Adaptive Radial Basis Functions for Option Pricing

Thesis submitted for the degree of

Doctor of Philosophy

at the

University of Leicester

by

Juxi Li

Department of Mathematics

University of Leicester

June 2015

“Our greatest weakness lies in giving up. The most certain way to succeed is always to try just one more time."

Thomas Edison

Abstract

In this thesis, we have developed meshless adaptive radial basis functions (RBFs) method for the pricing of financial contracts by solving the Black-Scholes partial differential equation (PDE). In the 1-D problem, we priced the financial contracts of a European call option, Greeks (Delta, Gamma and Vega), an American put option and a barrier up and out call option with this method. In the BENCHOP project with Challenge Parameter Set (Parameter Set 2) [97], we have shown that our adaptive method is highly accurate and with less computational cost in comparison with the finite difference method for the European call option and barrier up and out call option. And also we have presented the numerical result of the equally spaced RBF method for both Parameter Set 1 and 2. In our numerical simulations with Parameter Set 2, we note that our adaptive method is more accurate and faster than the equally spaced RBF method. For example for the barrier up and out call option, the equally spaced method (MQ) with 3000 uniform nodes has the maximum error of 1.30e-02 at three evaluation points, but our adaptive method (101 nodes) has maximum error of 9.98e-05 at the same three points. This is about 100 times better than the equally spaced method with about 30 times less CPU time. Since our adaptive strategy is accurate and efficient, we substantially increase the accuracy with fewer number of nodes.

We also developed an adaptive algorithm for the 2 assets Black-Scholes problem, in this algorithm we used the rectangular Voronoi points for the refinement, and the thin plate spline is used for the local approximation in order to assess the error. The numerical results of pricing a Margrabe call option are presented for both adaptive and non-adaptive methods. The adaptive method is more accurate and requires fewer nodes when compared to the equally spaced RBF method.

Acknowledgements

It is pleasure to acknowledge the help and support I have received from many people during an almost four years student life at University of Leicester. I would like to take this opportunity to acknowledge a few.

Firstly, I would like to express my utmost gratitude to my supervisor, Professor Jeremy Levesley, for his support, time and patience, who also has been my inspiration throughout my studies. I must say thanks to him for all of the help, encouragement and suggestions which are by no means limited to my research but have also proven valuable in my life. This thesis draws on his talents, knowledge and contribution.

I would like to thank all of staff in College House for their help and making so many things easier. I am grateful to Dr. Bo Wang, Dr. Andrea Cangiani and Dr. Manolis Georgoulis, and I also express my gratitude to my undergraduate tutor Professor Michael Tretyakov who referred me to Professor Jeremy Levesley.

On a personal level I would thank all of my friends in Michael Atiyah building, Oli, Sam, Matt, Ali, Ruhao, Yanshan and Yangzhang. Special thanks go to Jiayao, Jianxia, Masha and Laila who provided all the ultimate care, support and enjoyable time.

Finally, it gives me the most pleasure to thank my parents and my brother for the love, faith and support. Without their help this thesis would not be possible.

Contents

List of Figures

List of Tables

Abbreviations

AE = Absolute Error

ARBF = Adaptive Radial Basis Functions

ARSM = Adaptive residual subsampling method

BDF-2 = The Second Order Backward Differential Scheme

BUP = Boundary Update Procedure

bps = Basis Points

CPD = Conditional Positive Definite

DF = Degree of Freedom

FEM = Finite Element Method

FDE = Finite Difference Method

FVM = Finite Volume Method

GBM = Geometric Brownian motion

KdV = Korteweg de Vries

MQ = Multiquadric

OTC = Over The Counter

ODEs = Ordinary Differential Equations

PDEs = Partial Differential Equations

PD = Positive Definite

RBFs = Radial Basis Functions

RK4 = Fourth Order Explicit Range-Kutta Method

RMSE = Root Mean Square Error

SLM = Semi-Lagrangian Method

To my parents

Chapter 1

Introduction

1.1 Background

The three classical numerical methods are finite difference (FD), finite element (FE) and finite volume (FV), which have been developed successfully to solve many partial differential equations (PDEs) for decades. The behaviour of some material objects in nature can be modeled by PDEs, which means they can be solved by these classical numerical methods. For instance, weather and climate modeling in geophysics, biphasic model in engineering, the Maxwell's equations in electrodynamics and the Navier-Stokes equations in fluid dynamics, where it arises in our real life physical problems. In the engineering and other sciences, the finite difference method is well known for its simplicity. The main drawback of FDM is that it cannot handle higher dimensional geometrical objects as the discretization of PDE is based on a topological grid of line. FEM is the most flexible amongst these classical methods in term of geometry. Other methods, such as the spectral method, which is most accurate, but it has restrictions on geometry, and predefined periodic boundary conditions for the Fourier case. The new approach of solving PDEs by radial basis functions (RBFs) was proposed by Kansa [62]. Within this method the approximation depends on the pairwise distances between nodes, which is truly meshfree as no points and meshes are required to be connected. In general, computing the distances between points is less computationally costly than generating meshes in mesh based methods. In particular when the number of dimensions increases, it is difficult to generate the mesh for a mesh-based method, but for the RBF method, it does not increase the difficulty in terms of computing the distance between nodes. This means that the RBF method can be easily implemented for high dimensional problems with less computational cost in contrast to mesh-based methods. After Kansa [62] used the RBFs collocation method for a variety of engineering problems, it has gained significance popular in many research fields, such as mathematics, engineering, bioscience and finance.

For financial mathematics, apart from the three classical numerical methods, and spectral method, there are some different methods like Monte Carlo method, sparse grid method. The RBF method has been explored by [34, 48, 52, 53, 66, 80, 86]. In the next section we list some numerical methods which have been used to solve one or two dimensional Black-Scholes PDEs.

1.2 Numerical methods for the Black-Scholes equation

In 1973, an analytical formula of the European option was proposed by Black and Scholes [9]. Based on the assumption of risk neutrality, the log-normal Black-Schole partial differential equation is used to evaluate the European option value.

The difference between the European option and the American option is that the American option has an early exercise feature. Currently, there is no analytical solution existing for the American option. It can be solved if we treat it as a free boundary problem, and several numerical methods have been proposed to solve this free boundary problem; Finite difference method in [45], finite element method with penalty approach in [75], finite volume method in [8], lattice (or binomial) method in [23], projected SOR method in [25, 26], operator splitting method by Ikonen et al. [58], front-fixing finite difference method by Kwok et al. [64], Monte Carlo method in [48], recursive integration method in [57], a quasi-radial basis functions method in [52], a RBF collocation method in [34, 53, 80], and also Carr et al. [15] derived a way to solve the American put option problem based on the European option.

For multidimensional problems, such as a European or an American basket option, Reisinger et al. [86] proposed a method based on sparse grids to solve the European basket option. Fasshauer et al. [34] used the RBF method to solve an American basket option. The European basket option can be solved with an improved RBF method which is presented by Larsson et al. [66].

Apart from these methods, there are also some adaptive methods. For example, Persson et al. [70] proposed an adaptive finite difference method for European multi-asset option. Bungartz et al. [13] used an adaptive sparse grid method to solve the basket option problem. We have developed an adaptive RBF method to compute a Margrabe call option in this thesis. More details of adaptive RBF schemes are presented in the next section.

1.3 The RBF based adaptive methods

The adaptive radial basis function method is an alternative choice for solving both time dependent and time independent PDE problems. Here are some examples of RBFs based on adaptive schemes which have been proposed for PDE problems. Bozzini et al. [10] used multiquadric B-splines for their adaptive interpolation. The error indicator used for refinement is based on a predefined error threshold and the value of the root mean squared difference produced by scaled multiquadric interpolant. The linear convergence of an adaptive greedy algorithm has been presented by Hon et al. [54], where the method automatically concentrates on the largest residual error. Driscoll and Heryudono [28] illustrated their adaptive subsampling method (MQ basis functions) based on evaluating the residual error by inserting points half way between evaluation points. Behrens et al. [5] shows their adaptive method by combining both the semi-Lagrangian method (SLM) and RBF (the thin plate spline basis function) interpolation to solve the linear transport equation. Later, in [6] the method is extended to non-linear transport equations with the numerical example of the Burger's equation and the Buckley-Leverett equation being solved. The error residual depends on the difference between the approximation solution of PDE by SLM and reconstruction of local interpolation by the thin plate. Iske [60] used the finite volume particle method and scattered data reconstruction to the five spot problem. In his work, he discussed some numerical aspects approximation with polyharmonic spline, e.g., local approximation order, conditioning and numerical stability of the reconstruction problem. The combination of RBF and generalized finite difference stencil method have been proposed by Davydov [24]. Naqvi [77] used an adaptive RBF method to solve the 1-D Korteweg de Vries (KdV) equation, and her method is similar to adaptive method by Iske and Beherens, where she used the global and local RBFs approximation. In our adaptive method, we modify Naqvi's algorithm and extend it to compute a Margrabe call option via the 2-D Black-Scholes PDE.

1.4 Motivation

Most recently, Naqvi [77] successfully applied an adaptive RBFs algorithm for solving the one dimensional Korteweg de Vries (KdV) equation and Burger's equation. In this method, extra nodes are required in the high variation region, which is preferred over the equally spaced method in term of efficiency. Her result shows that the adaptive method can solve the KdV equation with less nodes and the same accuracy in comparison with the non-adaptive method. In other words, the distribution of nodes profile in the adaptive method reflects the profile of the solution. In the previous section, we note that many RBF based algorithms have been proposed for solving the Black-Scholes equation in the field of financial mathematics, but none of the adaptive schemes have been proposed for the multidimensional Black-Scholes equation. This is the main motivation which drives us to develop such an adaptive algorithm. Our adaptive algorithm is based on [6, 77]. In [6], Iske et al. have demonstrated an adaptive scheme to solve many different kinds of PDEs in 2-D, such as Burger's equation and the Buckley-Leverett equation. In their adaptive algorithm, the error indicator is a reflection of the local approximation quality of the interpolation around some specific node set. This useful error indicator can tell us where to refine the node and where not to. In PDE problems, this adaptive scheme can be summarised as high-order accurate, flexible with respect to geometry and easy to implement, which is the ideal numerical method.

1.5 Achievements

In this thesis, we are not aiming to produce an optimal algorithm for solving the Black-Scholes equation in both the 1-D and 2-D cases, but we show that the adaptive RBF method can numerically solve this problem in an efficient way. For instance, in [97] we demonstrated that our adaptive method could be 15 times faster than the FDM in terms of delivering the same accuracy for the European call option with Challenging Parameter Set. Also, for this sort of problem, the equally spaced RBF method could require at least 10 times the nodes to deliver similar accuracy solution when compared with our adaptive method (Details of numerical simulations presented in Chapter 4). Furthermore, we show that our adaptive algorithm works well in terms of accuracy in contrast to the equally spaced method for a Margrabe call option. In both the 1-D and 2-D cases of the

Black-Scholes equation, we illustrate that our adaptive method requires less nodes and can deliver similar accuracy when compared with the equally spaced RBF method.

1.6 Thesis outline

In Chapters 2 we document some definitions and notations of RBFs which will give an overview of the radial basis function. The multiquadric (MQ) and the Gaussian basis functions will be applied as a tool for approximating the solution of the Black-Scholes equation.

Chapter 3 focuses on the Black-Scholes model for one underlying asset, including the European call option, the American put option and the barrier up and out call option for the equally spaced RBF method.

Chapter 4 shows how our adaptive method performs for the same examples as in Chapter 3. Moreover, we introduce a new parameter set from [97] which might transforms our problem to a singularly perturbed PDE problem. Our adaptive algorithm for the 1-D Black-Scholes equation is presented, and our numerical results show that in the new parameter set, our adaptive method is highly accurate and faster than the equally spaced method, where the equally spaced method is struggling to deliver an accurate solution with small number of nodes (less than 200) which means a large number of nodes is required. In [97], we have shown our adaptive method is faster than FD method in terms of delivering a solution with relative error less than 1e-05 for an European call option, Delta, Gamma, Vega and barrier up and out call option.

Since we gain the insight of adaptivity of radial basis function for the Black-Scholes equation in one dimension, our adaptive method for two dimensional problem is presented in Chapter 5. A Margrabe call option is implemented in both adaptive and non-adaptive methods, the numerical results show our adaptive algorithm is more accurate than the equally spaced method. In our adaptive algorithm, we exclude the coarsen strategy which would be one of our development in our further work.

A summary of the thesis and further of the work is given in Chapter 6.

Chapter 2

Definition and Tools

2.1 Radial basis functions

Over the last two decades, due to the numerous advantages RBFs offer [30, 12, 99] the application of RBFs has had a fast development area in many research fields. Pena listed some RBF applications (Table 1) in [80], also he used the Kansa collocation method [62] to demonstrate a way to solve a range of different option pricing problems with RBFs in Wilmott magazine. In general, the advantages of RBFs are high-order accuracy, applicability in high dimension, easy to implement and mesh free. Some examples of these advantages are included as:

- High-order accuracy
  In [38], Larson and Fornberg concluded in elliptic problems stated that RBFs are more accurate than the standard second-order finite difference method (FDM) and the Fourier-Chebyshev pseudospectral method (PSM). Buhmann et al. and Madych et al. proved that infinitely smooth RBFs have spectral orders of convergence (faster than any polynomial order) in [73, 11].

- Applicability in high dimension
  In [17], Cecil et al. illustrated the example of solving the Hamilton Jacobi equation in 2 to 4 dimensions by using RBFs. Alternatively, finite difference (FDM), finite element (FEM), finite volume (FVM) can also be used to solve this problem [56, 95].

- Easy to implement
  Pena's eight step "cooking recipe" [80] showed that RBFs can be easily implemented for a range of different option pricing problems.

- Mesh free
  RBFs method only require an arbitrary set of nodes which does not require specific sub-domain which is extremely useful for high dimensional implementation, such as an American basket option [34, 66].

Furthermore, in complex geometries, infinitely smooth RBFs can achieve high order or spectral convergence when they have been applied to solve PDEs. With these advantages, RBFs could become one of the alternative ways to solve PDEs alongside more traditional methods (FDM, FEM and FVM).

In 1990, Kansa used the RBF (MQ) interpolation method to successfully approximate the numerical solution of partial differential equations (PDEs) of elliptic, parabolic and hyperbolic type [62]. As opposed to Kansa's collocation method, the symmetric collocation method was used by Wu [100] in 1992 which lets the basis functions also depend on the differential operators. Later, for initial and boundary conditions, Hon et al. extended this method to solve the engineering biphasic model with the nonlinear initial and boundary conditions in [22]. Fornberg et al. applied variate boundary treatments (edge enhancement techniques) to RBFs to investigate the problem of RBF approximations at the edge of an interval in both 1-D and 2D [19]. Furthermore, examples of option pricing (American and European option) with RBFs (Global RBFs and Quasi-RBFs) have been demonstrated by Hon in [53] [52], where the Boundary Updated Procedure (BUP) technique is applied to capture the free boundary condition problem in the American option. Hon also investigated the theoretical convergence of RBFs for the Black-Scholes equation [55]. In this thesis we use the Kansa collocation method [61, 62] which was mentioned above. Some others may use the symmetric radial basis function method [33, 39]. In this section, some definitions of RBFs interpolation and the Black-Scholes equation [9] are introduced.

2.1.1 Scattered data interpolation

Definition 2.1 (Scattered data interpolation problem)**.** Let $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \subseteq \mathbb{R}^d$, and data $f(\mathbf{x}_i)$, $i = 1, \ldots, N$, be given. The multivariate scattered data interpolation problem is to find a function $s : \mathbb{R}^d \to \mathbb{R}$ with data set $\mathbf{x}_i$, such that $s(\mathbf{x}_i) = f(\mathbf{x}_i), i = 1, \ldots, N$.

Scattered data interpolation for reconstructing an unknown function from a finite set of discrete data often appears to be a problem in practical applications. Finding a suitable algorithm of scattered data interpolation has been a problem in many areas of research for years. In 1982, Hardy's multiquadratic (MQ) has been ranked the best interpolation method among all twenty-nine interpolation methods in Franke's review paper [42], which means this could possibly be one of optimal algorithms to the scattered data interpolation problem.

2.1.2 Radial Basis Functions

One possible method of solving the multivariate scattered data function problem is using a linear combination of radial basis functions:

$$s(x) = \sum_{j=1}^{N} \lambda_j \phi(\|\mathbf{x} - \mathbf{x}_j\|), \quad \mathbf{x} \in \mathbb{R}^d, \tag{2.1}$$

where $\lambda_j, j = 1, \cdots, N$ is the unknown coefficient, and we use the Euclidean norm $\|\mathbf{x}\| = (\sum_{i=1}^{d} \mathbf{x}_i^2)^{1/2}$, and $\phi : \mathbb{R}_+ \to \mathbb{R}$ is a univariate function. There are many different types of RBFs; more details can be found in [69]. In Table 2.1 we have listed some common functions. Compactly supported radial basis functions have become popular, as they give rise to sparse matrices. Amongst the family of compact supported radial basis functions, the most popular one would be Wendland's functions [99]; examples are listed in Table 2.2. Wu's functions and other compact supported radial basis functions can be found in [32]. Franke examined and rated the Hardy's MQ to be one of the best interpolation methods in term of accuracy, speed and ease of implementation [42]. Later, Stead also concluded that MQ is more accurate for the gradient estimation from scattered data compared to some other methods [96]. Because of MQ's comparatively high accuracy, MQ is a favoured choice in the literature [80, 34, 53, 66]. In particular, MQ is also infinitely differentiable and the higher order partial derivatives of option price (*Greeks*, such

as, Δ and Γ) can be directly obtained by the derivative of the basis function of MQ and the Gaussian, where both MQ and the Gaussian are used for our basis functions from now on in this thesis.
The MQ and Gaussian basis functions are defined as

$$\phi(\|\mathbf{x}-\mathbf{x}_j\|)=\sqrt{(\mathbf{x}-\mathbf{x}_j)^2+c^2}, \tag{2.2}$$

$$\phi(\|\mathbf{x}-\mathbf{x}_j\|)=e^{-\frac{(\mathbf{x}-\mathbf{x}_j)^2}{c^2}}, \tag{2.3}$$

where c is the shape parameter which controls the shape of functions and has huge impact on both accuracy and the condition number of interpolation matrix. If a chosen a priori value of c (Table 2.1) is too large, the condition number increases significantly and the matrix tends to be badly ill-conditioned, which results in numerical instability. Moreover, in the case of a small value of c ($c \to 0$), an inaccurate solution is produced with well-conditioned linear system which is stated in [27]. In [89], Schaback observed that we cannot have both accuracy and a good condition number at the same time. The Contour-Pad$\acute{e}$ algorithm and RBF-QR have been proposed by Fornberg et al. [20, 40] to perform stably even when c tends to zero in 1-D. In [37], RBF-QR is extended to PDEs in three dimensions. Currently, there is not an easy way to determine the shape parameter, c a priori, as the choice is problem dependent [65]. The optimal shape parameter and the ill-conditioned interpolation problem are still active and open problem in RBF research. Throughout this thesis, we use the Kansa collocation method [62].

Name of RBFs	Functional Form $\phi(r) =$	Parameters
Gaussians	$e^{-(cr)^2}$	$c > 0$
Polyharmonic Splines	r^ν	$\nu > 0,\ \nu \notin 2\mathbb{N}$
Thin Plate Splines (TPS)	$r^{2k}\log(r)$	$k \in \mathbb{N}$
Multiquadric(MQ)	$(c^2+r^2)^{\frac{\nu}{2}}$	$\nu > 0,\ \nu \notin 2\mathbb{N},\ c > 0$
Inverse Multiquadric(IMQ)	$(c^2+r^2)^{\frac{\nu}{2}}$	$v < 0,\ c > 0$

TABLE 2.1: Example of some radial basis functions

Now we will introduce some basic, essential definitions. These can be found in any elementary textbook on RBFs (see [32] [99] [12]).

Definition 2.2 (Radial function). A function $\Phi : \mathbb{R}^d \to \mathbb{R}$ is defined as a radial if there is a univariate function $\phi : [0, \infty) \to \mathbb{R}$, such that $\Phi(\mathbf{x}) = \phi(r)$, where $r = \|\mathbf{x}\|$, and $\|.\|$ is some norm on $\mathbb{R}^d$ which is typically the Euclidean norm.

k	$\phi_{3,k}(r)$	smoothness
0	$(1-r)^2_+,$	C^0
1	$(1-r)^4_+[4r+1],$	C^2
2	$(1-r)^6_+[35r^2+18r+3],$	C^4
3	$(1-r)^8_+[32r^3+25r^2+8r+1],$	C^6

TABLE 2.2: Example of Wendland's compactly supported radial basis function $\varphi_{s,k}$,where $s=3$.

From the above definition, we know that $\phi(\|\mathbf{x}-\mathbf{x}_k\|)$ is a radial basis function and ϕ is a basic function. Suppose our finite data set $X \subseteq \Omega$ for some domain Ω in $\mathbb{R}^d$.

Definition 2.3 (Fill distance)**.** The fill distance is

$$h = h_{X,\Omega} = \sup_{\mathbf{x}\in\Omega} \min_{\mathbf{x}_j\in X} \|\mathbf{x}-\mathbf{x}_j\|, \tag{2.4}$$

and it is a measure that indicates how well the data set $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ fills out the domain Ω, and is the radius of the largest ball which does not intersect with any data.

Definition 2.4 (Separation distance)**.** The separation distance of $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ is defined by:

$$q = q_X = \frac{1}{2}\min_{i\neq j} \|\mathbf{x}_i-\mathbf{x}_j\|. \tag{2.5}$$

It gives the smallest possible radius between two data.

Definition 2.5 (Positive semi-definite matrices)**.** A real symmetric matrix A is called positive semi-definite if its associated quadratic form is non-negative, *i.e.*,

$$\sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i\lambda_j A_{ij} \geq 0, \tag{2.6}$$

for $\lambda = [\lambda_1, \dots, \lambda_N]^T \in \mathbb{R}^N$.

If the quadratic form (2.6) is zero only for $\lambda \equiv 0$, then A is called positive definite.

A positive definite matrix is non-singular, because only non-zero eigenvalues exist.

Definition 2.6 (Positive definite function)**.** A real valued continuous function Φ : $\mathbb{R}^d \to \mathbb{R}$ is positive definite if and only if it is even and

$$\sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i\lambda_j \Phi(\mathbf{x}_i-\mathbf{x}_j) \geq 0, \tag{2.7}$$

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

The function Φ is strictly positive definite on $\mathbb{R}^d$ if the quadratic form (2.7) is zero only for $\lambda \equiv 0$.

Theorem 2.7 (Bochner's Theorem). *A continuous function $\Phi \in C(\mathbb{R}^d)$ is positive definite on $\mathbb{R}^d$ if and only if it is the Fourier transform of a finite non-negative Borel measure μ on $\mathbb{R}^d$, i.e.*

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

Lemma 2.8 (Wendland). *The Gaussian $\Phi(r) = exp(-\alpha\|r\|^2)$, $\alpha > 0$, is positive definite on $\mathbb{R}^d$.*

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

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

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

Definition 2.9 (Conditionally positive definite function). A real valued continuous function Φ: $\mathbb{R}^d \to \mathbb{R}$ is conditionally positive definite of order m on $\mathbb{R}^d$ if and only if

$$\sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i\lambda_j\Phi(\mathbf{x}_i - \mathbf{x}_j) \geq 0, \tag{2.8}$$

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

$$\sum_{i=1}^{N} \lambda_i p(\mathbf{x}_i) = 0, \tag{2.9}$$

for any real valued polynomial p of degree at most $m-1$. The function Φ is called strictly conditionally positive definite of order m on $\mathbb{R}^d$ if the quadratic form (2.8) is zero only for $\lambda = 0$.

Theorem 2.10 (Buhmann, Micchelli). *Let $g \in C^\infty[0, \infty)$ be such that g' is completely monotonic but not constant. Suppose further that $g(r) \geq 0$. Then the interpolation matrix A is nonsingular for $\phi(r) = g(r^2)$.*

$A = \{\phi(\|\mathbf{x}_i - \mathbf{x}_j\|)\}_{\mathbf{x}_i,\mathbf{x}_j \in X}$ and X is set of points.

Proof. Since $g(r) \in C^\infty[0,\infty)$, then we have

$$g(r) = g(0) + \int_0^r g'(x)dx.$$

By changing the integral and replacing $g'(x)$ with the Bernstein-Widder representation. By Fubini's theorem:

$$g(r) = g(0) + \int_0^r \int_0^\infty e^{-\alpha x} d\mu(\alpha)dx.$$

Suppose $\lambda \in \mathbb{R}^X$ and $\sum_{i\in X} \lambda_i = 0$, then we have

$$\lambda^T A \lambda = -\int_0^\infty \sum_{i\in X}\sum_{j\in X} \lambda_i\lambda_j \alpha^{-1} e^{-\alpha\|i-j\|^2} d\mu(\alpha),$$

and

$$\int_0^r e^{-\alpha x}dx = -\alpha^{-1}e^{-\alpha r} + \alpha^{-1}.$$

Thus $\lambda^T A\lambda < 0$ for all λ and except $\lambda = 0$, this means there is one negative eigenvalue in A with remainder of positive eigenvalues. □

The proof of invertibility of Gaussian and MQ are directly taken from [99] and [12], respectively.

Definition 2.11 (m-unisolvent)**.** The set of points $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \subseteq \mathbb{R}^d$ is called m-unisolvent if the only polynomial of total degree at most m interpolating zero data on X is the zero polynomial.

