979.

[83] Wang, J. G and Liu, G.R. , *" On the optimal shape parameters of radial basis functions used for 2-D meshless methods"*, **Computer Methods in Applied Mechanics and Engineering**. Vol. 191, pp. 2611–2630, 2002.

[84] Wendland, H., *"Scattered Data Approximation"*, **Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge**, (2005).

[85] Wendland, H., *"Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree"*, **Adv. Comput. Math**. Vol. 4(1), pp, 389–396, 1995 .

[86] Wendland, H., *"Scattered Data Modelling by Radial and Related Functions"*, **Habilitation Thesis, Universitat Gottingen**, 2002.

[87] Wong, S.M., Hon, Y.C., Li, T.S., Chug, S.L. and Kansa, E.J. , *" Multizone decomposition of time dependent problems using the multiquadric scheme"*, **Comput. Math. Appl**. Vol. 37, pp. 23–45, 1999.

[88] Wu, Z., *"Dynamically knots setting in meshless method for solving time dependent propagations equation"*, **Computer Methods in Applied Mechanics and Engineering**. Vol. 193, pp. 1221–1229 (2004).

[89] Wu, Z., *"Compactly supported positive definite radial functions"*, **Adv. Comput. Math**. Vol. 4, pp, 283–292, (1995).

[90] Wu, .Z and Shaback R., *" Local error estimates for radial basis function interpolation of scattered data"*, **IMA J. Numer. Anal**. Vol. 13, pp. 13–27, 1993.

[91] Wu ,Z. and Hon, Y. C., *" Convergence error estimate in solving free boundary diffusion problem by radial basis function method"*, **Engineering Analysis with Boundary Elements.** . Vol. 27(1), pp. 73-79, 2003.

[92] *"Schemas sous forme conservative pour l'équation de Burgers"*: https://www.math.univ-paris13.fr/ basdevan/MACS1_EDP/Luneville/SchemaBurgers.html

[93] Xaio, J. R. and McCarthy, M. A., *" A local heaviside weighted meshless method for two-dimensional solids using radial basis functions"*, **Computational Mechanics**. Vol. 31, pp. 301–315, 2003.

[94] Xu, B.Z., Zhang, B.L. and Wei, G., *"Artificial neural network theory and application"*, **South China University of Technology Press, Guangzhou**, 1994.

[95] Xui Yu, Orian Zhang, Chao Xu., *" Numerical method for the time control problem governed by the Benjamin-Bona-Mahony equation"*, **International Journal of Computational Science and Engineering**. Vol. 13(3), pp. 296–302, 2016.

[96] Yang Wang, Chao Chen, Xuefeng Yan., *" Modified radial basis function neutral network integrated with multiple regression analysis and its application in the chemical industry procceses"*, **intelligent automation and soft computing**. Vol. 19(3), pp. 469–486, 2013.

[97] Zerroukat, M., Power, H. and Chen, C.S., *" A numerical method for heat transfer problems using collocation and radial basis functions"*, **Int. J. Numer. Methods .Eng**. Vol. 42(7), pp. 1263–1278, 1998.

[98] Zhang, S., Lin, Y. and Rao, M., *" Numerical solutions for second-kind Volterra integral equations by Galerkin methods"*, **Application of Mathematics**. Vol. 45(1), pp. 19–39, 2000.

- جامعة محمد البشير الإبراهيمي - برج بوعريريج -
UNIVERSITE MOHAMED EL BACHIR EL IBRAHIMI
BORDJ BOU ARRERIDJ

Faculté des Mathématiques et d'Informatique

Abstract :

This thesis deals with the applications of compactly supported radial basis functions for high dimensional reconstruction of surfaces (images) based on irregular samples. These methods without mesh (meshfree) based on the introduction of radial basis functions. Contrary to traditional methods, namely finite element (FEM) and finite difference (FDM) methods. We try to introduce the concept of this method through several applications.

Keywords :

Radial Basis Functions (RBF), multivariate interpolation, scattered data, numerical solution.

Résumé :

Cette thèse traite les applications des fonctions de base radiale, à support compact (CSRBF), pour la reconstruction bidimensionnelle de surfaces (images) à partir d'échantillons irréguliers. Ces méthodes sans maillage (meshefree) qui repose sur l'introduction des fonctions de base radiale. Contrairement, aux méthodes traditionnelles, a savoir la méthode des éléments finis (FEM) et la méthode des différences finies (FD). Nous essayons d'introduire le concept de cette méthode à travers plusieurs applications.

Mots clés :

Fonctions de base radiale (RBF), interpolation multivariée, données dispersées, solution numérique.

ملخص :

تتناول هذه الأطروحة تطبيقات وظائف الأساس الشعاعي المدعومة بشكل مضغوط لإعادة بناءالأسطح ذات الأبعاد العالية استنادا إلى عينات غير منتظمة. هذه الطرق بدون شبكة تستند إلى إدخال وظائف الأساس الشعاعي، على عكس الطرق التقليدية، كطريقة الفروق المحدودة وطريقة العناصر المحدودة. نحاول تقديم مفهوم هذه الطريقة من خلال عدة تطبيقات.

كلمات مفتاحية :

وظائف الأساس الشعاعية، استيفاء متعدد المتغيرات، بيانات مبعثرة، حل عددي .