Within a scattered data interpolation problem, there is a finite set of points $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \subseteq \mathbb{R}^d$ with corresponding value $f = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N)]$. Recalling Equation 2.1, we then have the following RBF interpolation:

$$s(\mathbf{x}) = \sum_{j=1}^N \lambda_j \phi(\|\mathbf{x} - \mathbf{x}_j\|), \quad \mathbf{x} \in \mathbb{R}^d. \tag{2.10}$$

With the above equation and corresponding value f, the value of coefficient λ_j can be found by enforcing the interpolation condition

$$s(\mathbf{x}_j) = f(\mathbf{x}_j), \quad j = 1, \cdots, N. \tag{2.11}$$

Rewriting Equation (2.11) as a linear system with $N \times N$ interpolation matrix A, $\lambda = [\lambda_1, \ldots, \lambda_N]^T$ and $f = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N)]^T$, we have

$$\begin{bmatrix} \phi(\|\mathbf{x}_1 - \mathbf{x}_1\|_2) & \phi(\|\mathbf{x}_1 - \mathbf{x}_2\|_2) & \ldots & \phi(\|\mathbf{x}_1 - \mathbf{x}_N\|_2) \\ \phi(\|\mathbf{x}_2 - \mathbf{x}_1\|_2) & \phi(\|\mathbf{x}_2 - \mathbf{x}_2\|_2) & \ldots & \phi(\|\mathbf{x}_2 - \mathbf{x}_N\|_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(\|\mathbf{x}_N - \mathbf{x}_1\|_2) & \phi(\|\mathbf{x}_N - \mathbf{x}_2\|_2) & \ldots & \phi(\|\mathbf{x}_N - \mathbf{x}_N\|_2) \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \end{bmatrix} = \begin{bmatrix} f(\mathbf{x}_1) \\ f(\mathbf{x}_2) \\ \vdots \\ f(\mathbf{x}_N) \end{bmatrix}.$$

This transfer the multivariate scattered data interpolation problem into the following linear system problem

$$A\lambda = f. \tag{2.12}$$

Precisely, to obtain the unique solution of $s = [s(\mathbf{x}_1), \ldots, s(\mathbf{x}_N)]^T$ and $\lambda = [\lambda_1, \ldots, \lambda_N]^T$, matrix A should be nonsingular which is true if it is positive definite. The condition number of A can be defined as

$$k(A) = \|A\|\|A^{-1}\| = \frac{\sigma_{\max}}{\sigma_{\min}}, \tag{2.13}$$

where $\sigma_{\max}$ and $\sigma_{\min}$ denote the maximum and minimum eigenvalues of A, respectively. An upper bound of $\|A^{-1}\|$ is given by Ball in [2] when $\phi(r) = \|r\|$, and Narcowich and Ward [78] gave the upper bound of $\|A^{-1}\|$ and upper bound condition numbers for $\phi(r) = \|r\|$ and $\phi(r) = \log(1+r^2)$ and MQ, $r \in \mathbb{R}^d$ and $\phi(\cdot) : \mathbb{R} \to \mathbb{R}^d$. In their results, the upper bounds on conditional number for the MQ in 2 and 3 dimension are $5.95\frac{\sqrt{1+D^2}}{p}e^{\frac{3}{p}}\left(\frac{D+2q}{2q}\right)^3$ and $8.55\frac{\sqrt{1+D^2}}{p}e^{\frac{4}{p}}\left(\frac{D+2q}{2q}\right)^3$, where q is half way of smallest distance or the separation radius and $D = \max_{i \neq j} \|\mathbf{x}_i - \mathbf{x}_j\|$, $\mathbf{x}_i \in X$. They also suggested that for small number of q, it is better to use the basis function of $\sqrt{4q^2 + r^2}$ as we can get the upper bound $\|A^{-1}\|$ to be less or equal to $\frac{6}{q}$. This is one of the reasons that we chose our shape parameter to be $4\Delta x$ (Δx is the distance between two neighboring nodes) in our equally spaced method in Chapter 3. In [79], the upper bound of Gaussian and thin plate spline are $\|A^{-1}\| \leq \frac{2sq^s}{\delta^s \rho(\frac{2\delta}{q})}$ and $\|A^{-1}\| \leq 2\sqrt{\pi}q^{-1}$, where $\rho(u)$ is strictly positive, decreasing and continuous on $(0, \infty)$, $\delta = 12\left(\frac{\pi\Gamma^2(\frac{s+2}{2})}{9}\right)^{\frac{1}{s+1}}$, Γ denotes Gamma function. Micchelli has discussed the RBFs in term of accuracy and condition numbers [76].

RBF intepolants can be extended with polynomial term which give us the guarantee of strictly positive definite. For instance, $X \subset \mathbb{R}^d$ and a function $\Phi : \mathbb{R}^d \to \mathbb{R}$ with a low degree polynomial of d variate and at most $m-1$ degrees P_m^d can be interpolated by RBFs with polynomial term

$$s(\mathbf{x}) = \sum_{j=1}^{N} \lambda_j \phi(\|\mathbf{x} - \mathbf{x}_j\|) + \sum_{i=1}^{M} d_i p_i(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^d, \tag{2.14}$$

where the polynomial $p_1, \ldots, p_M$ form a basis for the $M = \left(\begin{array}{c} m-1+d \\ m-1 \end{array} \right)$ dimensional linear space π_{m-1}^d of polynomial with degree of freedom less and equal of $m-1$ in d variables. From the definition of $(m-1)$-unisolvent, we are guaranteed a unique solution for the above interpolation problem and M is dimension of linear space π_{m-1}^d of total degree less or equal to $m-1$ in s variables.
The new linear system has been created by enforcing the interpolation condition $s(\mathbf{x}_i) = f(\mathbf{x}_i)$ with $i = 1, \ldots, N$. Finding the $N+M$ unknown coefficient values of λ_j and d_i, we also need an additional condition for the polynomial part to guarantee a unique solution of the N linear equations (the proof of which can be found in [32]). The following extra conditions are

$$\sum_{j=1}^{N} \lambda_j p_i(\mathbf{x}_j) = 0, \quad i = 1, \ldots, M. \tag{2.15}$$

From Equations (2.14) and (2.15), by enforcing $s = u$ and $u = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N)]^T$ we will have the following form

$$\begin{bmatrix} A & P \\ P^T & O \end{bmatrix} \begin{bmatrix} \lambda \\ d \end{bmatrix} = \begin{bmatrix} u \\ 0 \end{bmatrix}, \tag{2.16}$$

where $A_{lj} = \phi(\|\mathbf{x}_l - \mathbf{x}_j\|)$, with $l, j = 1 \ldots, N, P_{li} = p_i(\mathbf{x}_l), i = 1, \ldots, M$, $\lambda = [\lambda_1, \ldots \lambda_N]^T$,$u = [u_1, \ldots, u_N]^T$, $d = [d_1, \ldots, d_N]^T$, the O in the matrix form is $M \times M$ null matrix and 0 in the vector form is the zero vector with length M. In the remainder of this thesis, we only consider the case $M = 0$ (not adding any polynomial terms) for 1-D Black-Scholes problem as the thin plate spline is used to solve the spread option in Chapter 5. In a linear elliptic PDE with the Dirichlet boundary condition, the collocation equations are:

$$\begin{aligned} \mathcal{L}u(\mathbf{x}_i) &= f(\mathbf{x}_i) \quad i=1,\dots,N-N_B, \\ u(\mathbf{y}_j) &= g(\mathbf{y}_j), \quad j=N-N_B+1,\dots,N, \end{aligned} \tag{2.17}$$

i.e.

$$\begin{aligned} \textstyle\sum_{j=1}^{N} \lambda_j L\phi(\mathbf{x}_i-\mathbf{x}_j) &= f(\mathbf{x}_i), \quad i=1,\dots,N-N_B, \\ \textstyle\sum_{j=1}^{N} \lambda_j \phi(\mathbf{y}_i-\mathbf{x}_j) &= g(\mathbf{y}_i), \quad i=,N-N_B+1\dots,N, \end{aligned} \tag{2.18}$$

where L is a linear partial differential operator, N_B is the number of nodes on the boundary, ϕ denote the MQ radial basis function, u, g and f are real functions, where $\mathbf{y}$ is on the boundary domain of $\partial\Omega$. In the Kansa's collocation method [62], the approximate solution of u is defined as

$$\tilde{u}(\mathbf{x}) = \sum_{j=1}^{N} \lambda_j \phi_j(\mathbf{x}-\mathbf{x}_j), \quad \mathbf{x} \in \Omega \subseteq \mathbb{R}^d, \tag{2.19}$$

where $\lambda_j = [\lambda_1,\dots,\lambda_N]^T \in \mathbb{R}^N$ is an unknown coefficient vector. Equation (2.17) can be rewritten as

$$A\lambda = \begin{bmatrix} f \\ g \end{bmatrix}, \tag{2.20}$$

with

$$A = \begin{bmatrix} \tilde{A}_{\mathcal{L}} \\ \tilde{A} \end{bmatrix}, \tag{2.21}$$

where

$$\tilde{A}_{\mathcal{L},ij} = \mathcal{L}\phi(\|\mathbf{x}_i-\mathbf{x}_j\|), \quad i=1,\dots,N-N_B, j=1,\dots,N.$$

$$\tilde{A}_{ij} = \phi(\|\mathbf{x}_i-\mathbf{x}_j\|), \quad i=N-N_B+1,\dots,N, j=1,\dots,N.$$

and this solved the problem in a form of linear system.

2.1.3 Convergence

Truncation and rounding errors are two main type of errors when applying numerical schemes to approximate the solution of PDEs. The truncation error occurs by replacing a continuous problem with a discrete problem, and the rounding error occurs when we use finite precision. Schaback et al. used symmetric interpolation matrix to give the error bound of elliptic problem [41] and Fornberg et al. found the convergence rate of RBFs in 1-D of equally spaced points [36]. Fornberg et al. also noted that the RBF interpolation error structure oscillates around zero mean in their experiment ($\cos(wx)$). In financial applications, one of the localized RBF methods, the RBF partition of unity collocation method (RBF-PUM) has been tested for both stability and accuracy in [50] by Safdari-Vaighani et al.. There is no theoretical convergence analysis of our adaptive algorithm in this thesis and in any existing literature. Before going any further, we introduce some relevant and necessary definitions needed in the analysis of convergence of our adaptive method in the future.

Definition 2.12 (Reproducing kernel)**.** Let $\mathcal{H}$ be a real Hilbert space of functions $\mathrm{f} : \Omega(\subseteq \mathbb{R}^d) \to \mathbb{R}$ with inner product $\langle .,. \rangle$. A function $\Phi : \Omega \times \Omega \to \mathbb{R}$ is called a reproducing kernel for $\mathcal{H}$ if

1. $\Phi(.,\mathbf{y}) \in \mathcal{H}$ for all $\mathbf{y} \in \Omega$,
2. $f(\mathbf{y}) = \langle f, \Phi(.,\mathbf{y})\rangle_{\mathcal{H}}$ for all $\mathrm{f} \in \mathcal{H}$ all $\mathrm{y} \in \Omega$.

Theorem 2.13. *Suppose $\mathcal{H}$ is a Hilbert space of functions $f : \Omega \to \mathbb{R}$ with reproducing kernel Φ. Then we have*

1. $\Phi(\boldsymbol{x}, \boldsymbol{y}) = \langle \Phi(., \boldsymbol{x}), \Phi(., \boldsymbol{y})\rangle_{\mathcal{H}}$ *for* $\boldsymbol{x}, \boldsymbol{y} \in \Omega$.
2. $\Phi(\boldsymbol{x}, \boldsymbol{y}) = \Phi(\boldsymbol{y}, \boldsymbol{x})$ *for* $\boldsymbol{x}, \boldsymbol{y} \in \Omega$.
3. *If f_n converges to f in Hilbert space norm, then f_n also converges to f pointwise, e.g., for $n \to \infty$, $\|f - f_n\|_{\mathcal{H}} \to 0$ imply that $|f(\boldsymbol{y}) - f_n(\boldsymbol{y})| \to 0$ for all $\boldsymbol{y} \in \Omega$.*

Theorem 2.14. *Suppose $\mathcal{H}$ is a reproducing kernel Hilbert function space with reproducing kernel $\Phi : \Omega \times \Omega \to \mathbb{R}$ Then Φ is positive definite. Moreover, Φ is strictly positive definite if and only if the point evaluation functionals $\delta_{\boldsymbol{y}}$ are linearly independent in $\mathcal{H}^*$.*

$\mathcal{H}^*$ is the dual which is the space of bounded linear function on $\mathcal{H}$.

From the work of [99, 73], we know that smooth RBFs, such as Gaussian, inverse quadratics (IQ) and MQ on native spaces have been known to converge exponentially [73, 11]. Furthermore, Gaussian basis functions have small native space with analytical function [81], the proof of convergence were also given by Madych and Nelson in [72] that based on the fundamental work of [47]. Subsequently this has been extended by [76, 68, 82, 84, 90].

The convergence in scattered data interpolation was proved by Wu in [100], and the convergence rate of the RBFs method for the solution of PDEs were investigated in [41, 98]. For the shape parameter, Driscoll and Fornberg showed some result of convergence in both 1-D (converge to polynomial) and 2-D for smooth functions when c tends to zero with small number of N [27]. At equidistant distribution of center nodes, Platte and Driscoll applied the variable change, and found that there was a connection between polynomials and Gaussian interpolation [29]. For polynomial interpolation, the rate of convergence can be denoted by the method of classical approximation theory, such as the stability of the interpolation problem of Runge phenomenon analysis, which indicates the optimal distribution set of nodes. In Hermite interpolation, the investigation of rate of convergence of Hermite interpolation has been done by Luo and Levesley [67] with a modification method of variational approach of Madych and Nelson [71, 72] by a fixed conditional positive definite (CPD) function.

For the elliptic PDE problem, Carsten and Schaback [16] used symmetric collocation method to find an L_∞ error bound, and in term of L_2 norm, it has an additional convergence factor $h^{d/2}$ for MQ basis function.

2.2 Option pricing

Numerical methods for solving the Black-Scholes equation [9], which approximate the solution of a linear parabolic equation is an efficient way to price a derivative contract. However, with the number of assets increasing, the number of spatial spaces and computational demand are also increased, thus RBFs could be one method to resolve this problem. The Black-Scholes model has been used and extended to price a variety of options by many practitioners. The general setting for the underlying asset S in the Black-Scholes model is that S follows a Geometric Brownian motion (GBM) with

$$dS = \mu S dt + \sigma S dW, \tag{2.22}$$

where μ and σ are assumed to be constant, and represent the expected asset return and volatility of asset return, respectively. Based on GBM and the continuous hedging argument, the Black and Scholes PDE [9] is obtained by the following equation. This is a one dimensional non dividend payment Black-Scholes PDE for calculating the fair price of a European option with respect to time and underlying asset:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0, \tag{2.23}$$

where S follows a GBM in continuous time, σ is a constant volatility, r is a known and constant short-term interest rate and V is the option price.

2.2.1 European option

A European option (Vanilla option) is a derivative product that gives the holder the right, but not the obligation to exercise the option at end of its life time, at maturity time with a pre-agreed strike price. The discounted expected payoff function of the European option can be described as:

$$V_t = \mathbb{E}[e^{-r(T-t)} G(S_T)], \tag{2.24}$$

where K is the strike price.

$$G(S_T) = \max[\alpha(S_T - K), 0], \tag{2.25}$$

where $G(S_T)$ is the payoff function, $\mathbb{E}$ is the expectation in a risk-neutral world and α is equal to $+1$ for a call options and -1 for a put options. S_T is the underlying asset, T is maturity time and K is the strike price. European call (or put) options give the buyer the right to buy (or sell) an asset at maturity T.

2.2.2 American option

An American option can be executed at any time before the expiration date with pre-agreed strike price. It has the same payoff function Equation (2.25) as a European option. Since the value of American call option is the same as the value of European option, we only discuss the case of an American put option in our numerical example.

2.2.3 Barrier option

A barrier option is an option where the holder has the right to execute it if the underlying asset crosses (does not cross) a certain barrier level before the expiration date. There are many different kinds of barrier options on the market with both availability in call and put options. In general, barrier options can be divided into two categories and four basic path dependent forms, such as

- knock-out option:
  - up and out call/put
  - down and out call/put
- knock-in option:
  - up and in call/put
  - down and in call/put

The European style of barrier up-and-out call option will be demonstrated in our numerical experiments. The mathematical formula of the payoff function is

$$\text{Payoff} = \left\{ \begin{array}{ll} \max\{(S_t \text{ - K}), 0\ \}, & S_t < B, 0 \leq t \leq T, \\ 0, & \text{otherwise}, \end{array} \right\}, \tag{2.26}$$

where B is barrier level. In terms of premiums, barrier options have small premiums compared with European options.

2.2.4 Spread option

A spread option is an option whose underlying is based on the difference in price between two different assets. There is a wide variety of multi-asset options on the market. However, in this thesis, we mainly deal with a special case of the European style of spread option which has an analytical solution. In this spread option, for simplicity we consider $K = 0$ which is also called a Margrabe option [74]. The payoff function can be described as

$$\text{Payoff}(S_1, S_2) = \max\{(S_1 - S_2), 0\}. \tag{2.27}$$

Theorem 2.15 (*Itô* 's Lemma)**.** *Let* $\mathbf{B}(t)$ *be a Brownian motion and* $\mathbf{W}(t)$ *be an Itô drift-diffusion process which satisfies the diffusion equation:*

$$dW(t) = \mu(W(t), t)dt + \sigma(W(t), t)dB(t). \tag{2.28}$$

If $f(w, t) \in \mathbf{C}^2(\mathbb{R}^2, \mathbb{R})$*, then f(w,t) is also an Itô drift-diffusion process which satisfies the following differential equation:*

$$d(f(W(t), t)) = \frac{\partial f}{\partial t}(W(t), t)dt + \frac{\partial f}{\partial W(t)}(W(t), t)dW + \frac{1}{2}\frac{\partial^2 f}{\partial W(t)^2}(W(t), t)dt, \tag{2.29}$$

where $dt^2 = 0, dtdW(t) = 0$ *and* $dW(t)^2 = dt$*.*

Itô 's Lemma with two variables will be used in Chapter 5 for derivation of the two asset Black-Scholes formula.

Chapter 3

Option Pricing with Radial Basis Functions

3.1 European Call Option and Greeks

Let us recall the Black-Scholes equation for the non-dividend payment European option,

$$\frac{\partial V}{\partial t}+\frac{1}{2}\sigma^2 S^2\frac{\partial^2 V}{\partial S^2}+rS\frac{\partial V}{\partial S}-rV=0, \tag{3.1}$$

where S is the underlying asset value which follows a GBM in continuous time, σ is a constant representing volatility, r is a known and constant short-term interest rate and V is the price of the option. The initial condition of a European call option is given by the final payoff,

$$V(S,T)=\max\{S-K,0\}, \quad S\geq 0. \tag{3.2}$$

Here we have discretized both time and spatial spaces in a domain $\Omega\in[S_{\min},S_{\max}]\times[0,T]$ and the boundary conditions can be described as

$$\begin{array}{rcl} V(S_{\min},t) &=& \alpha(t), \quad 0\leq t\leq T, \\ V(S_{\max},t) &=& \beta(t), \quad 0\leq t\leq T. \end{array} \tag{3.3}$$

There are a few ways to numerically approximate the solution of the Black-Scholes equation, either solving it directly or with a variable change (it also can be transformed to heat equation). Firstly, we follow the eight step "cooking recipe" in [80] and [53] to reproduce the equally spaced RBF method. We propose a variable change with $x = \log(S)$ which is the same strategy in [53, 80], then, with the new payoff function $U(x,t) = V(S,t)$ Equation (3.1) can be rewritten as:

$$\frac{\partial U}{\partial t} + \frac{1}{2}\sigma^2 \frac{\partial^2 U}{\partial x^2} + \left(r - \frac{1}{2}\sigma^2\right)\frac{\partial U}{\partial x} - rU = 0. \tag{3.4}$$

Following the change of variables, the initial and boundary conditions can be expressed as

$$\begin{aligned} U(x,T) &= \max\{e^x - K, 0\}, \\ U(\log(S_{\min}), t) &= \alpha(t), \quad 0 \le t \le T, \\ U(\log(S_{\max}), t) &= \beta(t), \quad 0 \le t \le T, \end{aligned} \tag{3.5}$$

on the domain $\Omega = [x \times t]$ with $x = [\log(S_{\min}), \log(S_{\max})]$ and $t = [0, T]$.
By discretizing the domain with N nodes and M time steps, we use the RBF of form $\phi_j(x) = \phi(\|x - x_j\|)$ to approximate the solution of U:

$$u(x,t) = \sum_{j=1}^{N} \lambda_j(t)\phi_j(x) \simeq U(x,t), \tag{3.6}$$

where λ_j is an unknown time dependent coefficient vector. By substituting u in Equation (3.6) into Equation (3.4), we obtain the following linear system of equations with the initial and boundary conditions of Equation (3.5),

$$\frac{\partial u(x,t)}{\partial t} + \frac{1}{2}\sigma^2 \frac{\partial^2 u(x,t)}{\partial x^2} + \left(r - \frac{1}{2}\sigma^2\right)\frac{\partial u(x,t)}{\partial x} - ru(x,t) = 0. \tag{3.7}$$

In our numerical simulations, for a given set of data points $x_1, \dots, x_N$, we use MQ and Gaussian basis functions for the equally spaced method in 1-D which is presented in Table 3.1. Tensor products of RBFs are used in Chapter 5 to compute a Margrabe call option.

Since RBFs with uniform nodes are time independent, then the partial derivative of u with respect to time t is

$$\frac{\partial u(x,t)}{\partial t} = \sum_{j=1}^{N} \phi(\|x_i - x_j\|)\frac{d\lambda_j(t)}{dt}, \tag{3.8}$$

Name of RBFs	Functional Form $\phi =$
Gaussians	$e^{\left(-\frac{(x-x_j)^2}{c^2}\right)}$
Gaussians Tensor Product 2d	$e^{\left(-\frac{(x-x_j)^2}{c_1^2}\right)}e^{\left(-\frac{(y-y_j)^2}{c_2^2}\right)}$
Multiquadric(MQ)	$\sqrt{((x-x_j)^2+c^2)}$
Tensor Product of Multiquadrics 2d	$\sqrt{((x-x_j)^2+c_1^2)}\sqrt{((y-y_i)^2+c_2^2)}$

TABLE 3.1: Examples of radial basis functions in our numerical experiments

the first and second partial derivatives of u with respective to the underlying asset x can be denoted as:

$$\frac{\partial u(x,t)}{\partial x} = \sum_{j=1}^{N} \lambda_j(t) \frac{\partial \phi(\|x-x_j\|)}{\partial x}, \tag{3.9}$$

and

$$\frac{\partial^2 u(x,t)}{\partial x^2} = \sum_{j=1}^{N} \lambda_j(t) \frac{\partial^2 \phi(\|x-x_j\|)}{\partial x^2}. \tag{3.10}$$

For the MQ radial basis function, we have both first and second partial derivatives of ϕ as

$$\frac{\partial \phi(\|x-x_j\|)}{\partial x} = \frac{(x-x_j)}{\sqrt{(x-x_j)^2+c^2}}, \tag{3.11}$$

$$\frac{\partial^2 \phi(\|x-x_j\|)}{\partial x^2} = \frac{1}{\sqrt{(x-x_j)^2+c^2}} - \frac{(x-x_j)^2}{(\sqrt{((x-x_j)^2+c^2})^3}. \tag{3.12}$$

In the Gaussian case, the first and second partial derivatives of u are

$$\frac{\partial \phi(\|x-x_j\|)}{\partial x} = -2c^{-2}(x-x_j)e^{(-\frac{(x-x_j)^2}{c^2})}, \tag{3.13}$$

$$\frac{\partial^2 \phi(\|x-x_j\|)}{\partial x^2} = 4c^{-4}(x-x_j)^2 e^{-(\frac{(x-x_j)^2}{c^2})} - 2c^{-2}e^{(-\frac{(x-x_j)^2}{c^2})}. \tag{3.14}$$

The notation for u and its the partial derivatives are as follow:

$$u_x(x,t) = \frac{\partial u(x,t)}{\partial x} = \sum_{j=1}^{N} \lambda_j \frac{\partial \phi(\|x-x_j\|)}{\partial x},$$

$$u_{xx}(x,t) = \frac{\partial^2 u(x,t)}{\partial x^2} = \sum_{j=1}^{N} \lambda_j \frac{\partial^2 \phi(\|x - x_j\|)}{\partial x^2},$$

$$u_t(x,t) = \frac{\partial u(x,t)}{\partial t} = \sum_{j=1}^{N} \phi(\|x - x_j\|) \frac{d\lambda_j(t)}{dt},$$

$$u(x,t) = \sum_{j=1}^{N} \lambda_j \phi(\|x - x_j\|).$$

We collocate at the points x_i, $i = 1, \ldots, N$, and for notational simplicity, we use $A_x\lambda = \sum_{j=1}^{N} \lambda_j(t) \frac{\partial \phi(\|x_i - x_j\|)}{\partial x}$, $A_{xx}\lambda = \sum_{j=1}^{N} \lambda_j(t) \frac{\partial^2 \phi(\|x_i - x_j\|)}{\partial x^2}$, $A\lambda^t = \sum_{j=1}^{N} \phi(\|x_i - x_j\|) \frac{d\lambda_j(t)}{dt}$ and $A\lambda = \sum_{j=1}^{N} \lambda_j(t)\phi(\|x_i - x_j\|$. In matrix form, we can rewrite Equation (3.7) as

$$A\lambda^t + \frac{1}{2}\sigma^2 A_{xx}\lambda + \left(r - \frac{1}{2}\sigma^2\right) A_x\lambda - rA\lambda = 0. \tag{3.15}$$

Hon et al. mentioned that A in the equally spaced RBF is invertible [53] which is verified by Powell in [83]. The MQ is CPD of order 1, and the Gaussian is positive definite. Let's rewrite Equation (3.15) as:

$$\lambda^t = -\left[\frac{1}{2}\sigma^2 A^{-1} A_{xx} + \left(r - \frac{1}{2}\sigma^2\right) A^{-1} A_x - rI\right] \lambda. \tag{3.16}$$

Equation (3.16) also can be written as

$$\begin{aligned} \lambda^t &= P\lambda, \\ \text{where } P &= -\tfrac{1}{2}\sigma^2 A^{-1} A_{xx} - \left(r - \tfrac{1}{2}\sigma^2\right) A^{-1} A_x + rI, \end{aligned} \tag{3.17}$$

and I is the $N \times N$ identity matrix. In Equation (3.17), the matrix P can also be expressed as:

$$P = (P_{ij})_{1 \le i,j \le N},$$

where

$$\begin{aligned} P_{ij} = -\frac{1}{2}\sigma^2 \left(\phi(\|x_i - x_j\|)\right)^{-1} \frac{\partial^2 \phi(\|x_i - x_j\|)}{\partial x^2} \\ - (r - \frac{1}{2}\sigma^2) \left(\phi(\|x_i - x_j\|)\right)^{-1} \frac{\partial \phi(\|x_i - x_j\|)}{\partial x} + r\delta_{ij}, \end{aligned} \tag{3.18}$$

and δ_{ij} is the Kronecker delta function.

3.1.1 Time integration scheme

Equation (3.16) is a linear system of ordinary differential equations (ODEs) in time. We can obtain our coefficient vector λ with any backward time integration schemes, for example the Theta Method, fourth order explicit Range-Kutta Method (RK4) and the second order backward differential scheme (BDF-2). Details of these methods can be found in [80, 53, 66]. Our chosen time integration scheme is the backward Crank Nicolson scheme (CN). The first order homogeneous ODE with equalised time step Δt with initial conditions given by Equation (3.5) can be rewritten as

$$\frac{\lambda^t - \lambda^{t-\Delta t}}{\Delta t} = \frac{1}{2} P(\lambda^t + \lambda^{t-\Delta t}), \quad 0 < t \leq T. \tag{3.19}$$

Rearranging the above expression we obtain

$$\left(I - \frac{1}{2}\Delta t P\right) \lambda^t = \left(I + \frac{1}{2}\Delta t P\right) \lambda^{t-\Delta t}. \tag{3.20}$$

We define matrices L and R,

$$\begin{aligned} L &= \left(I - \tfrac{1}{2}\Delta t P\right), \\ R &= \left(I + \tfrac{1}{2}\Delta t P\right), \end{aligned} \tag{3.21}$$

which then gives the following linear system of equations,

$$\begin{aligned} L\lambda^t &= R\lambda^{t-\Delta t}, \\ \text{i.e. } \lambda^t &= L^{-1} R \lambda^{t-\Delta t}, \end{aligned} \tag{3.22}$$

$0 < t \leq T$, where L and R are $N \times N$ matrices.

3.1.2 Boundary updating system

Under the European call option framework, we need to satisfy the following boundary conditions:

$$\begin{aligned} V(0,t) &= 0, \quad 0 \leq t \leq T, \\ V(S,t) &= S - e^{-r(T-t)}K, \quad S \to \infty. \end{aligned} \tag{3.23}$$

Following the proposed variable change, the initial and boundary conditions of the log transformed Black-Scholes equation are

$$\begin{aligned} u(x,T) &= \max\{e^x - K, 0\}, \\ u(\log(S_{\text{min}}),t) &= 0, \quad 0 \leq t \leq T, \\ u(\log(S_{\text{max}}),t) &= S_{\text{max}} - e^{-r(T-t)}K, \quad 0 \leq t \leq T, \end{aligned} \tag{3.24}$$

where S_{max} and S_{min} represent the maximum and minimum values that we take to satisfy the boundary conditions in our spatial domain. In the following numerical experiments, we discretize our spatial space domain and time domain with N equally spaced nodes and with M equidistant time steps, respectively. For the Dirichlet boundary condition of first order time dependent ODEs (Equation (3.16)), we can use the Boundary Update Procedure (BUP) in [53], which states

```
For $n = M$ to 0.
Step 1: Compute $u^n = \phi\lambda^n$.
Step 2: Update the first and last elements $u^n(1)$ and $u^n(N)$.
Step 3: Calculate $\lambda^n$ by using $\phi^{-1}u^n$.
```

Here, n represents the number of time steps or iteration steps. For $n = M + 1$, the unknown initial coefficient vector $\sum_{j=1}^{N} \lambda_j^M$ can be obtained by $\phi^{-1}u^M(x)$. Throughout the iteration of BUP, we can find the unknown coefficient vector $\sum_{j=1}^{N} \lambda_j^0$ for $n = 0$, which is at $t = 0$. The option value can be obtained by calculating

$$U(x,0) \simeq \sum_{j=1}^{N} \lambda_j^0 \phi_j. \tag{3.25}$$

3.1.3 Method of Lines

In PDEs, the method of lines (MOL) consists of replacing the spatial derivatives with RBFs that transform PDEs into ODEs. For example, Shen used the MOL-RBF to solve time-dependent problem of Korteweg de Vries (KdV) equation in [93], Naqvi used adaptive RBF-MOL to solve the 1-D KdV equation and Burger's equations in her thesis [77]. In our case, we use RBF-MOL to solve the log transformation of Black-Scholes Equation (3.4), which can be rearranged as follows:

$$\frac{\partial u}{\partial t} + \mathcal{L}u = 0, \quad t \in [0, T], \tag{3.26}$$

where

$$u(x,t) = \sum_{j=1}^{N} \lambda_j(t)\phi(r_j), \quad i = 1, \ldots, N \quad \text{and} \quad r = \|x_i - x_j\| \tag{3.27}$$

This is the same as Equation (3.16) and the approximation of the solution can be obtained by any backward time integration scheme.

3.1.4 The shape parameter

In the equally spaced method, an appropriate shape parameter has been chosen in our MQ and Gaussian basis functions. Details of research literature on the optimal shape parameter are given in [35, 20, 14, 21, 51, 87]. In particular, Larsson et al. [66] used non-adaptive (non-uniform nodes) method to show the best shape parameter in their work is obtained by numerical simulation, which is defined as $c = 1 + \frac{N}{20}$ for a number of nodes that is between 20 to 60 in the 1-D Black-Scholes problem. The cross validation technique has been used to obtain the optimal shape parameter in [21, 51]. Pippa proposed the leave one out cross validation (LOOCV) algorithm based on the number of data distribution and computational precision [87]. Within our equally spaced method, we choose the shape parameter to be $4\Delta x$ for MQ, which was proposed by [80] and [53]. However, for the Gaussian basis function, we choose the shape parameter to be $c = 2\Delta x$, where Δx is the distance between two neighboring nodes. Figure 3.1 shows how the shape parameter c can affect the condition number and RMSE. The left figure shows that the condition number increases when c increases, and the right figure indicates that when both c and the condition number increase to certain level (below 1e+10), we can have a good RMSE, but we can see that when the condition number reach to 1e+10, in particular around 1e+15, our RMSE explode completely.

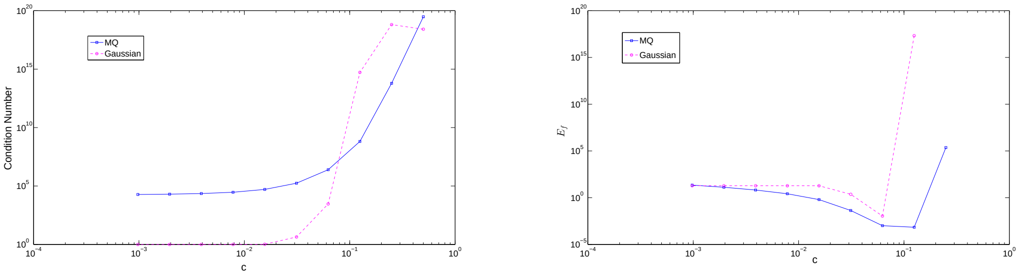

FIGURE 3.1: Condition number and RMSE in log-log plot for the shape parameter c.

3.2 Numerical Results

All of our numerical simulations are based on the problems of the BENCHOP project [97], which gives us comparisons to other methods of solution. In this simulation experiment, we will compare two radial basis function, MQ and Gaussian. Due to the choice of the shape parameter, there would be some oscillations in the Gaussian basis function. Moreover, we will use the same setting for both basis functions, such as the number of nodes, time steps and spatial domain. Since the error profile (absolute error plot in each time step) of both basis functions are broadly similar, and for clarity reason, we list all of them in Appendix A. In the following one dimensional problem, we present the numerical results between two standard RBF basis functions, MQ and Gaussian. In this section, we use the Parameter Set 1 from the BENCHOP project [97], which tests both basis functions for the European call option, Greeks, an American put option and a barrier up and out call option. Parameter Set 1 is

Parameter Values	
σ	0.15
r	0.03
T	1.0
K	100
B	125

TABLE 3.2: BENCHOP project, Problem 1 in [97], Parameter Set 1, B is the barrier level.

and B is the barrier level in the barrier up and out option.

3.2.1 Error measure

For computational purposes, we need to restrict our problem to a finite domain. We measure the error in the region $[\frac{4K}{5}, \frac{6K}{5}] = [80,\ 120]$, where value of the spot price is close to the strike price (K), and we know that there is very small probability that the stock will be out of the money or be further away from the strike. Let $P(x, 0)$ be the analytical solution of the Black-Scholes equation, x be the log transform of S and 0 is current time for t. The absolute error and the relative error functions are

$$E_{\text{abs}}(x) = |P(x, 0) - u(x, 0)|, \tag{3.28}$$

and

$$E_{\text{rel}}(x) = \frac{|P(x, 0) - u(x, 0)|}{P(x, 0)}, \tag{3.29}$$

where u is our approximation solution of U. Let N be some number of the evaluation points, and

$$x_j = x_{\min} + (j - 1)\Delta x, \quad j = 1, \cdots, N, \tag{3.30}$$

and

$$\Delta x = \frac{x_{\max} - x_{\min}}{N}. \tag{3.31}$$

The financial norm of the error is the max error, $E_{\max}$

$$E_{\max} = \max_{0 \le i \le N}(|P(x_i, 0) - u(x_i, 0)|), \tag{3.32}$$

where $i = 1, \cdots, N$.

The second error norm is the root mean square error (RMSE),

$$E_f = \sqrt{\frac{1}{N} \sum_{1 \le i \le N} |P(x_i, 0) - u(x_i, 0)|^2}, \tag{3.33}$$

All numerical experiments were performed using Matlab with a 64-bit Windows 7 operating system on a HP desktop with Intel Core i5 CPU @ 3.1GHz with 8GB RAM. None of the Matlab code has been optimized.

3.2.2 European call option

We first compare the approximation error of equally spaced methods for the European call option. In order to assess the accuracy of the numerical method, we use the max error ($E_{\max}$) and the RMSE (E_f), which compare the numerical solution with the analytical solution of Black-Scholes equation. In Table 3.3, we compare $E_{\max}$ of approximation solution of current call option value with maturity time T using MQ and the Gaussian basis functions with various number of nodes N and time steps M. Within this numerical experiment, the number of time steps are the same as the number of nodes in the spatial domain ($M = N$). As seen in Table 3.3, $E_{\max}$ decreases as the number of nodes and times steps increase, both basis functions can achieve $E_{\max}$ around 6.6e-04 for 640 nodes and 640 time steps. Figure 3.2 represents the RMSE against number of nodes, as we can see the RMSE value decreases as the number of nodes increases for N between 20 and 1280. For 160 nodes and time steps, the RMSE value is 2.258e-03 for the Gaussian and 1.330e-03 for MQ. In our numerical simulations, we try to avoid using a large number of nodes as it can give us an accurate solution, but it is computationally expensive. Table 3.3 and Figure 3.2 show that $N = M = 160$ delivers a solution with $E_{\max}$ around 2.3e-03 and E_f around 1.2e-03; this is our start point to test the equally spaced RBF method.

Basis function	N=20	N=40	N=80	N=160	N=320	N=640
MQ	1.13	2.01e-01	3.32e-02	1.33e-03	1.06e-03	6.36e-04
Gaussian	1.44	2.31e-01	4.16e-02	2.26e-03	1.16e-03	6.55e-04

Table 3.3: Max error ($E_{\max}$) of the RBF approximation for MQ and the Gaussian.

Figure A.1 and Figure A.2 represent the absolute error surface of the European call option in time for Parameter Set 1 in Table 3.2. The largest absolute error occurs when the spot price S is close to the strike price (K) at time step 1 with $E_{\max} = 2.452\text{e-}01$ in MQ and $E_{\max} = 3.45\text{e-}01$ in the Gaussian. We also note that there are several small peaks appearing at time step 1 as well. The numerical results in Figure 3.3 indicates that the relative error is below 1.7e-03 for both basis functions. In particular, when $S = K$ we have E_{rel} around 1.12e-04, and the RMSE for MQ and the Gaussian are 8.7994e-04 and 1.1e-03, respectively. The oscillation in the Gaussian is related to the choice of shape parameter. This is an issue with all Compact support functions.

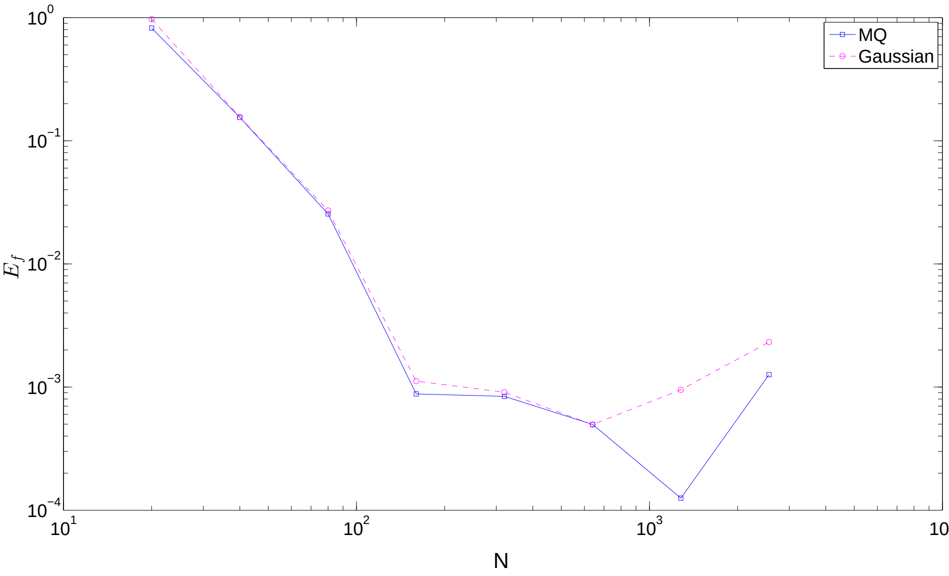

FIGURE 3.2: E_f of MQ and Gaussian against degree of freedom (DF)(number of nodes).

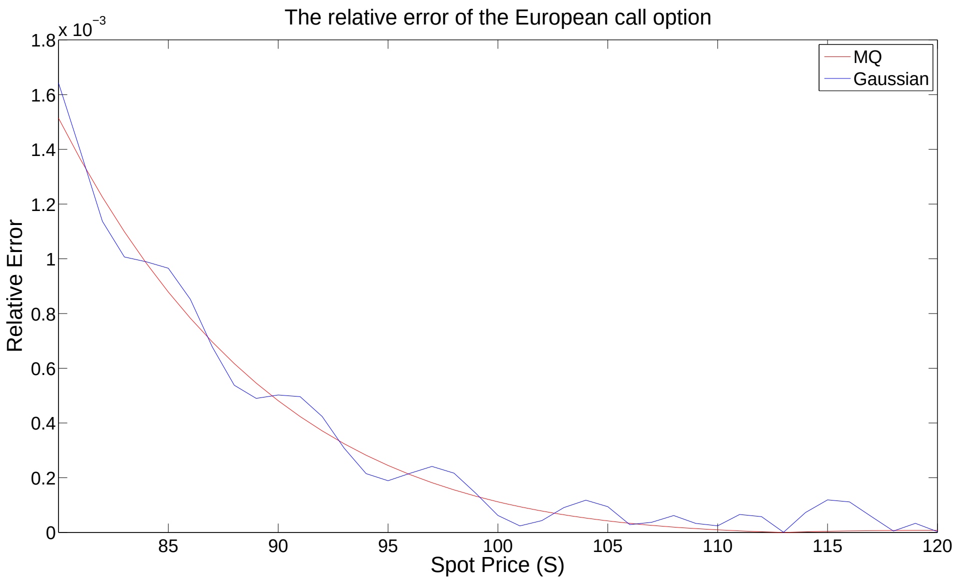

FIGURE 3.3: Relative error of European call option for MQ and the Gaussian basis functions at $t = 0$.

3.2.3 Greeks

In the financial market, the Greeks: Delta, Theta, Gamma, Vega, Rho and Psi, measure different facets of the risk of an option position. They are useful as they can predict the direction of movement of an option trade, and show how to protect the position against adverse movements in critical market variables. In this thesis, we discuss three Greeks: Delta, Gamma, and Vega.

Delta (Δ) is the sensitivity with respect to the underlying asset S. It is the rate of change in value with respect to the asset. Delta is defined as

$$\Delta = \frac{\partial V}{\partial S},$$

Gamma (Γ) is the second derivative of the option price with respect to the underlying asset S,

$$\Gamma = \frac{\partial^2 V}{\partial S^2}.$$

Vega (ν) is the measure of sensitivity of an option price with respect to volatility σ. Similarly to Δ, it gives the direction and extent of movement of the option price. Both call and put options have the same Vega, and Vega is always positive, as is volatility. The Vega is defined as the first derivative of underlying asset with respect to volatility σ

$$\nu = \frac{\partial V}{\partial \sigma}.$$

3.2.3.1 Delta

The value of Delta can be approximated by

$$\Delta(x,0) = \frac{1}{S}\sum_{j=1}^{N} \lambda_j^t \frac{\partial \phi(\|x - x_j\|)}{\partial x}, \tag{3.34}$$

where $x \in [\log(S_{\min}), \log(S_{\max})]$ and $t \in [0,T]$. In the RBF method, we can calculate Delta without any additional computational effort. Figure A.3 and Figure A.4 show the absolute error surface of Delta in time for Parameter Set 1. The structure of the absolute error surface behaves in Delta similar to the error surface of option value, though the error in Delta has more oscillation at time step 1 ($t = T$). At time step 1 $E_{\max}$ for MQ and the Gaussian are 1.473e-01 and 1.479e-01, respectively. Figure 3.4 shows the relative error of Delta for the spot prices from 80 to 120 at $t = 0$ with maturity time T, the Delta value of the Gaussian is highly oscillatory. The RMSE of Delta for MQ is 5.3434e-05, and 5.7118e-04 for the Gaussian, which means MQ has 10 times less RMSE value than the Gaussian.

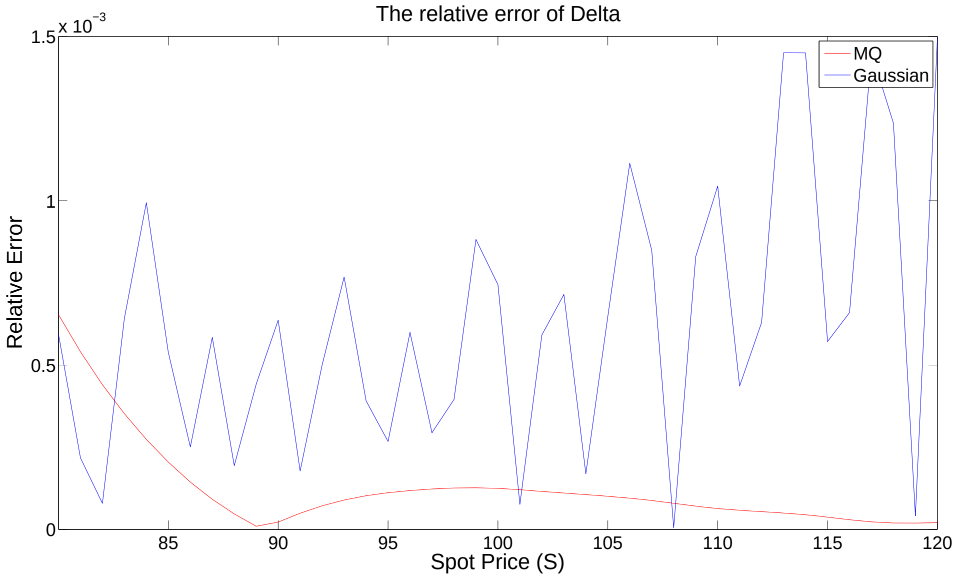

FIGURE 3.4: Relative error of Delta for MQ and the Gaussian basis functions at $t = 0$.

3.2.3.2 Gamma

The value of Gamma can be approximated with

$$\Gamma(x,0) = \frac{1}{S^2}\sum_{j=1}^{N}\lambda_j^t\frac{\partial^2\phi(\|x-x_j\|)}{\partial x^2}, \tag{3.35}$$

where $x \in [\log(S_{\min}), \log(S_{\max})]$ and $t \in [0,T]$. Gamma is the second partial derivative with respect to underlying asset S, and as before we do not require any additional computational effort to obtain the Gamma value in the RBFs method. Figure A.5 and Figure A.6 represent the absolute error surface of Gamma value in time, and two large kinks appear at $S = 95$ and 98. In Figure 3.5, we present the relative error with different spot prices at $t = 0$. As we can see the relative error of the Gaussian oscillates around the MQ. The maximum relative errors of MQ are smaller than the Gaussian. Figure 3.6 is the plot of absolute error against spot prices for MQ and the Gaussian basis functions. It indicates that $E_{\max}$ is just under 9.5e-03 which means the values of Gamma appear to be extremely small (maximum Gamma value is 0.028). The RMSE for MQ is 5.9e-03 and 6.0e-03 for the Gaussian.

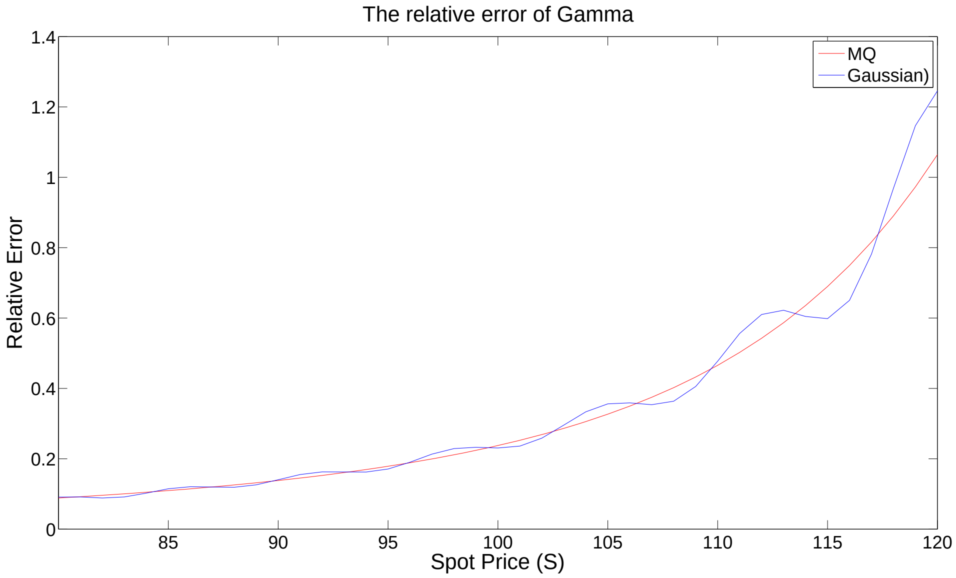

FIGURE 3.5: Relative error of Gamma for MQ and the Gaussian basis functions at $t = 0$.

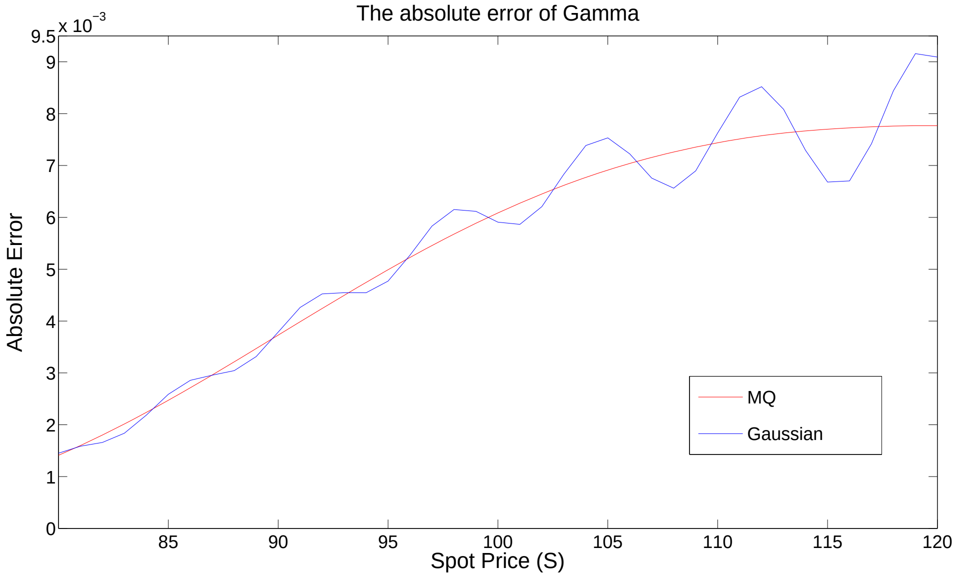

FIGURE 3.6: Absolute error of Gamma for MQ and the Gaussian basis functions at $t = 0$.

3.2.3.3 Vega

The Vega can be approximated with

$$\nu(x, \sigma) = \frac{u(x, \sigma + a) - u(x, \sigma)}{a}, \tag{3.36}$$

where a is an increment of σ. We tested our Vega with $a = 0.001\sigma$. In the following Table 3.4, we use the Parameter Set 1 which has been given in Table 3.2 with both N and M equal to 160 to calculate the relative error at three specific evaluation points. The result shows that we can deliver our approximation solution with high accuracy for both MQ and the Gaussian basis functions, with maximum of E_{rel} at 6.03e-04. Overall, MQ seems perform better than the Gaussian for Vega valuation, in particularly when $S = 110$, MQ is 5 times more accurate than the Gaussian.

Basis function	S = 90	S = 100	S = 110
MQ	5.44e-04	1.27e-04	1.17e-04
Gaussian	5.09e-04	1.56e-04	6.03e-04

TABLE 3.4: The relative error (E_{ref}) of Vega approximation for MQ and the Gaussian.

3.2.4 American Put Option

An American option is a contract that can be exercised at any time before maturity. The optimal exercise strategy is the one which maximises the payoff value. However, we do not know in advance what the optimal exercise strategy is. The concept of free boundary conditions is associated by the optimal exercise strategy as it separates the optimal exercise region from the holding region. The optional exercise price is the difference between exercise value and non exercise value, which depends on the remaining time of the contract (time to maturity), volatility and other parameters. Unlike the European option, a free boundary is involved, which poses a difficulty for most numerical approximation methods. To date, no analytical solution of American options exists. In [15], Theorem 1.1 showed that we can use the value of European put option with the earlier exercise premium to replace the value of an American put option. The value of a European put option is

$$V_E = -SN\left(-d^+\left(\frac{S}{K}, T-t\right)\right) + Ke^{-r(T-t)}N\left(-d^-(\frac{S}{K}, T-t)\right), \tag{3.37}$$

where $d^{\pm} = \frac{1}{\sigma\sqrt{T-t}}[ln\frac{S}{K} + (r \pm \frac{\sigma^2}{2})(T-t)]$ and $N(.)$ is the standard normal distribution function. The value of an American put option can then be written as

$$V_A = V_E + rK\int_t^T e^{-r(T-u)}N(-d^-(\frac{b(t)}{b(u-t)}, T-u))du, \tag{3.38}$$

where $0 \leq t < T$ and $b(t)$ is the optimal exercise strategy at time t with $b(T) = K$. More details of this method and proofs can be found in [15]. In Table 3.5 we present numerical results of an American put option that shows that RBFs can capture the earlier exercise feature. The initial and boundary conditions can be described as

$$\begin{aligned} U(x,T) &= \max\{e^x - K, 0\}, \\ U(\log(S_{\min}), t) &= Ke^{-r(T-t)}, \quad 0 \leq t \leq T, \\ U(\log(S_{\max}), t) &= 0, \quad 0 \leq t \leq T, \end{aligned} \tag{3.39}$$

and the earlier exercise feature is described as

$$V(S_f(t), t) = \max\{K - S_f(t), 0\}, \quad 0 \leq t \leq T, \tag{3.40}$$

$$\frac{\partial V(S_f(t), t)}{\partial S} = -1, \quad 0 \leq t \leq T, \tag{3.41}$$

where $S_f(t)$ is the free boundary (or the optimal exercise strategy). In our numerical implementation, we used the Boundary Update Procedure given in [53]. There is a small modification in the step 2:

Step 2: Updating the entire element of $u^n(i)$ with $u^n(i) = \max\{V(S_f(n), n), u^n(i)\}$, where $i = (1, \ldots, N)$.

Results presented in Table 3.5 show that both MQ and the Gaussian can achieve a maximum E_{ref} of 3.45e-04 over the three evaluation points. We use the same Parameter Set 1 in Table 3.2 and keep the same value of M and N as 160. In our computation, we take our spatial domain as $S \in [1, 2K]$. The reference value we used is based on [97].

Basis function	S = 90	S = 100	S = 110
MQ	9.79e-05	3.45e-04	2.02e-04
Gaussian	1.03e-04	3.44e-04	1.75e-04

TABLE 3.5: The relative error (E_{ref}) of approximation solution of American put option for MQ and the Gaussian.

3.2.5 Barrier up and out call option

The last example in Problem 1 is the barrier up and out call option. In this problem, we show an example of the RBF method that can be used to capture the discontinuity of the payoff function. The barrier up and out call option is the same as a European option if and only if the underlying asset does not hit a pre-specified barrier level. Once the underlying asset hits a pre-specified barrier level, then the option becomes worthless. Figure A.7 and Figure A.8 represent the absolute error surface with respect to time for Parameter Set 1. Unlike for Figure A.1 for the European option, the largest absolute error occurs when the spot price approaches the barrier level ($b = 125$). The largest error seems to be massive (over 1.4). We test it further with the relative error for the barrier up and out call option and plot it in Figure 3.7. The maximum value of E_{ref} for MQ is 9.6e-03 and 3.28e-02 for the Gaussian. The RMSE for MQ is 1.77e-02 and 4.39e-02 for the Gaussian.

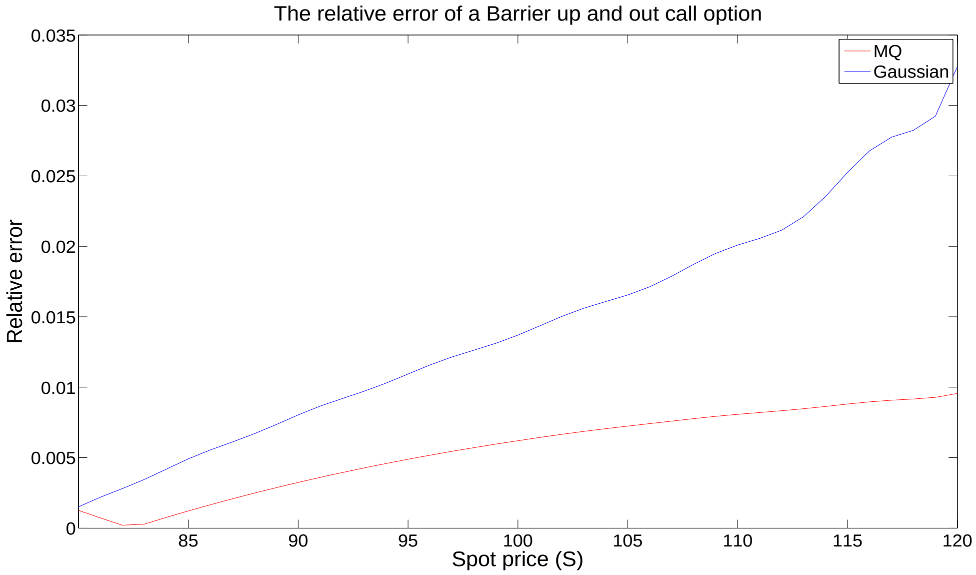

FIGURE 3.7: Relative error of a barrier up and out call option for MQ and the Gaussian basis functions at $t = 0$.

3.2.6 Conclusion

In a European call option, we see that the Gaussian oscillates around MQ which means its first and second derivatives are more oscillated than itself, where the relative error of Delta and Gamma plots (Figure 3.4 and Figure 3.5) in our numerical simulations have proved that. However, since the relative error of a European call option for both basis functions is below 1.8e-3 in comparison with an analytical solution, we can say the equally spaced RBF method did well in term of accuracy and CPU time as both basis functions require around 0.2 second time in Matlab. In Vega valuation, we see that both basis functions can achieve high accuracy in term relative error (around 6.04e-4 in Table 3.4). An American put option, the relative error for these three evaluation points are of approximately at 3.46e-4 (Table 3.5), this is also high accuracy in comparison with our reference value. In the last experiment, we see that the MQ has a small relative error (around 1e-2) for 40 evaluation points (80 to 120) in contrast to the Gaussian. Overall, MQ is more stable and more accuracy than the Gaussian in our experiments, thus we use MQ for remainder. As we have noted that the error mainly occurred around the strike price (K) where the initial condition is applied. In the next Chapter, we propose an adaptive method based on RBFs which can reduce the error caused by the initial non-smooth payoff function. Moreover, we aim to use a minimum number of nodes to achieve reasonable accuracy.

Chapter 4

Option Pricing with Adaptive Radial Basis Functions

4.1 Introduction

In this section, we describe our adaptive radial basis functions method (ARBF) as follows: By a given set of chosen interpolation points $\mathbf{x}_1, \ldots, \mathbf{x}_N \in \mathbb{R}$, the adaptive algorithm uses a predefined error indicator to move the nodes automatically in space, which could minimize the memory usage to limited nodes at each time (dimension) in comparison with the fixed nodes algorithm. There are some other popular adaptive schemes, such as [6, 77]. The numerical results of our adaptive method are presented in this chapter. In addition, we compare our result with the equally spaced RBF method in [80, 53]. Our adaptive method has several advantages in comparison with the equally spaced method which we illustrate through numerical experiments. For example, it is highly accurate, stable and efficient as fewer nodes are required. The numerical result of a European call option, a barrier call up and out option, and an American put option is produced with both adaptive and non-adaptive methods.

4.2 Adaptive Radial Basis Functions Method for One Dimensional Black-Scholes Equation

The one dimensional non-dividend payment Black-Scholes equation is

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0, \tag{4.1}$$

where S follows a GBM in continuous time, σ is a constant volatility, r is a known and constant short-term interest rate and V is the price of option. Proposing the same variable change in Chapter 3 with $x = \log(S)$, and discretizing the domain with N equally space nodes, we will have our RBF $\phi_j = \phi(\|x - x_j\|)$ used for approximating the solution of U, such as

$$\frac{\partial U}{\partial t} + \frac{1}{2}\sigma^2 \frac{\partial^2 U}{\partial x^2} + (r - \frac{1}{2}\sigma^2)\frac{\partial U}{\partial x} - rU = 0, \tag{4.2}$$

$$\phi(\|x - x_j(t)\|) = \sqrt{(c(t)(x - x_j(t)))^2 + 1}, \tag{4.3}$$

and

$$u(x,t) = \sum_{j=1}^{N(t)} \lambda_j(t)\phi(\|x - x_j(t)\|), \tag{4.4}$$

where $\lambda_j(t)$ is the unknown time dependent coefficient vector set. N and x_j change because we are using adaptive techniques.

4.2.1 Adaptive algorithm

In general, most of the adaptive methods for PDEs have their own adaptive cycle, which can be divided into 4 separate parts. These are the approximation of the solution, error indicator, coarsening or refinement and data output. Here are a few examples of adaptivity strategy in literature. Driscoll and Heryudono illustrated the adaptive residual subsampling method [28] which evaluates the error at half way of the initial points. Later, Chan [18] used the same adaptive method for the classical Merton jump-diffusion problem. We follow closely the algorithm in Naqvi's thesis [77], which has a similar adaptive structure as adaptive residual subsampling method (ARSM) as Driscoll et al. and Chan in [28, 18]. They compute the interpolation error at evaluation points half way between initial interpolation points.

The adaptive cycle is done by MQ basis functions with MOL, it transforms our PDE to ODE, and solve it with any backward time integration schemes. In each adaptive cycle, the approximation solutions are generated by MQ basis functions, which is corresponding to the "Solve" step. Our error indicator deicides whether we require refinement or coarsening. In our error indicator, we compute the interpolation error between the global approximation and the local approximation. The local approximation solutions are generated by the piecewise cubic spline, where the cubic spline is reconstructed by 8 nearest points for all evaluation points, x_i and $2 \leq i \leq N-1$. A good approximation means the residual ($r_{\text{err}} = \mid u_N(x_i) - \tilde{u}_{\mathcal{N}_{x_i}}(x_i) \mid$) is below the redefined error thresholds err_{ref}. $u_N(x)$ denotes the solution of global approximation at x and $\tilde{u}_{\mathcal{N}(x)}$ is the solution of local approximation at x. The decision of coarsening or refining is based on the error assessment, which can be generalised as refining if r_{err} is great than the predefined error thresholds (err_{ref}) and coarsening if r_{err} is below the predefined error thresholds (err_{crs}). We insert nodes at half way between initial interpolation nodes for our refinement phase. The coarsening strategy is a bit different from the refinement strategy, because removing too many nodes could cause large gaps between nodes and possibly lead our interpolation matrix to be ill-condition. Therefore, we introduce a maximum distance of nodes to avoid this possibility and the maximum number of nodes we can remove is about 50% of new adding nodes. To avoid removing too many nodes in a certain region, we also give a rule which allow us to remove about 50% of coarsening nodes in each side. When the coarsening or refinement cycle ends, we re-adapt the shape parameters. The Adaptive cycle ends if all r_{err} are below r_{err}; then the time integration scheme is applied. Figure 4.1 shows an example of our refinement in the 1-D problem. In this example, we assume the red node to be the refinement node for both figures, and after the refinement strategy, the blues nodes are the new added nodes, which are presented in the figure on the right. Figure 4.2 shows an example of our coarsening strategy in the 1-D problem. In this problem, we assume the maximum distance of nodes is 1, our refinement and coarsening region is in [2, 4], the total number of coarsening nodes is 5 (blue nodes in the figure on the left) and the number of new added nodes is 4. Under these conditions, we know the maximum number of nodes we can remove is 2. The blue nodes in the left figure denote the

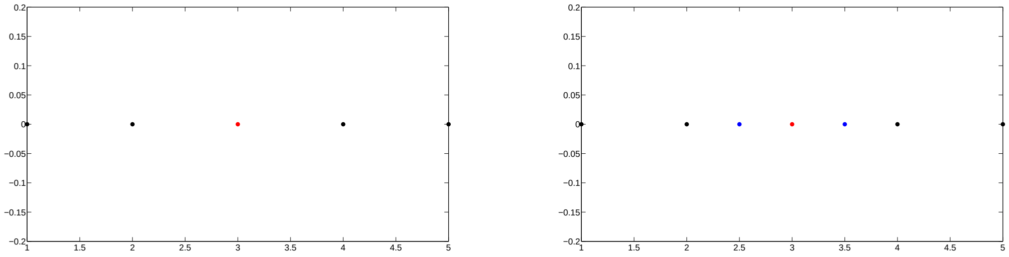

FIGURE 4.1: An example of refinement.

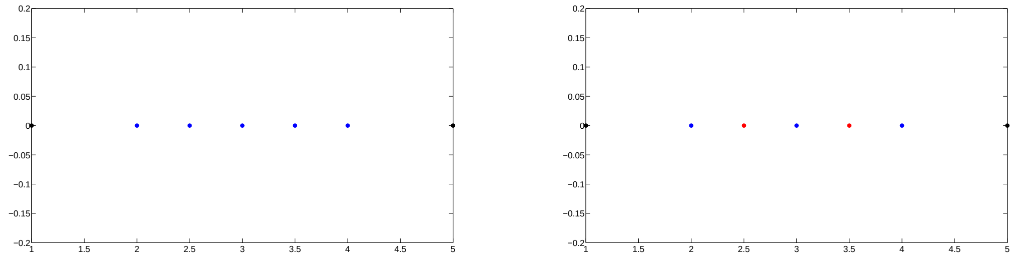

Figure 4.2: An example of coarsening.

coarsening nodes. In our coarsening strategy we remove 1 node from the left and 1 from the right, which is based on the given information. Firstly, we consider the left side of blue node at (2, 0). By removing this node, the distance between node at (1, 0) and node at (2.5, 0) is 1.5 which exceeds our maximum distance condition, leading to its automatic rejection. Now we look at the second blue node at (2.5, 0), this node satisfies our maximum distance condition, which means we can coarsen it and colour it with red (right side of Figure 4.2). On the right side, we start to check the maximum distance condition with the blue node at (4, 0), which does not satisfy our maximum distance condition. Therefore we move on to check node at (3.5, 0) and find it that fulfills our maximum distance condition. Right panel of Figure 4.2, we have coloured the coarsening nodes with red colour. Within the 1-D problem, we know the error occurs around the strike region which is due to the non-smooth initial conditions, thus we divide our node set into 2 region: non-coarsen-or-non-refine region and coarsen-refine region. In Subroutine 1, it redistributes the nodes with our non-coarsen-or-refine region with predefined maximum distance, and it keeps extra nodes at the boundary. The redistribution rule is based on removing nodes in non-coarsen-or-refine regions with the maximum distance condition. In the coarsen-or-refine region, we can summarise our coarsening or refinement strategy as follows:

Let err_{ref} and err_{crs} be the error thresholds for refinement and coarsening, respectively. The refinement and coarsening strategy can be summarized as follows:
For a given set of interpolation points $x_1, \ldots, x_N$, we will not change the two end points (first and last interpolation points).

1. - For each x_i, determine $\mathcal{N}_{x_i}$;
   - Compute $\tilde{u}_{\mathcal{N}_{x_i}}$, the local interpolant using cubic splines;
   - Compute $r_{\text{err}} = \mid u_N(x_i) - \tilde{u}_{\mathcal{N}_{x_i}}(x_i) \mid$.

2. Refine if $r_{\text{err}} > err_{\text{ref}}$.

3. Coarsen if $r_{\mathrm{err}} < err_{\mathrm{crs}}$.

Now, we will explain and discuss our adaptive algorithm for the Black-Scholes PDE. There are a few differences in comparison with the equally spaced RBF method. Let us name our main algorithm as the main function, and name the refinement and coarsening functions 1 and 2 as our subroutines 1 and 2. Our main Algorithm 1 can be described as follows.

Algorithm 1 Algorithm for adaptive radial basis function - one dimensional non-dividend payment Black-Scholes equation

```
1. Input the parameters (included $err_{\mathrm{ref}}$, $err_{\mathrm{crs}}$) and coarsen or refine region,
node distribution, maximum distance.

2. Choose a basis function and construct the node set.

3. Call **Refinement and coarsening function 1** (output - node distribution,
adaptive shape parameter and approximation of solution, $u$).

4. Boundary update.

5. call **Extra time steps** ($i = 0$ to $T_1$).

**for** $i = T_1$ to $T$ **do**

    Compute the differentiation matrices.

    Compute the coefficient matrix with output of coarsen and refine functions
    and use the time integration scheme.

    Recall **Refinement and coarsening function 2**, compute new set of node
    distributions, adaptive shape parameter, approximation of solution, $u$.

    Boundary update

**end for**

Compute the value of option
```

In our main algorithm, apart from extra time steps and the refinement or coarsening strategy, the algorithm is more or less the same as the equally spaced RBF method, where the BUP system from Hon and Mao [53] is used, and the extra time step treatment is presented in Subroutine 3. In other words, the method is straight forward and easy to implement. Here (Subroutine 1 and Subroutine 2) are our refinement coarsening and function 1 and 2 which are the key parts of our adaptive method as they contains the error indicator and the adaptive shape parameter. Subroutine 1 is also an immediately coarsening and refinement strategy.

Subroutine 1: Refinement and coarsening function 1

```
1. Input $err_{\text{ref}}$ , $err_{\text{crs}}$ and coarsen or refine region, node distribution, maximum
distance.

2. Compute number of nodes.

3. Redistribute the nodes.

while ($r_{\text{err}} > err_{\text{ref}}$) do

    Compute adaptive shape parameter set.

    Compute the MQ basis function of $u$

    for $2 \leq i \leq N-1$ do
        a) For each $x_i$, determine $\mathcal{N}_{x_i}$

        b) Compute $\tilde{u}_{\mathcal{N}_{x_i}}$, the local interpolant using cubic splines;

        c) Compute $r_{err} =\mid u_N(x_i) - \tilde{u}_{\mathcal{N}_{x_i}}(x_i) \mid$.
    end for

    Refine if $r_{err} > err_{ref}$.

end while $r_{err} < err_{ref}$

Output value of $u$, $c$ and node distribution $(x)$.
```

In Subroutine 3, dt is the size of the time step that is used in the equally spaced method. However, in our adaptive method, we use a non-uniform time steps treatment to reduce the error caused by the non-smooth initial conditions. Therefore, we use the small time steps after Subroutine 1. Figure 4.3 is an example of our adaptive method in Subroutine 1. It starts with the left sub-figure, which is the uniform node distribution at $t = T$. The right sub-figure shows the redistribution of nodes after Subroutine 1. In Figure 4.4, the left sub-figure shows the new node distribution after Subroutine 3 (extra time steps) in Algorithm 1 (step 5). The right sub-figure shows the final node distribution at $t = 0$ (backward time integration applied). Figure 4.3 and Figure 4.4 also show that our adaptive method can capture the error profile as the node distribution follows the change of error profile in time (from $t = T$ to $t = 0$).

Subroutine 2: Refinement and coarsening function 2

```
1. Input $err_{\text{ref}}$ , $err_{\text{crs}}$ and coarsen or refine region, node distribution, maximum distance.

2. Compute number of nodes.

while ($r_{\text{err}} > err_{\text{ref}}$) do

    Compute adaptive shape parameter set.

    Compute the MQ-interpolant at u

    for $2 \leq i \leq N-1$ do
        a) For each $x_i$, determine $\mathcal{N}_{x_i}$

        b) Compute $\tilde{u}_{\mathcal{N}_{x_i}}$, the local interpolant using cubic splines;

        c) Compute $r_{err} =\mid u_N(x_i) - \tilde{u}_{\mathcal{N}_{x_i}}(x_i) \mid$.
    end for

    Refine if $r_{err} > err_{ref}$.

    Coarsen if $r_{err} < err_{crs}$.
end while $r_{err} < err_{ref}$

Output value of $u$, $c$ and node distribution $(x)$.
```

Subroutine 3: Extra time steps

```
1. Use small size of time steps, e.g. $\frac{dt}{5}$.

2. Switch back to $dt$ for the remaining number of time steps, such as $(dt)$.

3. Use larger size of time steps (e.g. $2dt$) if we need reduce the computational cost.
```

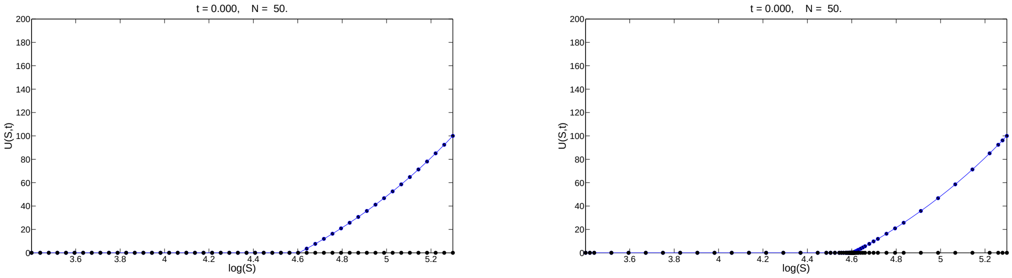

Figure 4.3: An example of Subroutine 1 for a European Call option.

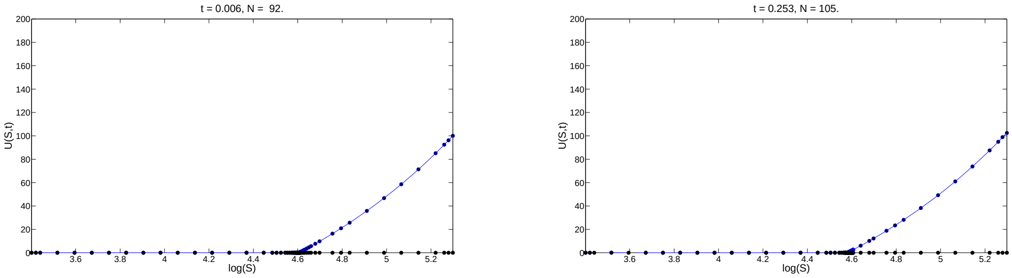

Figure 4.4: An example of adaptive RBF method.

4.2.2 Error indicator

Adaptive methods for PDE solving based on error indicators are an active area of research in both finite elements and RBFs [1, 28, 5, 63, 92]. Iske et al. and Driscoll et al. pointed that the finer grids should be placed in a high variation region [59, 60, 28, 5]. We used a simple and straight forward method (for the reason of computation cost and complicated data structure, we should keep the adaptation rule to be as simple as possible in term of efficiency) for our error indicator, which finds the pointwise error $r_{\text{err}} =\mid u_N(x_i) - \tilde{u}_{\mathcal{N}_{x_i}}(x_i) \mid$ for all points excluding the first and last points. x is a set of points, u_N is the approximation solution of u produced by global MQ basis functions, $u_{\mathcal{N}_{x_i}}$ is the local approximation of u_N by using cubic spline with closest 8 nodes. In the 2-D case, we use the Voronoi points which we present in Chapter 5. The predefined thresholds err_{ref} and err_{crs} are used to compare with our residual r_{err}.

4.2.3 The adaptive shape parameter

The crucial part of adaptivity is the shape parameter, c, which has been noted in [28]. The c in our adaptive method is a center dependent shape parameter, unlike in the equally spaced method as we have redefined our MQ basis function in Equation (4.3). In our method, we apply the adjustment of shape parameter that gives to each center an individual shape parameter where it depends on the distance to the nearest neighbour. Numerical result shows that the adjustment of the shape parameter set performs well as the shape parameter affects both the accuracy of the result and condition number. In our algorithm, the adaptive shape parameter is automatically computed during the refinement and coarsening strategy. For each individual node, its own shape parameter is produced by selecting the minimum distance between two arrays, such as, $\min\{[\infty, \frac{1}{dx}], [\frac{1}{dx}, \infty]\}$, where dx is the distance between 2 nodes, and since we need to restrict the problem to a finite domain. The reason of including ∞ is to show that the first and last nodes only have one close node thus we introduce one extra node where it has distance of ∞. Therefore we assigned first and last nodes with the shape parameter of ∞. In Chapter 2, we know that the condition number $k(A)$ of our interpolation matrix A depends on the maximum and minimum of the eigenvalue λ. The low condition number means the matrix is well conditioned, otherwise it is ill-conditioned. For example in Equation (2.12), for the linear system problem, if A is an ill-conditioned matrix (large condition number), the error in solution f is much larger than the error in λ. In other words, the error rate of λ depends on the error change of f. Therefore, increasing the value of minimum eigenvalue $\lambda_{\min}$ could be one potential way to construct a well conditioned matrix as it helps to prevent the increment of the condition number. Figure 4.5 shows the log-log plot of RMSE and the condition number vs shape parameter c. The left panel of Figure 4.5 shows the RMSE against c (all of the constant shape values come from our adaptive shape parameters due to the non-equally node set), and the right panel gives the log-log plot of the condition number against all of the constant shape parameters. Both figures show that our adaptive shape parameter can have a good condition number and smaller RMSE when compared with the constant shape parameter. Also we need to point out that to find the optimal constant shape parameter is expensive and inefficient for the non-uniform node set problem. In this experiment, as we used the MQ from Table 2.1, thus the shape parameter is produced by selecting the minimum distance between an array $4[\infty, dx]$ and an array $4[dx, \infty]$.

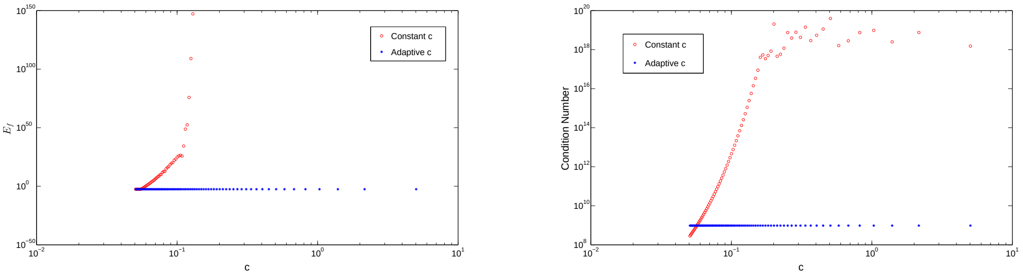

FIGURE 4.5: RMSE and condition number plot for the adaptive (the center dependent shape parameters) and constant shape parameter c.

4.2.4 Time integration scheme

In our adaptive algorithm, we use the backward technique with the CN scheme which can be found in [80, 53]. Moreover, we also use non-uniform time steps for our adaptive method since we know that a peak exists at the initial condition. To reduce the size of the peak we require finer grids in the region of these kinks at the initial conditions and small time steps. Within this method, we require extra time steps at first iteration time by comparing with uniform time step, and may also need some extra coarsening and refinement strategies in these extra time steps.

4.3 Numerical Results

In the following numerical experiments, we introduce an additional parameter set given in Table 4.1 which we call Parameter Set 2 (or the challenge parameter set [97]). As we can see, the σ is relatively small, the interest rate r is large and small value of time to maturity T in comparison with Parameter Set 1 (Table 3.2) which we still follow in our numerical simulations. To avoid confusion, all Parameter Set 1 results are in Appendix B. The three evaluation points are 98, 99 and 100 for a European call option and 97, 98, 99 for an American put option and a barrier up and out call option, which is much closer to the strike price (K) if we compare it with Parameter Set 1.

Parameter Values	
σ	0.01
r	0.10
T	0.25
K	100
B	125

Table 4.1: BENCHOP project, Problem 1 in [97], Parameter Set 2, B is the barrier level for the barrier up and out option.

The error measures used in previous chapters will be applied in the following numerical simulations. Our ARBF algorithm for numerically solving Equation (4.2) with initial condition Equation (3.5) is demonstrated with a European call option in Figure 4.6, the parameters we use are in Table 2 [80]. The Parameter Set 1 is $T = 1, K = 15, \sigma = 0.3, S = 15$, and $r = 0.05$, and in here an example of large time steps is demonstrated.

In Figure 4.6, the red nodes represent the value of RBFs interpolant in the interpolation points, the blue line represents the call payoff function with 301 equally spaced nodes and the blue nodes represent the value of RBFs interpolant in the interpolation point which lie on the blue line; this means that the RBFs interpolation scheme is implemented properly. Figure 4.6 shows that ARBF-MQ with non-uniform time step, as seen in the algorithm. We use 20 uniform nodes and 30 equally time step to start with, after the immediately refine and coarsen strategy, we have 22 nodes which are placed non-uniformly. After step 1, we use smaller time step ($\frac{dt}{5}$) and the number of nodes increased (23) with a different position to the pervious step. In step 3, we use $2dt$ which is 15 steps in total, the nodes has been reduced to 20. Overall, we used 20 time steps and our relative error

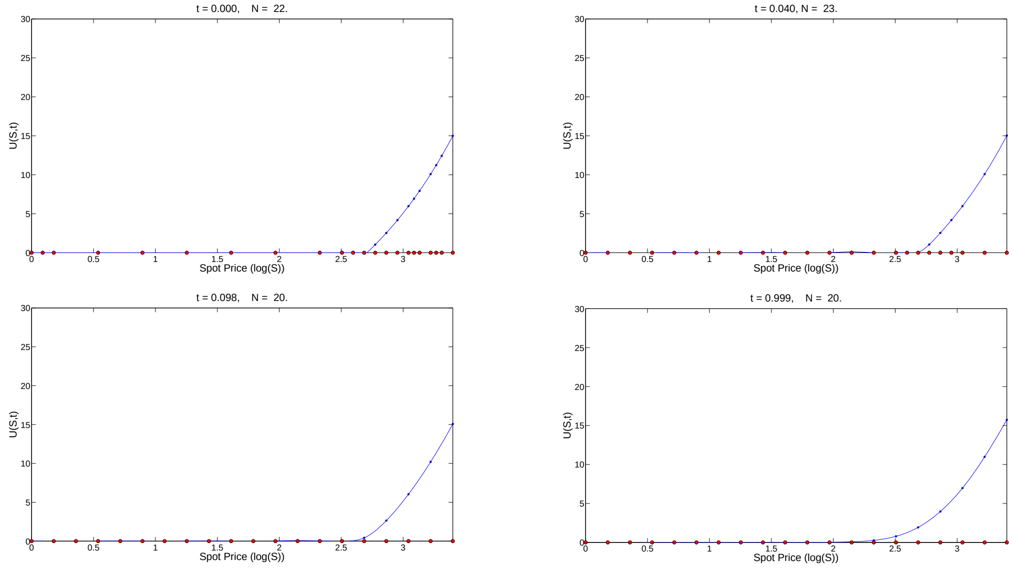

FIGURE 4.6: Example of ARBF-MQ with non-uniform time step ($\frac{dt}{5}, 2dt$).

is 7.5647e-05 which is 9 times better than equally spaced RBF method ($E_{\text{rel}} =$ 7.10e-04, 80 equally spaced nodes and 80 time steps). This example shows our ARBF-MQ can achieve better interpolation result with fewer required of nodes. Also, we note that the position of nodes did not change after step 3, this is because the error is below our error indicator and for computational efficiency no refinement and coarsen strategy is required. For simplicity, we do not increase the size of time step after step 2 in the following experiments in this Chapter.

4.3.1 European call option

In our numerical simulations, given a set of interpolation points $x_1, \ldots, x_N$, we denote our maximum and minimum logarithms price as $x_{\max} = \log(S_{\max})$ and $x_{\min} = \log(S_{\min})$, $x \in [\log(30), \log(2K)]$. Figure B.1 shows the absolute error surface of a European call option for ARBF-MQ in time for Parameter Set 1 (Table 3.2). Note that the maximum value of E_{abs} in initial step is 3.136e-03 which is 78 times better when compared with the equally spaced method in Figure A.1 (2.452e-01). This indicates that our ARBF-MQ can reduce the error which is caused by the non-smooth initial payoff function. When $M = 0$ an immediate refinement and coarsen strategy is applied, the number of nodes is reduced from 160 to 99. In the first step of "small time steps" in step 2 (Subroutine 3), we add an extra node. After that we used 101 nodes for the remaining time steps. The total number of time steps in our method is 205. In comparison with equally spaced RBF method, we used extra 45 time steps, but with a total of $\frac{2}{3}$ of nodes required. The relative error of a European call option in Parameter Set 1 at $t = 0$ (option value at today with maturity time T) is shown in Figure B.3. It shows that ARBF-MQ can achieve higher accuracy than the equally spaced RBF method (Figure 3.3) as the maximum relative error (E_{rel}) for a European call is 9 times less. Table 4.2 shows the maximum condition number and RMSE, overall, the ARBF-MQ has small condition number and smaller error in E_f.

Error	ARBF-MQ	MQ (160)
E_f	2.8770e-04	1.15e-02
Max Condition Number	2.0475e+07	6.0362e+07

Table 4.2: The RMSE of a European call option for both methods in Parameter Set 2.

In Parameter Set 2, we note that when $\sigma = 0.01$ our Equation (4.2) tends to be a singularly perturbed PDE and if $\frac{1}{2}\sigma^2$ approaches zero then the solutions or derivatives approach a discontinuous limit [88]. To solve this problem and achieve a good approximation solution, we require a large number of uniform nodes in the equally spaced method. Figure 4.7 shows the equally spaced RBF method with 160 nodes cannot approximate the solution well and oscillations appear throughout time. Next, we increased the number of nodes to 1500 uniform nodes in Figure 4.8, and we can see that the absolute error decreased significantly with the maximum absolute error at 6.19e-03 caused by initial non-smooth payoff function at time step 1. Figure 4.9 shows that our adaptive method can reduce the maximum absolute error in Figure 4.8 from 6.19e-03 to 7.788e-05 with 10 time fewer nodes (maximum of 150 nodes which is $\frac{1}{10}$). The reason why our adaptive method works is because our adaptive method has a similar strategy as Shishkin meshes (details can be found in [94]), where we place a large number number of nodes at the peak region to reduce the error, where as the equally spaced method uses the same amount of nodes in whole region. Figure 4.10 shows we have small absolute error at $t = 0$. Figure B.8 shows the absolute error of a European call option with our adaptive method for the uniform time steps, we can see that it also has the better accuracy than the equally spaced method (1500 nodes). Node distribution in time is produced in Figure B.9 for both adaptive and non-adaptive with Parameter Set 2. Figure 4.10 represents the absolute error of a European call option value at current time with maturity time T. As we can see in Figure 4.10, our adaptive has higher accuracy than the equally spaced method (1500 nodes) apart from the spot price at 96 and 97. Figure 4.11 shows the node distributions of adaptive RBF interpolation in time. We can see that the nodes slowly increase and the maximum of nodes we used is 150 nodes.

Error	ARBF-MQ (150)	MQ (1500 nodes)
E_{ref}	6.64075e-05	2.42706e-03
Max Condition Number	3.7275e+08	6.0362e+07
CPU time	14.52102	25.06761

TABLE 4.3: Maximum E_{ref}, condition number and CPU time of a European call option for both methods in Parameter Set 2.

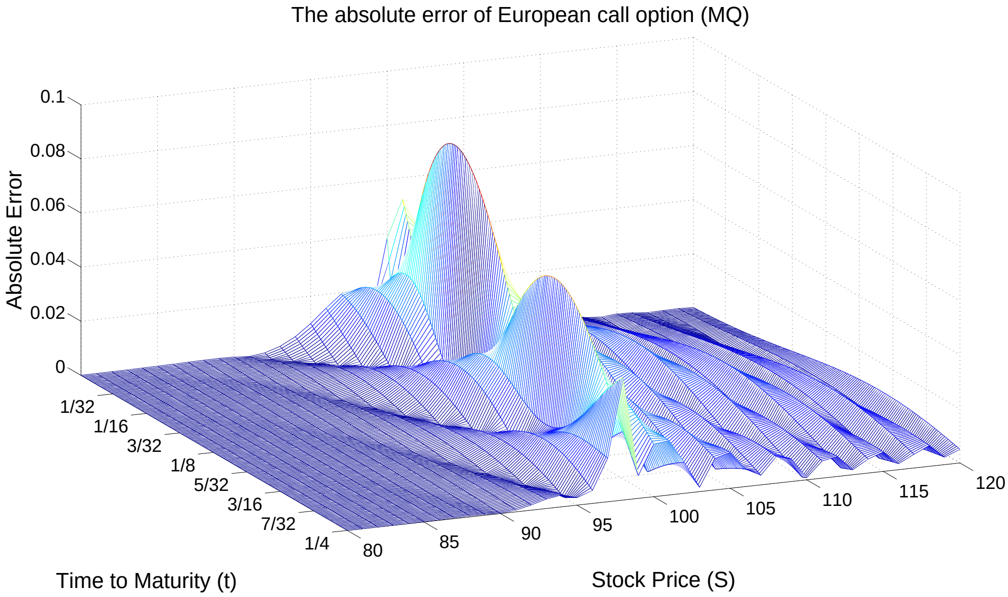

FIGURE 4.7: Surface of absolute error of a European call option in Parameter Set 2 with uniform nodes (160).

Table 4.3 shows the maximum relative error E_{ref} at three evaluation points (98, 99 and 100), condition number and CPU time. Though from the condition number in our adaptive method is a bit larger than the equally spaced method, we have better accuracy and less CPU time.

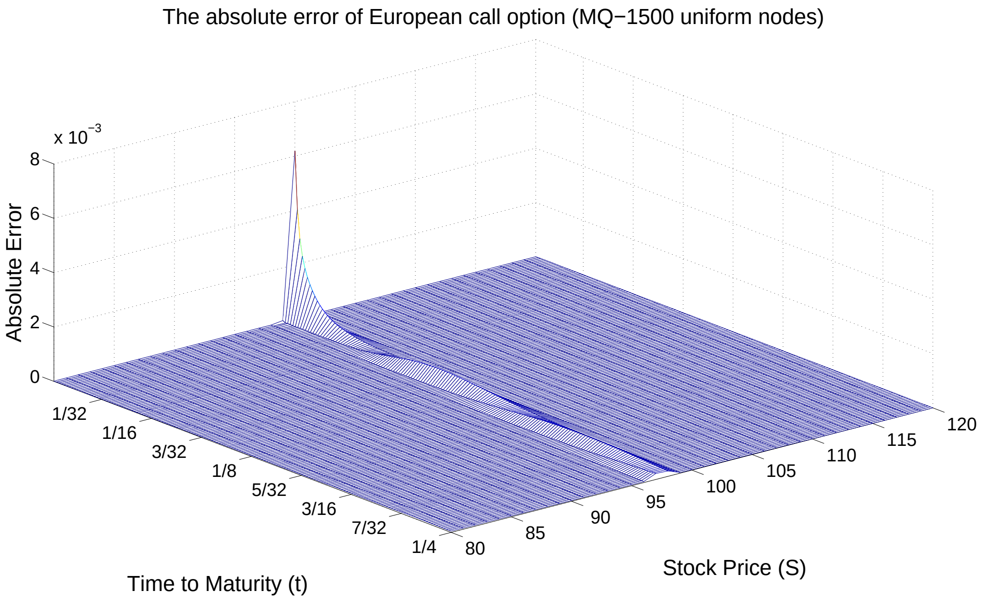

FIGURE 4.8: Surface of absolute error of a European call option in Parameter Set 2 with non-adaptive MQ (1500) method.

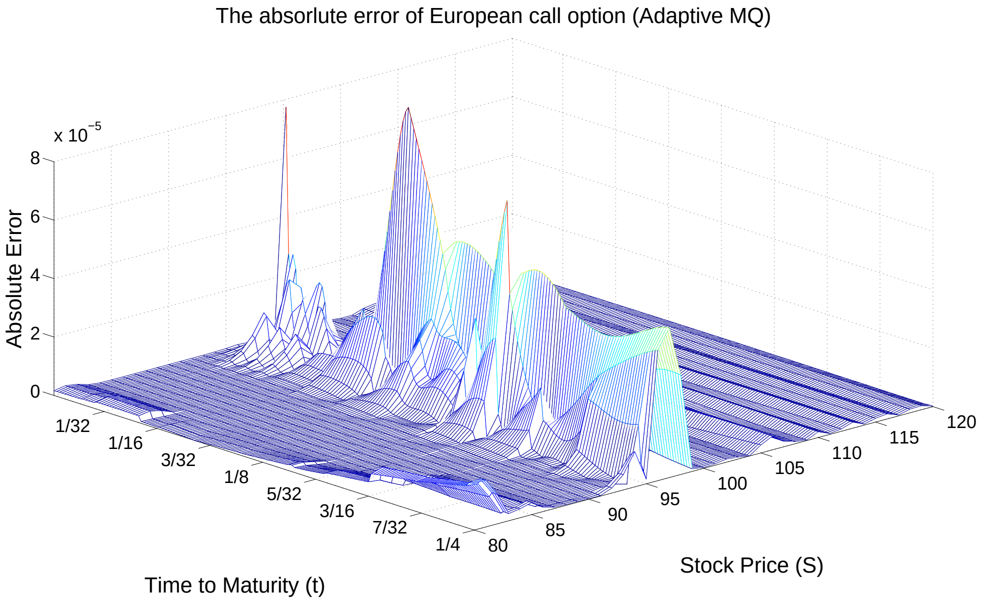

FIGURE 4.9: Profile of absolute error of a European call option in Parameter Set 2 with ARBF-MQ method.

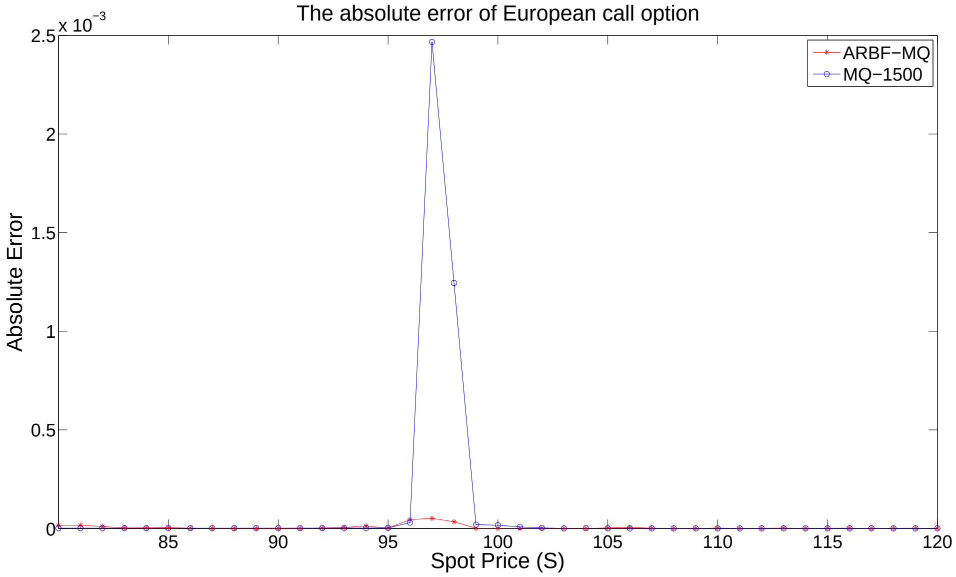

FIGURE 4.10: Absolute error of a European call option for both adaptive and non-adaptive MQ methods at $t = 0$.

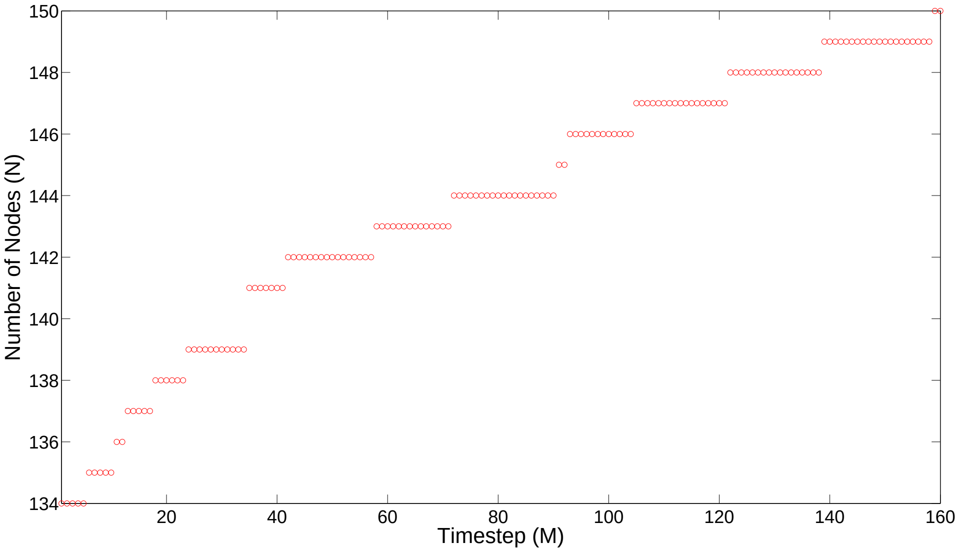

FIGURE 4.11: ARBF-MQ node distribution in time for a European call option in Parameter Set 2.

4.3.2 Greeks

In this subsection, we discuss the first and second partial derivatives of U with respect to the underlying asset S. One of the advantage of using the RBF method is that the partial derivatives (Δ and Γ) can be directly obtained by differentiation. Firstly, we present our results for Parameter Set 1 in comparison with the results we had in Chapter 3 (the equally spaced method). From Figure B.4 and Figure B.6 we can say that the absolute error of the Delta and Gamma at initial time step have been successfully reduced. The relative error of Delta value (the current value of Delta with maturity T) is given in Figure B.5, and shows that E_{rel} is approximately 5.2e-04 which is better than the MQ we had in the equally spaced method where E_{rel} around 6.6e-04 (Figure 3.4). Figure B.7 shows E_{rel} of Gamma value(at current value with maturity T) against spot price. In comparison with Figure 3.6, our adaptive method has 10 times smaller relative error since our maximum of E_{rel} is of approximately 1.2e-02. Figure 4.12 and Figure 4.15 show the absolute error of Delta and Gamma in the equally spaced method. We can see that a kink appears in the first time step for both Delta and Gamma. Figure 4.13 and Figure 4.16 show our adaptive method has reduced the absolute error of Delta (around 10 times smaller) and produced 14 times more accurate solution in Gamma in comparison with the equally spaced method. The absolute error of Delta and Gamma values at today with maturity T is produced in Figure 4.14 and Figure 4.17.

Table 4.4 shows the RMSE of Delta and Gamma value at today with Maturity T. From this table we note that the RMSE of our Delta in the adaptive method is much better the equally spaced method. The same is true for Gamma. Next we present our Vega in Table 4.5. In this simulation, the three evaluation points are 97, 98 and 99 in Parameter Set 2, the maximum number of nodes we used is 130, we can see that our adaptive method produced a smaller relative error and CPU time when compared with the equally spaced method.

	Delta		Gamma	
Error	ARBF-MQ (150)	MQ (1500 nodes)	ARBF-MQ (150)	MQ (1500 nodes)
E_f	6.19805e-06	1.20113e-03	3.29233e-04	7.64816e-03

Table 4.4: The maximum RMSE of Greeks for both methods in Parameter Set 2.

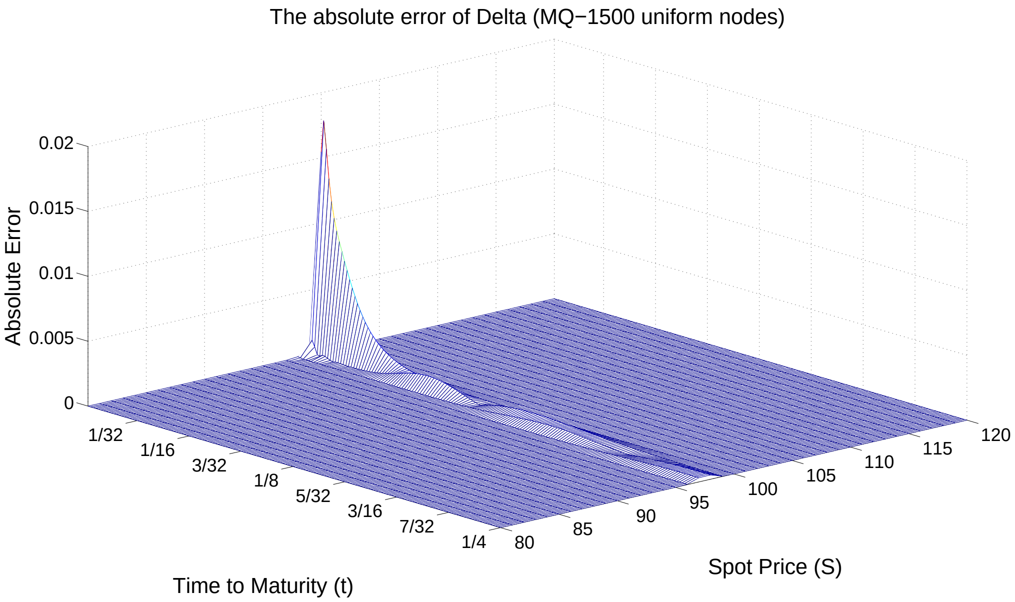

FIGURE 4.12: Profile of absolute error of Delta in Parameter Set 2 with non-adaptive MQ (1500) method.

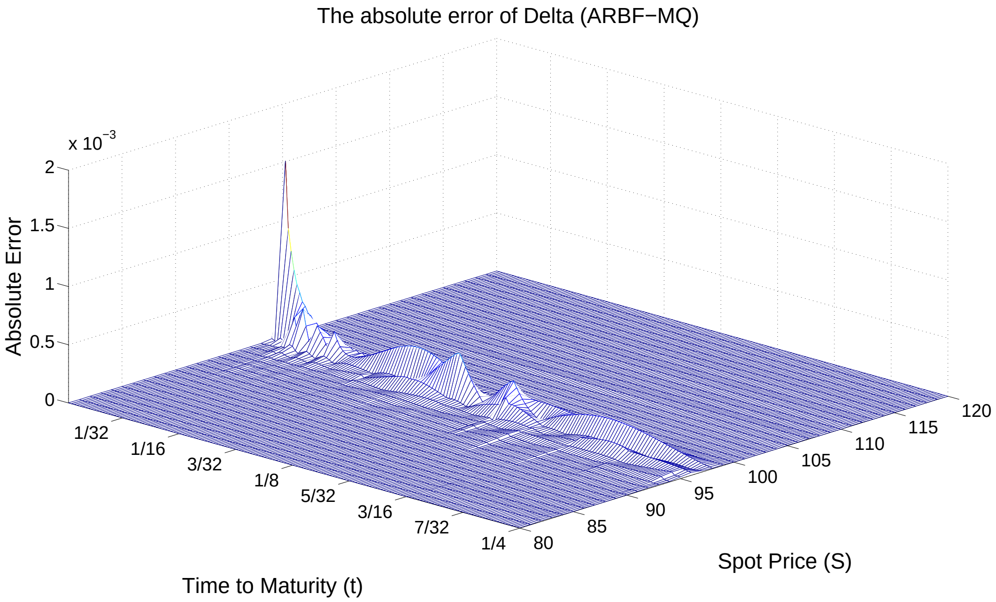

FIGURE 4.13: Profile of absolute error of Delta in Parameter Set 2 with ARBF-MQ method.

	$S = 97$	$S = 98$	$S = 99$	CPU time
MQ (1500)	6.9e-03	2.8e-03	1.4e-03	49.994
ARBF-MQ (130)	1.18e-06	9.26e-06	1.09e-05	3.652

TABLE 4.5: The relative error E_{ref} of Vega for Parameter Set 2.

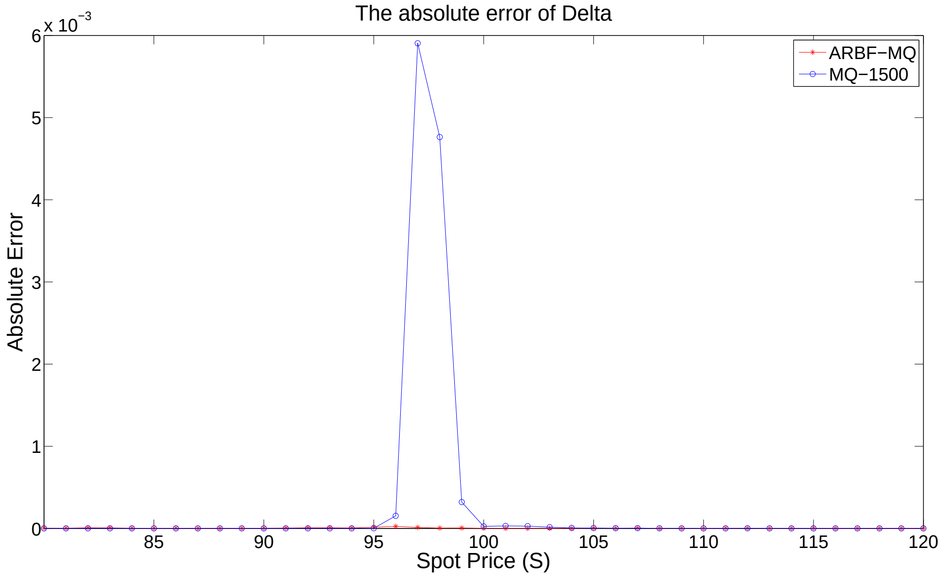

FIGURE 4.14: Absolute error of Delta for both adaptive and non-adaptive MQ methods at $t = 0$ in Parameter Set 2.

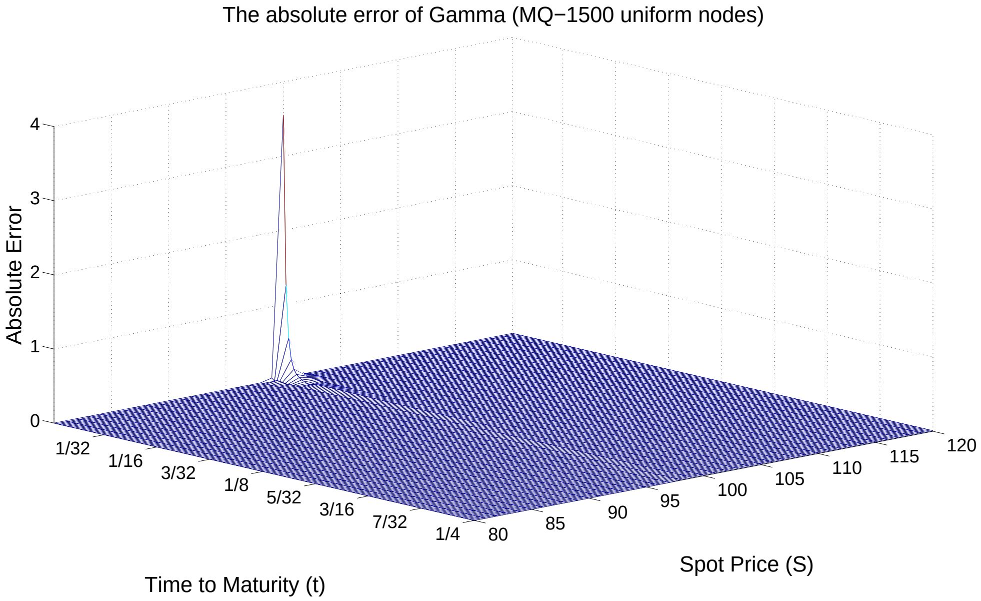

FIGURE 4.15: Profile of absolute Error of Gamma in Parameter Set 2 with non-adaptive MQ (1500) method.

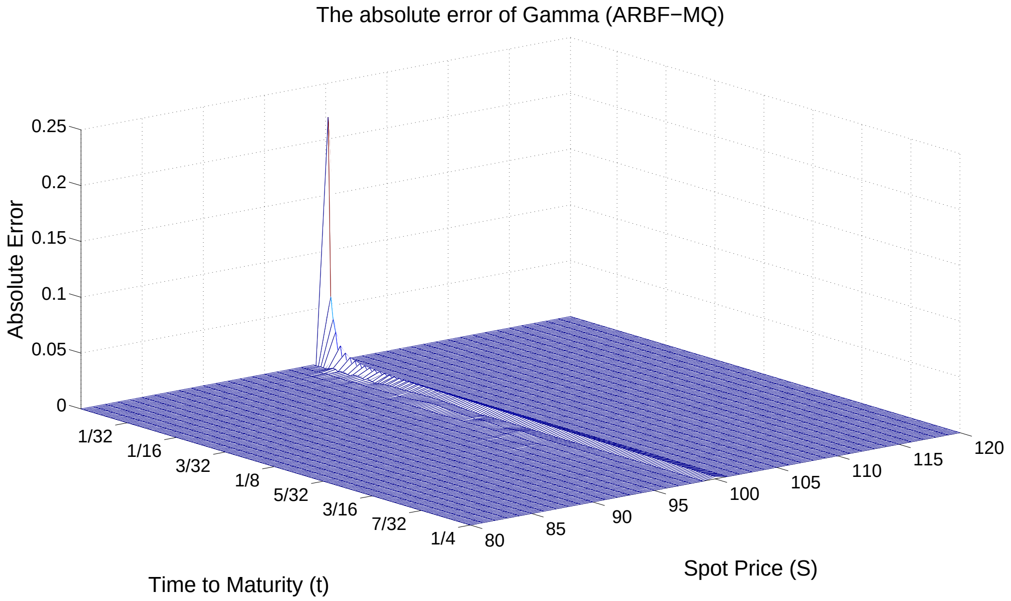

FIGURE 4.16: Profile of absolute error of Gamma in Parameter Set 2 with ARBF-MQ.

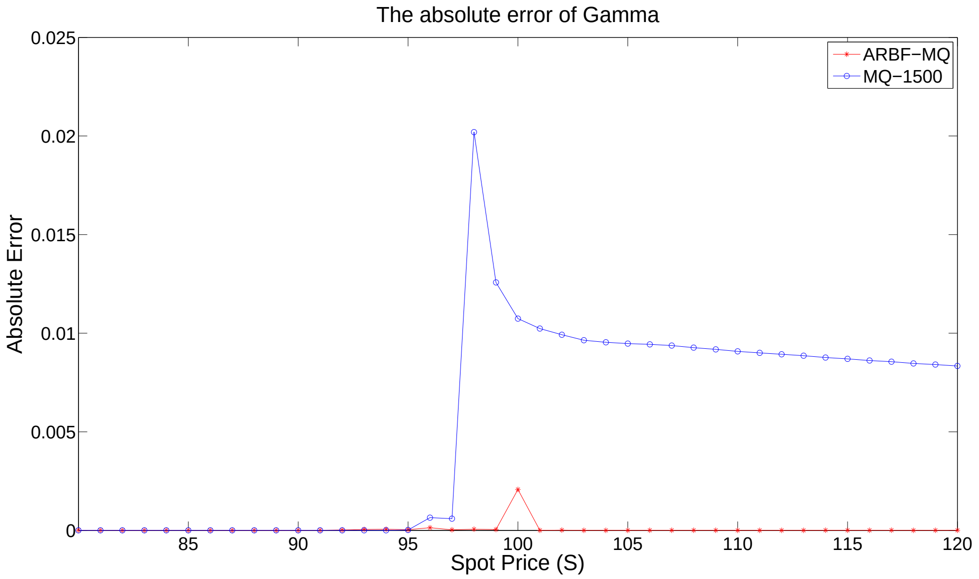

FIGURE 4.17: Absolute error of Gamma for both adaptive and non-adaptive MQ methods at $t = 0$ in Parameter Set 2.

4.3.3 American Put Option

In the American option, we used the same treatment as we did in the equally spaced method, where the Boundary Update Procedure given in [53] is used. Now we used Parameter Set 1 and 2 to test an American put option. Our reference values for an American put option at spot price 97, 98 and 99 are $U = [3, 2, 1]$ in Parameter Set 2 and $U = [10.7264867100, 4.8206081848, 1.8282075840]$ for Parameter Set 1 where evaluation points are 90, 100 and 110. Table 4.6 shows the maximum relative error at the three evaluation points and the CPU time for Parameter Set 2 . We can see that our adaptive method can achieve higher accuracy and requires a small number of nodes. In the equally spaced method by delivering such accuracy, it used 3000 nodes and a total number of 205 time steps; this costs 34 time more CPU time in comparison with adaptive method and requires 22 times more nodes.

	MQ(3000)	ARBF-MQ (135)
E_{ref}	1.53e-05	1.17e-05
CPU time	206.94012	6.06531

Table 4.6: The maximum relative error (E_{rel}) of approximation solution of an American put option and CPU time, Parameter Set 2.

4.3.4 Barrier up and out call option

Figure B.10 shows the surface of absolute error of a barrier up and out call option for spot price from 80 to 120. In comparison with Figure A.7 and Figure A.8, we can see that the absolute error at the initial condition has been reduced as well as the spot price (120) which is close to barrier level (125). Figure B.11 represents the relative error of a barrier call option at spot price between 80 and 120 at today with maturity time T for Parameter Set 1. The relative error is less than 4.6e-03, and for $S = 120$ and $S = K$ our adaptive method can perform nearly 50 and 2 times better than the equally spaced method in comparison with Figure 3.7. Figure 4.18 gives the node distribution in each time step for Parameter Set 1. We can see that the maximum number of nodes we used is 67. In Table 4.7, we present our result of maximum relative error in these three predefined evaluation points for Parameter Set 2.

	MQ(3000)	ARBF-MQ (101)
E_{ref}	1.30e-02	9.98e-05
CPU time	210.290490	7.422248

TABLE 4.7: The maximum relative error (E_{rel}) of approximation solution of a barrier up and out call option and CPU time for Parameter Set 2.

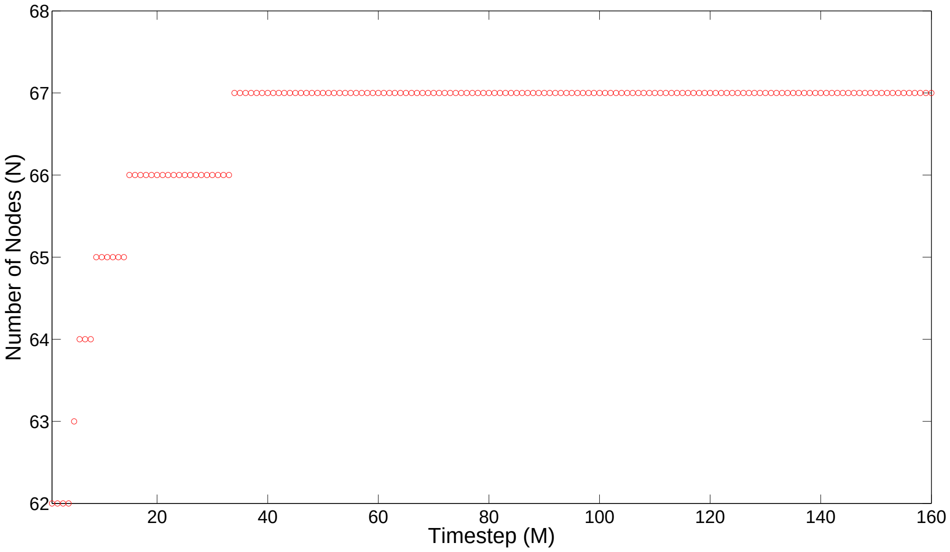

FIGURE 4.18: ARBF-MQ node distribution in time for a barrier up and out call option in Parameter Set 1.

4.4 Conclusion

In this chapter, we have compared our adaptive method with the equally spaced method for a number of variate options. For the European option with Parameter Set 2, our adaptive method has higher accuracy (maximum relative error less than 1e-05) in terms of maximum relative error at 3 evaluation points and less CPU cost than the equally spaced method. We also note that the equally spaced method requires at least 10 times more nodes to deliver a solution with relative error less than 2.5e-3. Moreover, for Delta and Gamma, the equally spaced method can only deliver a maximum relative error around 8e-03, but our adaptive method can deliver a maximum relative error less than 4e-04 which is about 20 times more accurate. Parameter Set 2, we require $\frac{1}{22}$ of nodes and $\frac{1}{34}$ of CPU time in contrast to 3000 nodes for the equally spaced method to obtain similar levels of accuracy. Finally in barrier up and out call option with Parameter Set 2, we gain both high accuracy and less CPU time in comparison with the equally spaced method (3000) where our adaptive method requires 101 nodes. Additionally in [97], to deliver a solution with a relative error less than 1e-5 for 3 evaluation points in Parameter Set 2, we have showed that our adaptive method is faster than FD method in Table 3 and 4 for some financial contracts. The numerical results are based on the Tintin cluster, which consists of 160 dual AMD Opteron 6220 (Bulldozer) nodes and locates at Uppsala Multidisciplinary Center for Advanced Computational Science (UPPMAX). Overall, our adaptive method can reduce the error caused by a non-smooth payoff function, and it is useful for a problem (due to singularly perturbed problem) which the equally spaced method cannot solve properly.

Chapter 5

Adaptive Radial Basis Function for Spread Options

5.1 Multi-asset Black-Scholes Equation

Nowadays, computers have become more and more powerful than ever before, but sometimes they are still not able to perform simulations for high dimensional problems as the problem size grows exponentially with the number of dimensions, which is also limited by the memory of computers. However, in finance, the derivative products have become more and more complicated than ever before and with a huge interest of demand in multi-asset products where the price of an option is based on d underlying assets, such as an European style of basket option and a spread option. To price an European style of spread option through solving the 2-dimensional Black-Scholes equation is still active research as to today there are no analytical solutions. Currently, there are some methods to price multi-dimensional contracts, such as Monte Carlo methods [46], sparse grid methods [13, 44, 43], and finite difference methods. In high dimensional problems, computational time is another key point for the financial market. In this chapter, we introduce the equally spaced and adaptive methods to solve 2 dimensional Black-Scholes PDE for a special case of a spread call option ($K = 0$ which is a Margrabe call option). The non-dividend payment of d-dimensional Black-Scholes problem is defined as

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d}\rho_{ij}\sigma_i\sigma_j S_i S_j \frac{\partial^2 V}{\partial S_i S_j} + \sum_{i=1}^{d} r S_i \frac{\partial V}{\partial S_i} - rV = 0, \qquad (5.1)$$

where d is the number of assets whose price at time t is denoted by $S(t) = (S_1(t), \dots, S_d(t))$, r is the risk free interest rate, V is the price of option, σ_i denotes the volatility of i-th underlying asset and ρ_{ij} is the correlation between asset i and j.

5.2 Equally space radial basis function for a Margrabe call option

Margrabe was the first person to treat each asset of spread option separately who also derived an analytical solution for the special case ($K = 0$) of spread option, this also known as Margrabe option [74]. In this section, we will implement the equally spaced RBFs method for a Margrabe option; the parameter set was taken from the problem (6) of the BENCHOP project [97] and we refer to it as Parameter Set 3:

Parameter Values	
σ_1	0.15
σ_2	0.15
r	0.03
T	1
K	0
ρ	0.5

Table 5.1: BENCHOP project, Problem 6 in [97], Parameter Set 3

The non-dividend payment for 2 assets, Black-Scholes equation is

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sum_{i=1}^{2}\sum_{j=1}^{2}\rho_{ij}\sigma_i\sigma_j S_i S_j \frac{\partial^2 V}{\partial S_i S_j} + \sum_{i=1}^{2} r S_i \frac{\partial V}{\partial S_i} - rV = 0, \tag{5.2}$$

where V is the price of option, S_1 and S_2 are the underlying asset 1 and 2, σ_1 and σ_2 represent the volatility of asset 1 and 2, r is the risk free interest rate, the correlation between asset 1 and 2 is ρ_{12} or ρ_{21} ($\rho_{12} = \rho_{21}$). The payoff function is

$$V(S_1, S_2) = \max(S_1 - S_2 - K, 0). \tag{5.3}$$

The equally spaced RBF method for a Margrabe call option is more or less the same as for the 1-D European call option, the basis functions we used are in Table 3.1.

In our simulations, we propose a variable change, such as $U(s_1, s_2) = V(S_1, S_2)$, $s_1 = \log(S_1)$ and $s_2 = \log(S_2)$. The Equation (5.2) can be rewritten as

$$\frac{\partial U}{\partial t} + \frac{1}{2}\left(2\rho_{12}\sigma_1\sigma_2\frac{\partial^2 U}{\partial s_1 s_2} + \sigma_1^2\frac{\partial^2 U}{\partial s_1^2} + \sigma_2^2\frac{\partial^2 U}{\partial s_2^2}\right) + r\left(\frac{\partial U}{\partial s_1} + \frac{\partial U}{\partial s_2}\right) - rU = 0. \quad (5.4)$$

The initial condition is

$$U(s_1, s_2, T) = \max\{e^{s_1} - e^{s_2} - K, 0\}. \quad (5.5)$$

The boundary conditions depend on the spatial domain we have taken for each asset. The approximation solution of U is defined as

$$u(s,t) = \sum_{j=1}^{N} \lambda_j(t)\phi(\|s - x_j\|) \simeq U(s,t), \quad s = (s_1, s_2) \quad (5.6)$$

Now, the eight steps "cooking recipe" in [80] can be applied to solve Equation (5.4).

5.2.1 Numerical simulations

Figure 5.1 shows the profile at $t = 0$ of value of the Margrabe call option by using Parameter Set 3, the MQ meshfree approximation based on 40×40 nodes with CN method in time and 30 time steps. The left axis has been labeled as S_1, which represents the spot value of S_1. The right side of axis denotes the spot price of S_2. Because the correlation value of two assets is 0.5 and the payoff function is $S_1 - S_2$, which means that the option value increases just after $S_1 > S_2$. It also shows that the largest error occurs around the left side of diagonal line. In other words, we would expect our adaptive methods to place extra nodes at the right side of the diagonal line.

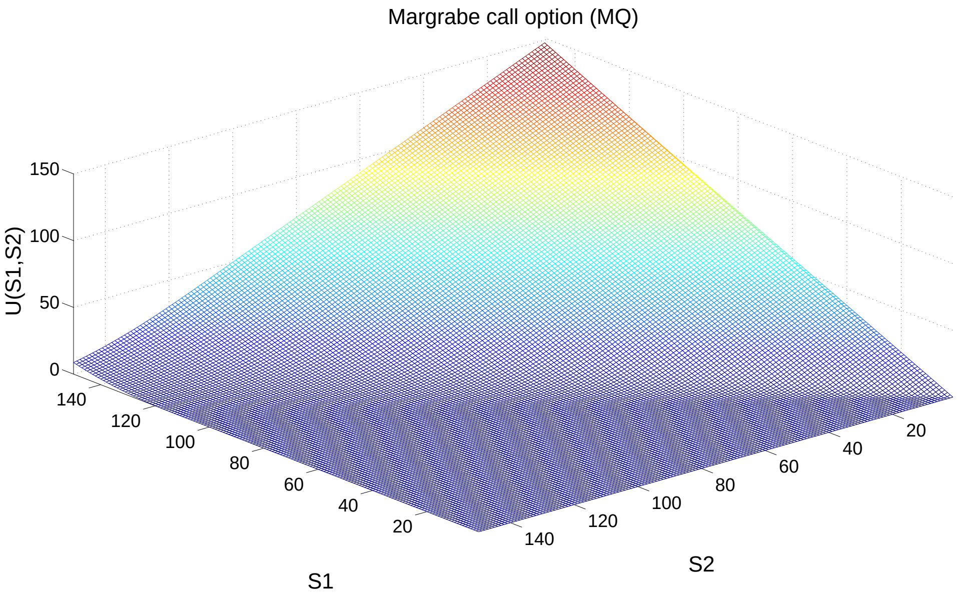

Figure 5.1: Profile at $t = 0$ Margrabe call option - MQ

5.3 Adaptive radial basis function for a Margrabe call option

Iske et al. [5, 6, 59, 60] have applied the adaptive method to many different 2-D problems, such as 2-D Burger's equation, and the slotted cylinder problem. Naqvi [77] has successfully applied the adaptive radial basis function to solve a Franke type function in her thesis. The Voronoi diagram has been used in coarsening and refinement strategy in their work, and we will follow closely the strategy in [6, 77]. In our 2-D non-dividend payment Black-Scholes problem, we present our latest development which only contains refinement. Before going any further, we introduce some relevant and necessary definitions need for our 2-D Algorithm. We start with the Voronoi diagram,

Definition 5.1 (Voronoi diagram)**.** For a fixed node set $X \subset \mathbb{R}^d$ and any $y \in X$, the Voronoi tile

$$V_X(y) = \{x \in \mathbb{R}^d : d_X(x) = \|x - y\|\} \subset \mathbb{R}^d \tag{5.7}$$

of y w.r.t X contain all points in $\mathbb{R}^d$ is x. And the set of $(V_X(x))_{x \in X}$ is called the Voronoi diagram of X.

Voronoi points are vertices of Voronoi tile $V_X(x)$ and $V_X(x)$ is a convex polyhedron which is non-empty. More details of Voronoi diagrams can be found in [85]. In our local interpolation approximation, we use thin plate splines which was proposed in [6, 59], and it is the special case of polyharmonic splines in 2-D. Duchon [31] gave the general framework of thin plate spline in 1977. The thin plate spline and polyharmonic spline basis functions can be found in Table 2.1. We use the following general form for our local interpolation approximation

$$s(x) = \sum_{j=1}^{N} \lambda_j \phi_k(\|\mathbf{x} - \mathbf{x}_j\|) + \sum_{|\alpha| \leq k} d_\alpha p^\alpha, \tag{5.8}$$

and

$$p^\alpha = p_1^{\alpha_1} p_2^{\alpha_2} \ldots p_d^{\alpha_d}, \quad p = (p_1, \ldots, p_d)^T \in \mathbb{R}^d,$$

where $\|.\|$ represents the Euclidean norm on $\mathbb{R}^d$, $\alpha = (\alpha_1, \ldots, \alpha_d)$ with $|\alpha| = \alpha_1 + \ldots + \alpha_d$. Schaback and Wu [91] given the following form for the local error estimate for thin plate spline interpolation at $x \in \Omega$,

$$|u(x) - s(x)| \leq C \cdot h^k_{\mathcal{N},\varrho}(x), \tag{5.9}$$

where C is a positive constant which depends on u, and $h_{\mathcal{N},\varrho}(x)$ is the local fill distance of $\mathcal{N}$ around x for some positive radial ϱ.

In [7], Iske et al. observed that reduction in the local fill distance, $d_{\mathcal{N}}(y)$ can improve the local error. Let $\mathbf{x}_1 = (x_{1,1}, x_{1,2}) \subseteq \mathbb{R}^2$, $d = 2$, and $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \subset \mathbb{R}^2$, the thin plate spline for our 2-D Black-Scholes equation is

$$s(\mathbf{x}) = \sum_{j=1}^{N} \lambda_j \|\mathbf{x} - \mathbf{x}_j\|^2 \log(\|\mathbf{x} - \mathbf{x}_j\|) + d_0 + d_1\mathbf{x}_1 + d_2\mathbf{x}_2. \tag{5.10}$$

Iske et al. [49] have shown that the thin plate spline interpolation is second order accurate. Therefore, the thin plate spline is used for our local approximation in the error indicator.

5.3.1 Adaptive algorithm

In our 2-D adaptive algorithm, the adaptive cycle is the same as the 1-D (the approximation of the solution, error indicator, refinement and data output) case, except we included a coarsening strategy. Firstly, for a given set of data $\mathbf{x} = (x_1, x_2) \subseteq \mathbb{R}^2$, and $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \subset \mathbb{R}^2$, we use the MQ basis function to approximate the solution of $U = \{U_1, \ldots, U_N\}$. Once the approximation solution $u = \{u_1, \ldots, u_N\}$ is generated, it passes to our error indicator. The error indicator checks our global approximation solution $u = \{u_2, \ldots, u_{N-1}\}$ against local approximation solution at 8 nearest points ($\mathcal{N}_{x_i}$) to x_i with $2 \leq i \leq N-1$. A good approximation is defined by the residual with predefined threshold err_{ref}, which is $r_{\text{err}} =| u_N(x_i) - u_{\mathcal{N}_{x_i}}(x_i) |$ where $\mathcal{N}_{x_i}$ denotes the 8 nearest points without x_i itself. Then the decision is passed to our refinement phase, the Voronoi points are inserted if the refinement phase is required. Figure 5.3 shows an example of refinement for one point in uniform nodes set, and the red nodes are the Voronoi points which have been inserted to refine the error for a specific point x. Because of the non-coarsening strategy involved, the Voronoi points in our problem are rectangular nodes. The refinement region in here excludes our boundary nodes. The final step is the solution output.

Algorithm 2 Algorithm for adaptive radial basis function - two dimensional non-dividend payment Black-Scholes equation

```
1. Input the parameters (included $err_{\text{ref}}$, node distribution).
2. Use MQ basis function and construct the node set.
3. Call 2-D refinement function (output - node distribution, adaptive shape parameter, approximation of solution $u$).
4. Boundary update.
for $i = 0$ to $T$ do
    Compute the differentiation matrices.
    Compute the coefficient matrix with output of refine function and use time integration scheme.
    Recall 2-D refinement function, compute new set of node distribution, adaptive shape parameter, approximation of solution $u$.
    Boundary update
end for
Compute the value of option
```

Algorithm 2 is the main function, and Subroutine 1 is the 2-D refinement function. In the 2-D problem, we used the same adaptive shape parameter strategy as in 1-D, which is selecting the nearest distance for each node. Here we demonstrate the refinement strategy in Figure 5.2. In the left panel of Figure 5.2, the refinement node is coloured with red and the rectangular Voronoi points are colored with blue which are the new added nodes. We also excluded our boundary nodes in the refinement region in our 2-D problem.

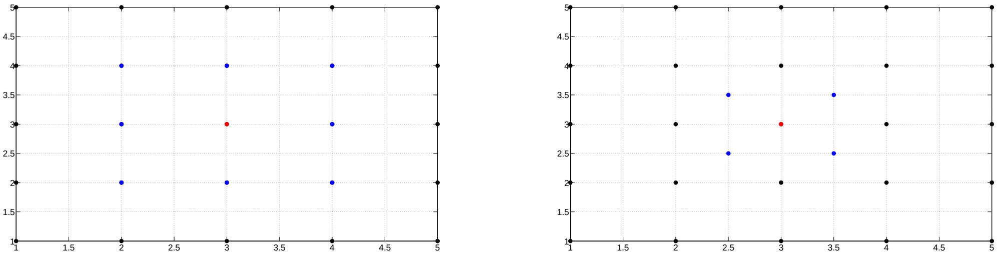

Figure 5.2: An example of refinement in 2-D case

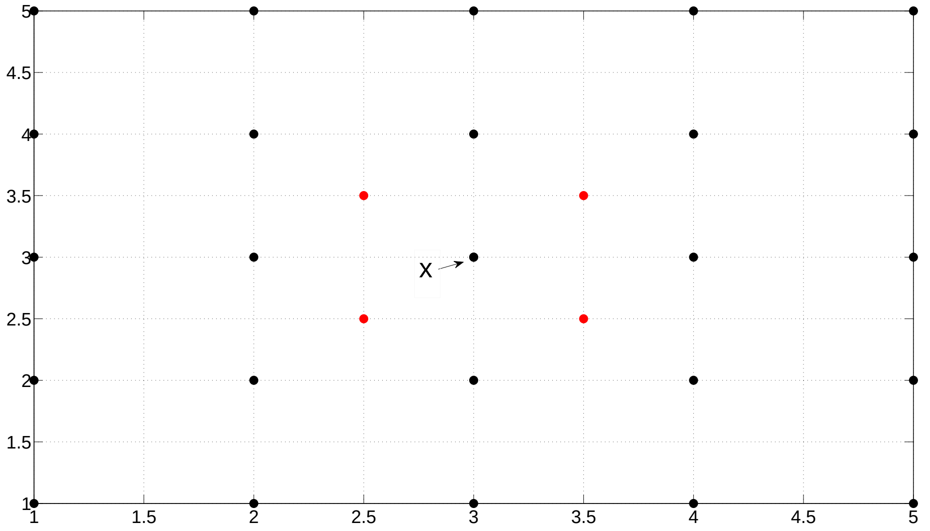

FIGURE 5.3: Refinement of a point x in 2-D, red nodes are inserted Voronoi points

Subroutine 1: 2-D Refinement function

```
1. Input $err_{\mathrm{ref}}$, node distribution.

2. Compute number of nodes.

while ($r_{\mathrm{err}} > err_{\mathrm{ref}}$) do

    Compute adaptive shape parameter set.

    Compute the MQ-interpolant at u

    for $2 \leq i \leq N-1$ do
        a) For each $x_i$, determine $\mathcal{N}_{x_i}$

        b) Compute $\tilde{u}_{\mathcal{N}_{x_i}}$, the local interpolant using thin plate spline;

        c) Compute $r_{err} =\mid u_N(x_i) - \tilde{u}_{\mathcal{N}_{x_i}}(x_i) \mid$.
    end for

    Refine with Voronoi points if $r_{err} > err_{ref}$.

end while $r_{err} < err_{ref}$

Compute the value of option
```

5.3.2 Numerical simulations

Table 5.2 shows the relative error of 5 evaluation points in Parameter Set 3 for both adaptive and equally spaced methods. In the adaptive method the maximum number of nodes used is 955, but for the equally spaced method, 1600 nodes are used to generate the solution. In simulations we chose $S_{\max} = 3K$, $S_{\min} = 1$ and 30 time steps. In the MQ, the Gaussian and the tensor product MQ basis functions, the shape parameter is $(\log(S_{\max}) - \log(S_{\min}))/N$, where N is number of nodes. We note that our ARBF-MQ uses less nodes and better accuracy than the equally spaced RBF method (40 × 40). Figure 5.4 and Figure 5.5 show the profile of a Margrabe call option in Parameter Set 3 at $t = 0$, we can see that we use more nodes in the region where the non-smooth payoff function is applied.

RBFs / S_1, S_2	100, 90	100, 100	100, 110	90, 100	110, 100
ARBF-MQ	4.1e-03	9.1e-03	2.44e-02	2.55e-02	4.5e-03
MQ (40 × 40)	2.07e-02	3.22e-02	1.067e-01	1.230e-01	2.13e-02
Tensor Product MQ (40 × 40)	1.87e-02	3.28e-02	9.92e-02	1.124e-01	1.98e-02
Gaussian (40 × 40)	2.19e-02	2.86e-02	8.09e-02	9.15e-02	2.24e-02

TABLE 5.2: The relative error of a Margrabe option for both methods

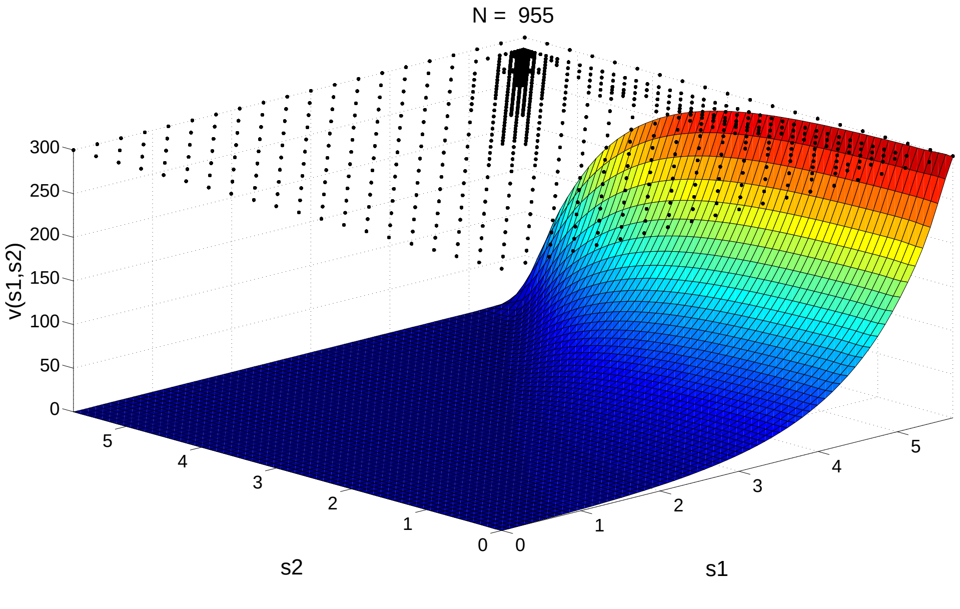

FIGURE 5.4: Profile at $t = 0$ Margrabe call option - ARBF-MQ (1), Parameter Set 3

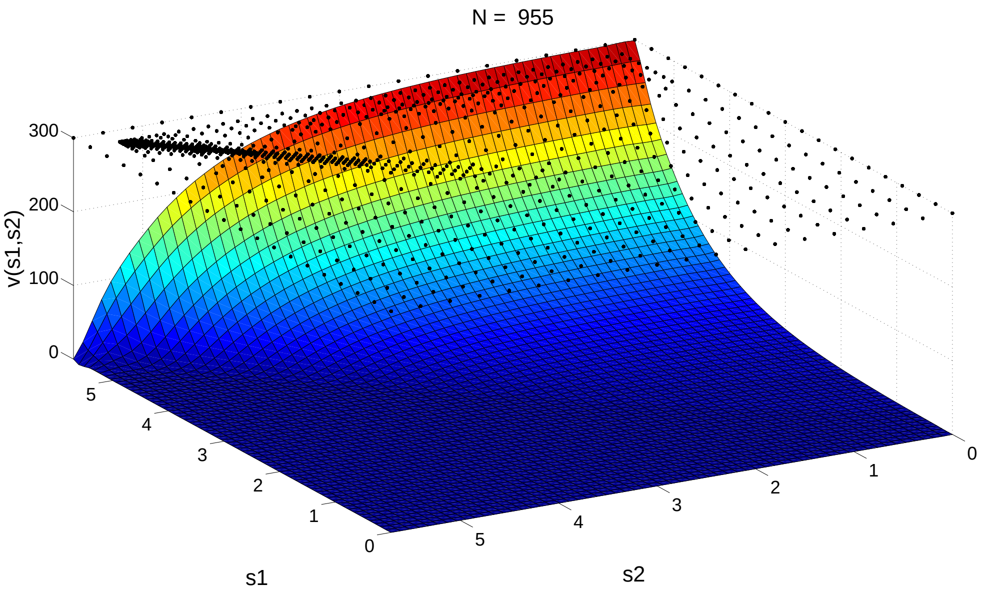

FIGURE 5.5: Profile at $t = 0$ Margrabe call option - ARBF-MQ (2), Parameter Set 3

5.4 Conclusion

In this chapter, we have priced a Margrabe call option for Parameter Set 3 for both the equally spaced RBF method and the adaptive method. We can see that our adaptive method performs well in comparison with the equally spaced method in Table 5.2. Since this is our latest development we had so far, therefore, we have not included the coarsen strategy in our algorithm. However, we believe that by adding coarsen strategy in the algorithm can reduce the number of nodes we used for the same tolerance and possibly reduce the computational cost as less nodes are required in each time steps. For instance, after the immediate refinement of the non-smooth payoff function, we had a total 920 nodes, if the coarsen strategy is added, we can reduce the number of nodes.

Chapter 6

Conclusions and Future Work

6.1 Conclusions and future work

We introduced a new adaptive RBF method for the pricing of financial contracts, such as a European option, an American option, a barrier up and out option and a Margrabe option. This is an adaptive scheme where the error indicator depends on the residual of the local approximation reconstruction and global approximation interpolation. Iske et al. and Naqvi [6, 77] applied this adaptive scheme to solve many different kinds of PDEs in both 1-D and 2-D PDEs, such as the 1-D KdV equation and 2-D Burger's equation. We have successfully applied this adaptive scheme to solve both the 1-D and 2-D Black-Scholes partial differential equations. In the 1-D problem, the MQ-RBF has been used for approximating the solution of the PDEs. The decision of error indicator r_{err} relies on the predefined error thresholds err_{ref} and err_{crs}. The reconstruction of local approximation plays a crucial part, where r_{err} is the difference between our global approximation solutions and local reconstruction solutions. Because the boundary nodes are not in our refinement and coarsening region, our local reconstruction strategy does not apply to them. The idea of local reconstruction strategy for a selected node x is based on reconstruction of an original function with 8 nearest neighbour nodes of x and their solutions by using the cubic spline. The approximate solution at x can be found with this original function. Once we have the approximate solution at x, we compare it to the global approximation solution we had with MQ-RBF. If $r_{\text{err}} > err_{\text{ref}}$, we refine the node by placing one on left and another one on right (both of them should be placed in the half way of its closest node). The coarsening rule is driving by the condition of $r_{\text{err}} < err_{\text{crs}}$. The maximum number of nodes that we can remove is based on the number of new added nodes in the refinement phase. The maximum number of nodes we can remove is 50% of new added nodes, and in some cases we could have a situation in which we can only remove a certain number of nodes that is way below the level of 50% of new added nodes. Our coarsening rule also allows us to remove the half of total coarsening nodes with the condition of the maximum distance rule. This maximum distance is the distance that we can have for two nodes. The example of refinement and coarsening strategy can be found in Chapter 4.

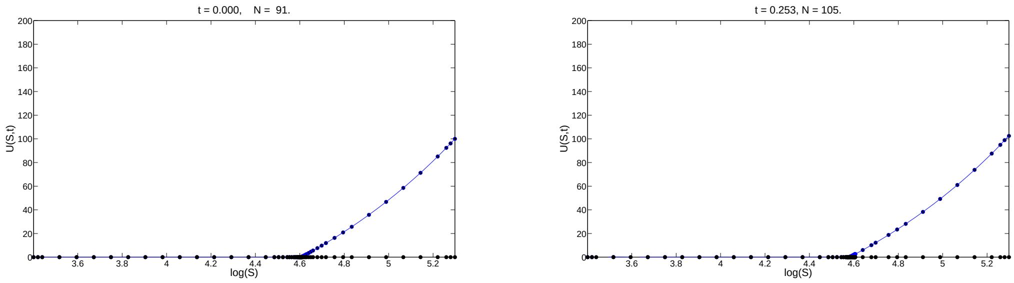

Figure 6.1: An example of adaptive RBF node distribution for a European call option with Parameter Set 2

Figure 6.1 shows one example node behaviour when the error profile changes during the time (from $t = T$ to $t = 0$) for a European call with Parameter Set 2. The right figure gives the node distribution at $t = T$ and left is the node distribution at $t = 0$. We can see our node distribution changes when the error profile moves. Moreover, with our adaptive method we have delivered an accurate solution as FDM in Parameter Set 2, but with less computational cost for the pricing of financial contracts of a European call option (15 times faster), Delta (10 times faster), Gamma (10 times faster), Vega (20 times faster) and barrier up and out call option (3 times faster) in [97]. Also we note that the equally spaced RBF method cannot work well with a problem like a singularly perturbed PDE, unless a large number of nodes is required. Our adaptive method can deal with this sort of problem as our method is similar to the Shishkin meshes [94] which places the extra nodes in the active area. However, for Parameter Set 1, we can gain high accuracy, but the computational cost is our main drawback in comparison with the equally spaced RBFs method. In our future work, we can try to reduce the computational cost by looking at the combination of the adaptive method with sparse grids for the 2-D problem (detail of sparse grids can be found in [13, 44, 43]), where it might help us to reduce the computational cost and improve the node distribution. In our numerical experiments, we note that for Parameter Set 1, after we reduced the error caused by the non-smooth initial condition and extra time steps applied, it seems our node distribution does not change. For this reason we can pause our coarsening and refinement strategy as the coarsening and refinement strategy are expensive in our adaptive method and take about 80% of our total computational time. Now, we propose a new adaptive method in Algorithm 3 (main algorithm) and Subroutine 1 which is more efficient than the one in Chapter 4 based on the computational cost. This new adaptive algorithm delivers a similar solution when compared with our adaptive method in [97], but with about $\frac{1}{30}$ of CPU time for the European call option and around $\frac{1}{50}$ of CPU time for the barrier up and out call option for Parameter Set 1. In Subroutine 1, we also modified our coarsening strategy with the rule of deleting the maximum 2% of coarsening nodes. For the coarsening nodes, we firstly reordered them from the smallest error to largest error, then remove 2% of coarsening node with the smallest errors. We also use the small uniform node set (20 nodes to start with) which is more efficient than the redistribution strategy that we have proposed in Chapter 4. Our simulation shows that we have used the maximum of 41 nodes for the barrier up and out call option and 45 nodes for the European call option, respectively.

However, for Parameter Set 2, we need to constantly apply the coarsening and refinement strategy due to the change of error profile, which means we need to develop an indicator which can help to define whether we need to pause or not.

Algorithm 3 Algorithm for adaptive radial basis function - one dimensional non-dividend payment Black-Scholes equation

```
1. Input the parameters (included $err_{\text{ref}}$, $err_{\text{crs}}$) and coarsen or refine region, node distribution, maximum distance.

2. Choose a basis function and construct the node set.

3. Call Refinement and coarsening function (output - node distribution, adaptive shape parameter and approximation of solution $u$).

4. Boundary Update.

5. Extra time steps ($i = 0$ to $T_1$).

for $i = T_1$ to $T$ do

    Compute the differentiation matrices.

    Compute the coefficient matrix and use the time integration scheme.

    Compute the approximation of solution, $u$.

    Boundary Update

end for

Compute the value of option
```

Subroutine 1: Refinement and coarsening function

```
1. Input $err_{\mathrm{ref}}$ , $err_{\mathrm{crs}}$ and coarsen or refine region, node distribution.

2. Compute number of nodes.

while ($r_{\mathrm{err}} > err_{\mathrm{ref}}$) do

    Compute adaptive shape parameter set.

    Compute the MQ-interpolant at u

    for $2 \leq i \leq N-1$ do
        a) For each $x_i$, determine $\mathcal{N}_{x_i}$

        b) Compute $\tilde{u}_{\mathcal{N}_{x_i}}$, the local interpolant using cubic splines;

        c) Compute $r_{err} =\mid u_N(x_i) - \tilde{u}_{\mathcal{N}_{x_i}}(x_i) \mid$.
    end for

    Refine if $r_{err} > err_{ref}$.

    Coarsen if $r_{err} < err_{crs}$.
end while $r_{err} < err_{ref}$

Output value of $u$, $c$ and node distribution ($x$).
```

Due to the limitation of time, we have not included the coarsening strategy in our adaptive algorithm for two assets Black-Scholes PDE. In this problem, our adaptive algorithm can be divided into 4 steps. Step 1, we use the MQ-RBF to approximate the PDE solution, then we pass it to the error indicator which is Step 2. The decision of error indicator only depends on predefined error thresholds err_{ref}. In the error indicator, the residual is $r_{\text{err}} = \mid u_N(x_i) - \tilde{u}_{\mathcal{N}_{x_i}}(x_i) \mid$ is the same as in the 1-D adaptive algorithm, but the local construction of solution is produced by the thin plate spline. For a specific node x, we also used 8 nearest neighbour nodes of x and their solution to reconstruct the original function. The local approximation solution is calculated by inserting x into the original function. For the future work, before we considering the coarsening strategy, we should include one extra condition for the node distribution by defining a rule of inserting the rectangular Voronoi nodes based on the maximum and minimum distance of inserting nodes. This can also can be applied when the coarsening strategy is involved. For instance, in the left figure of Figure 6.2, we try to refine the red node, but in this situation, we should not add the rectangular Voronoi nodes (blue nodes) as it could cause the problem to the node at $(4, 4)$. Because the node clustering at a certain area but it also causes a large gap, this could possibly lead our interpolation matrix to become singular. To fix this problem, we either add some extra rectangular Voronoi nodes which have been colored in red in the right panel of Figure 6.2 or pause the refinement for that node. Figure 6.3 gives an example of inserting the Voronoi nodes at random nodes set, which would be an example of adding the coarsening strategy in our algorithm as the extra nodes needed in the active region. In Figure 6.4 and Figure 6.5, we also note that the profile of node distribution does not change much, where as in the 1-D problem (in Figure 6.1) we can see the nodes moves in time. We believe that the coarsening strategy could reduce the number of nodes. The random points distribution in [6] could be another way to start as well (detail can be found in [5, 6, 59, 60]. These methods could help with the error caused by the non-smooth payoff function as more nodes been placed in the active region. Figure 6.4 shows the profile of node distribution when we applied the non-smooth payoff function at $t = T$ with our refinement strategy. For $t = 0$, the profile of node distribution displays in Figure 6.5. In 2-D problem of Margrabe call option, we show our adaptive method is better than the equally spaced method, but our adaptive method in here is only considered the refinement. The evolution of solution in radial basis function is computationally expensive and this is one of drawbacks we will be concerned in our future work. One possible approach is using the faster evaluation of radial basis function (multipole expansions), as in the work of Beatson et al. [4, 3]. With this method, we could possibly reduce the computational cost for our 2-dimensional problem by using our adaptive RBF method. Throughout our latest work of a 2-D algorithm for a Margrabe option, we not only need to further develop this algorithm and extend it further, but also we are required to do some convergence analysis of our adaptive RBF method for the Black-Scholes PDEs. (For example, the stability of node distribution (CFL condition), shape parameters and the proof of non-singularity of interpolation matrix differentiation).

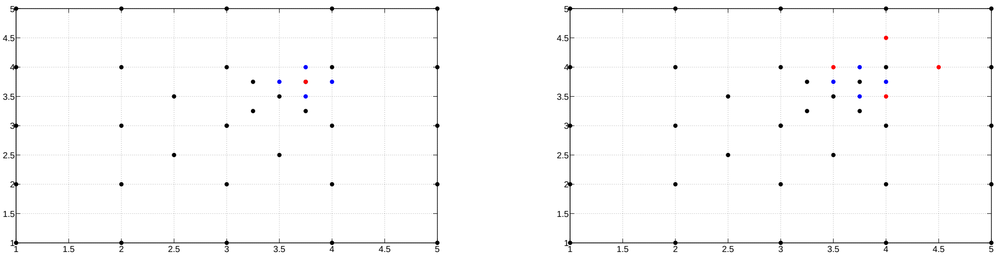

FIGURE 6.2: Condition of refinement of a point x in 2-D

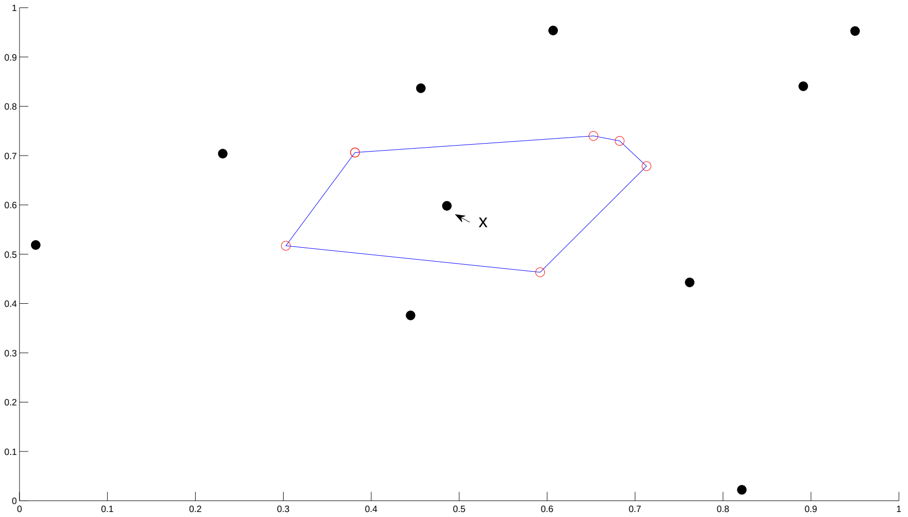

FIGURE 6.3: Refinement of a point x in 2-D, red node are inserted Voronoi points

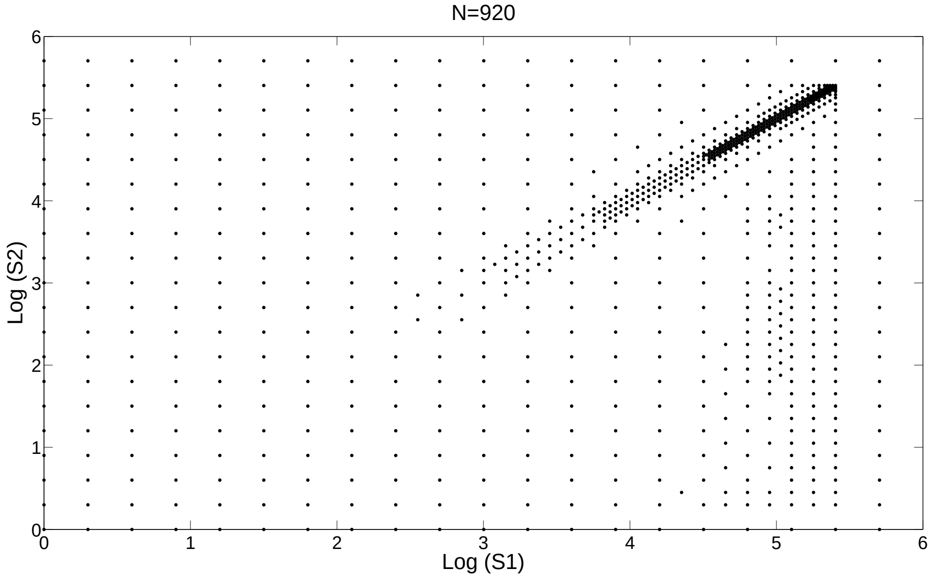

FIGURE 6.4: Profile of node distribution at $t = T$, a Margrabe call option, Parameter Set 3

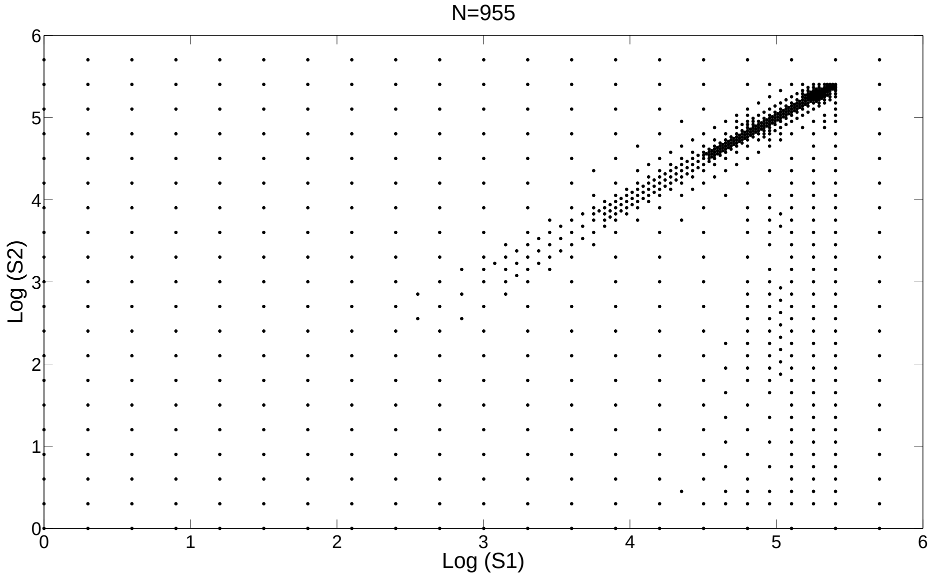

FIGURE 6.5: Profile of node distribution at $t = 0$, a Margrabe call option, Parameter Set 3

Appendix A

Numerical results in Chapter 3

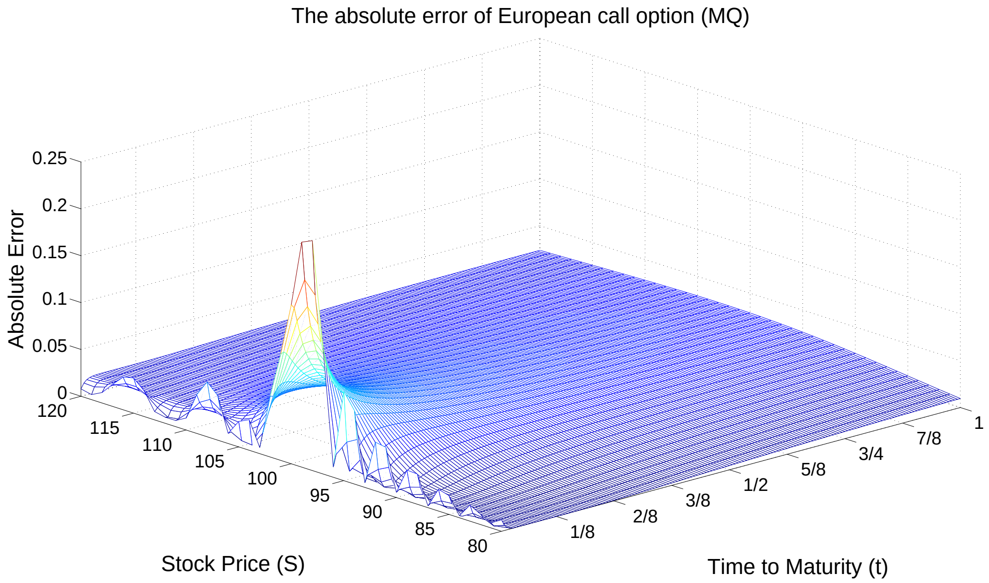

FIGURE A.1: Profile of absolute error of European call option in Parameter Set 1 for uniform distribute nodes with basis function MQ.

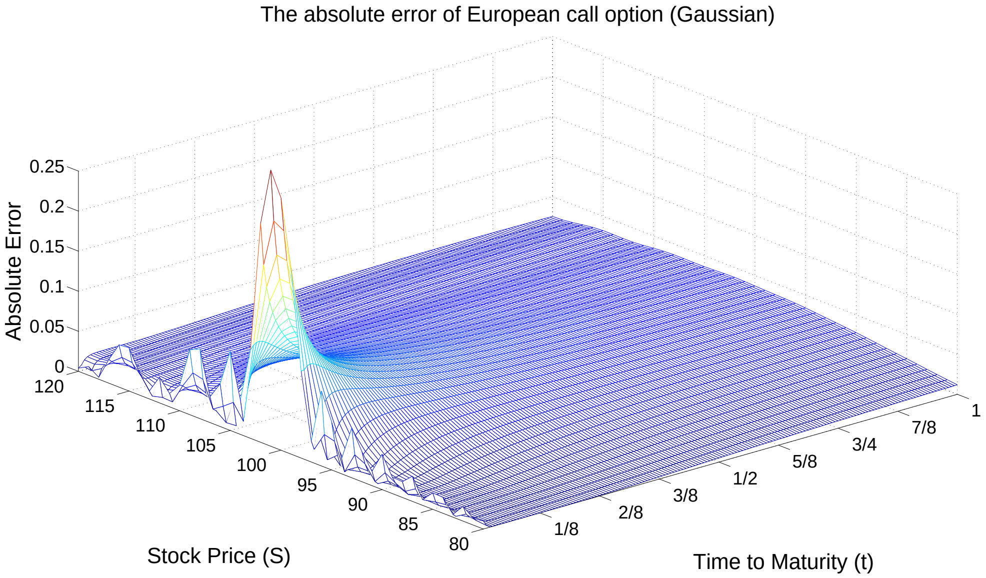

FIGURE A.2: Profile of absolute error of European call option in Parameter Set 1 for uniform distribute nodes with basis function Gaussian.

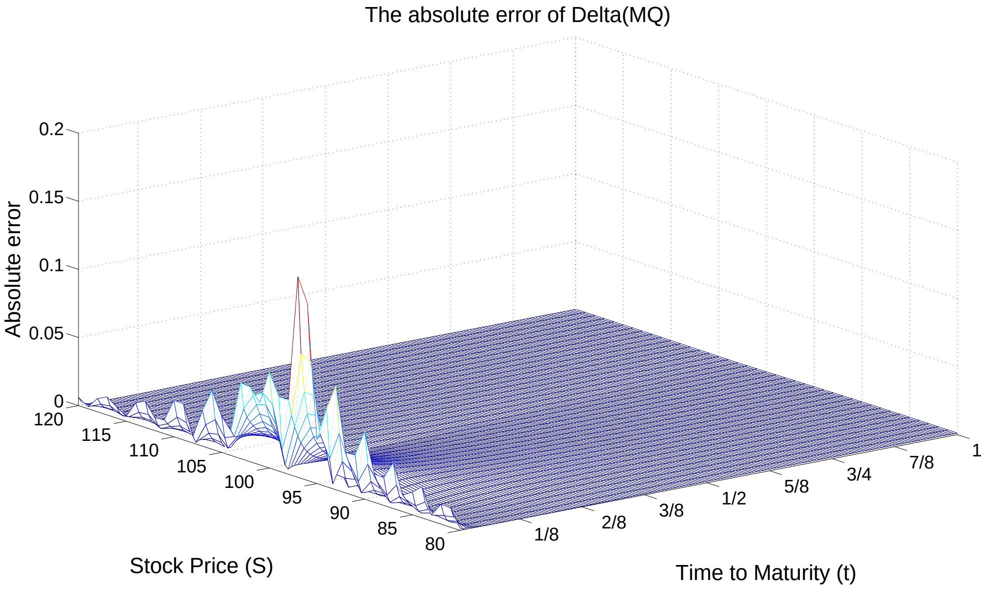

FIGURE A.3: Profile of absolute error of Delta in Parameter Set 1 for uniform distribute nodes with basis function MQ.

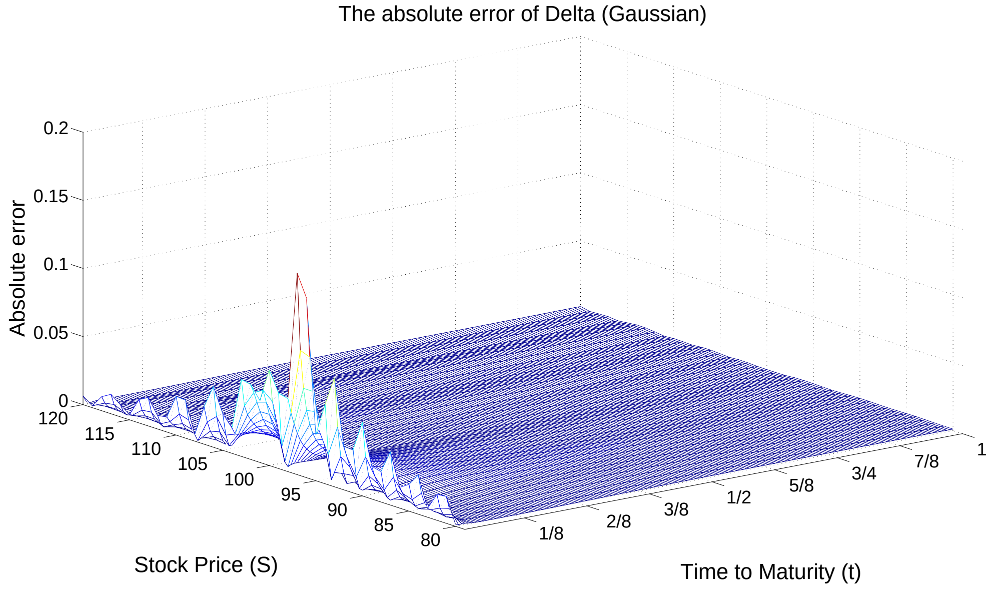

FIGURE A.4: Profile of absolute error of Delta in Parameter Set 1 for Uniform distribute nodes with basis function Gaussian.

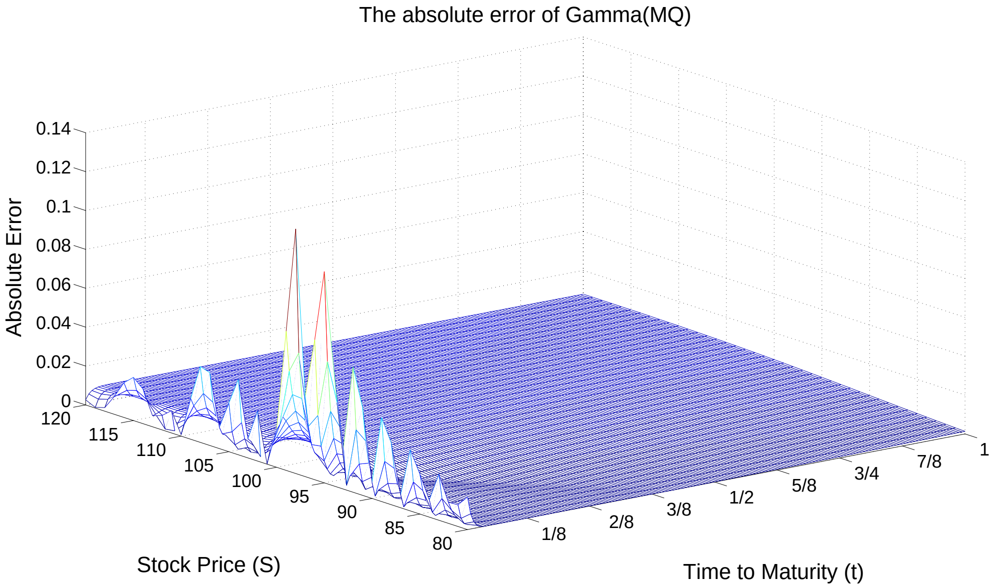

FIGURE A.5: Profile of absolute error of Gamma in Parameter Set 1 for uniform distribute nodes with basis function MQ.

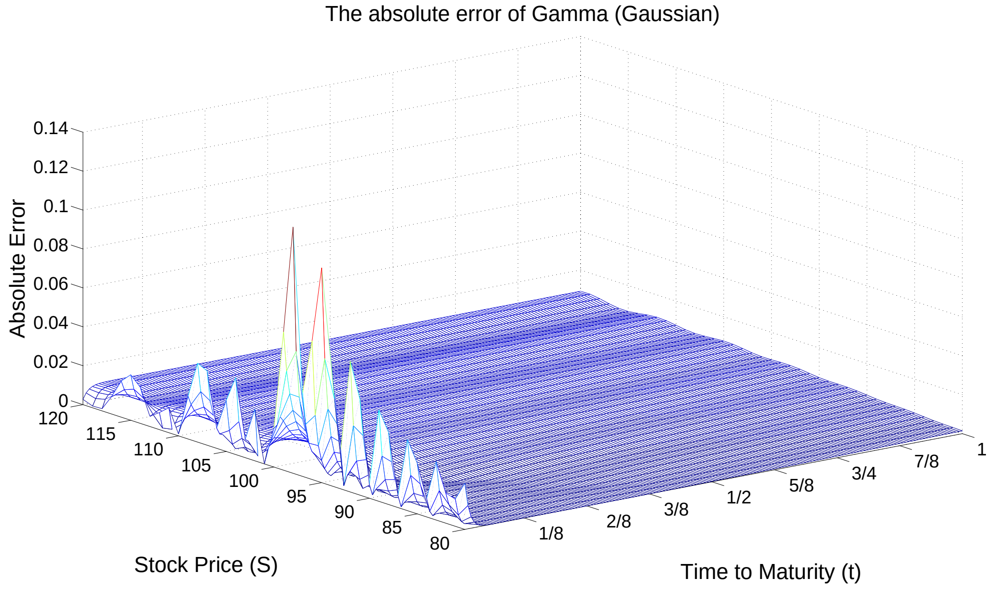

FIGURE A.6: Profile of absolute error of Gamma in Parameter Set 1 for uniform distribute nodes with basis function Gaussian

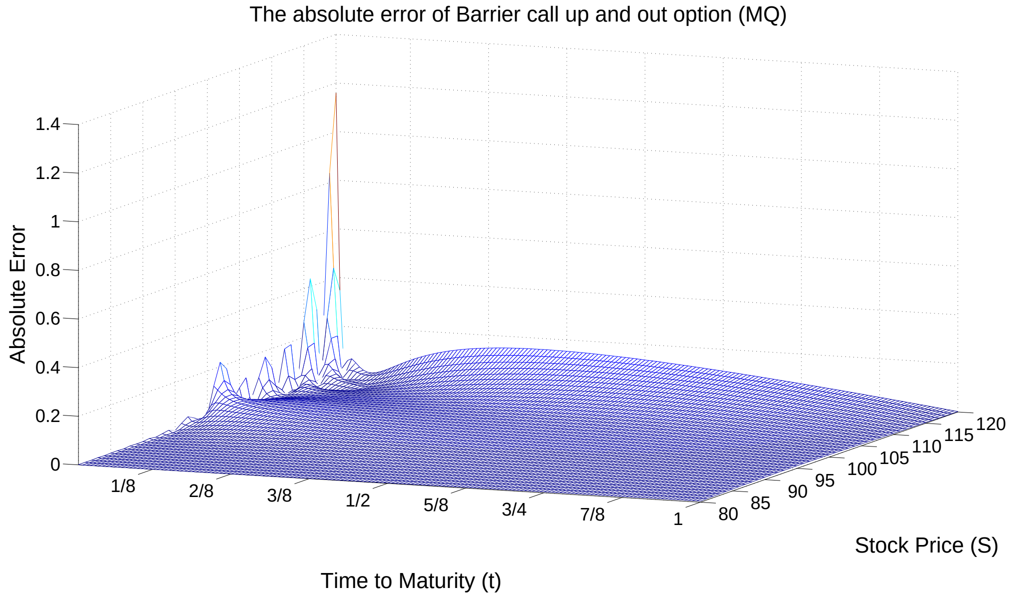

FIGURE A.7: Profile of absolute error of barrier up and out call option in Parameter Set 1 for uniform distribute nodes with basis function MQ.

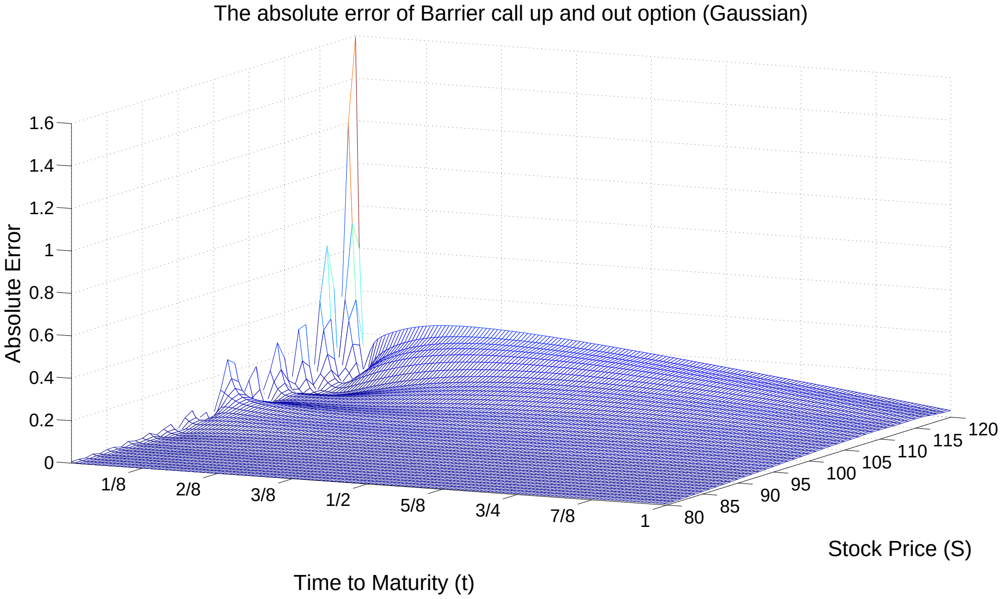

FIGURE A.8: Profile of absolute error of barrier up and out call option in Parameter Set 1 for uniform distribute nodes with basis function Gaussian.

Appendix B

Numerical results in Chapter 4

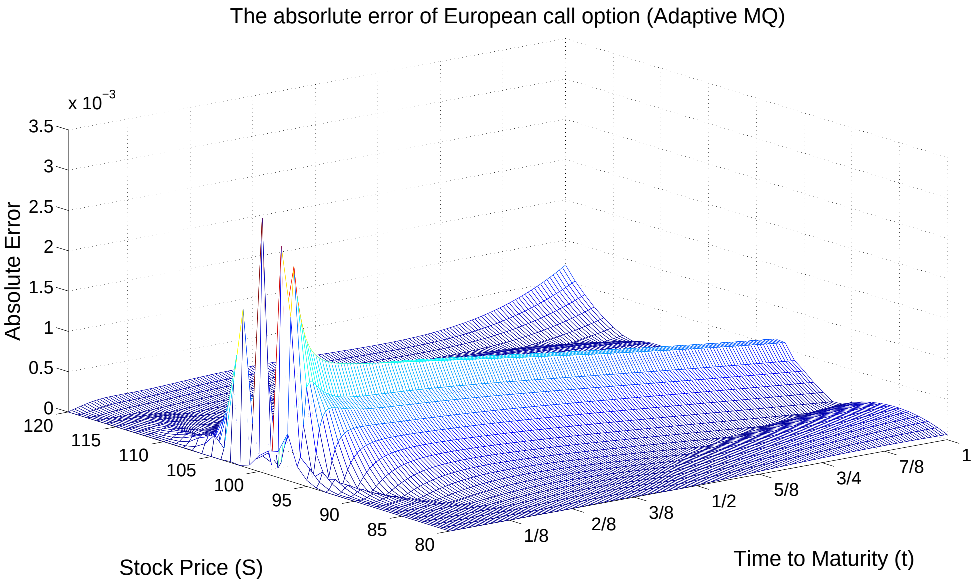

FIGURE B.1: Profile of absolute error of European call option in Parameter Set 1 for ARBF-MQ.

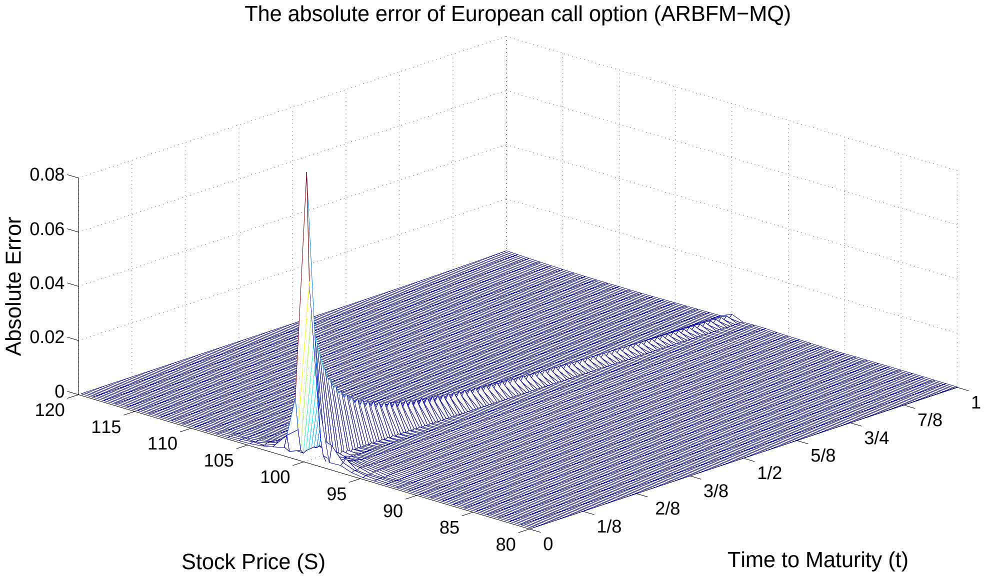

Figure B.2: Profile of absolute error of European call option in Parameter Set 1 for ARBF-MQ, uniform time step.

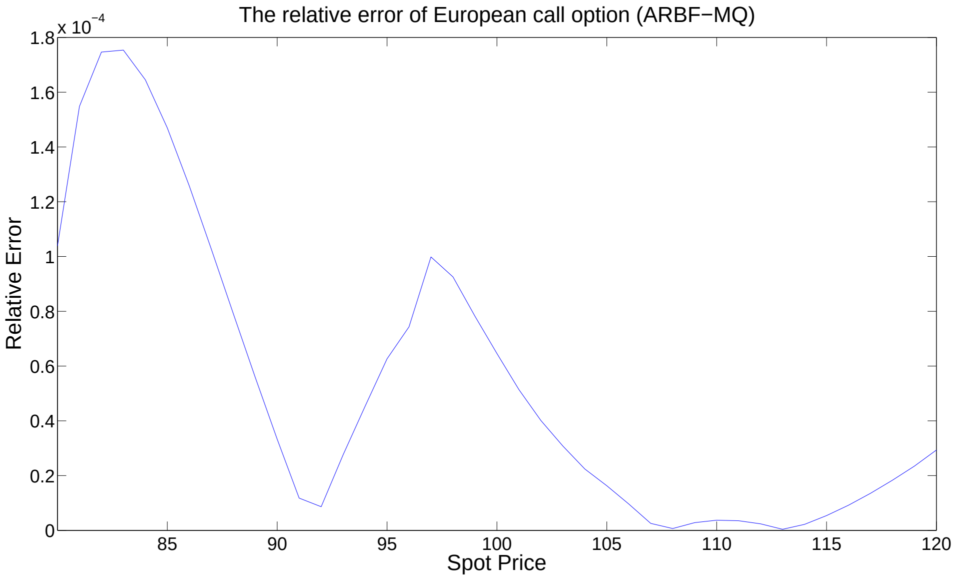

Figure B.3: Relative error of European call option in Parameter Set 1 for ARBF-MQ, t=0.

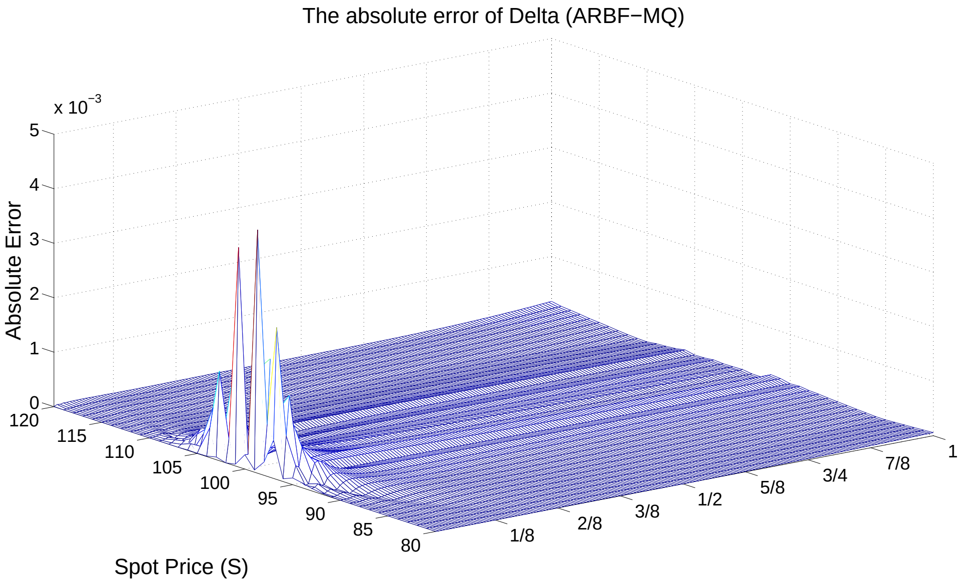

Figure B.4: Surface of absolute error of Delta in Parameter Set 1 for ARBF-MQ.

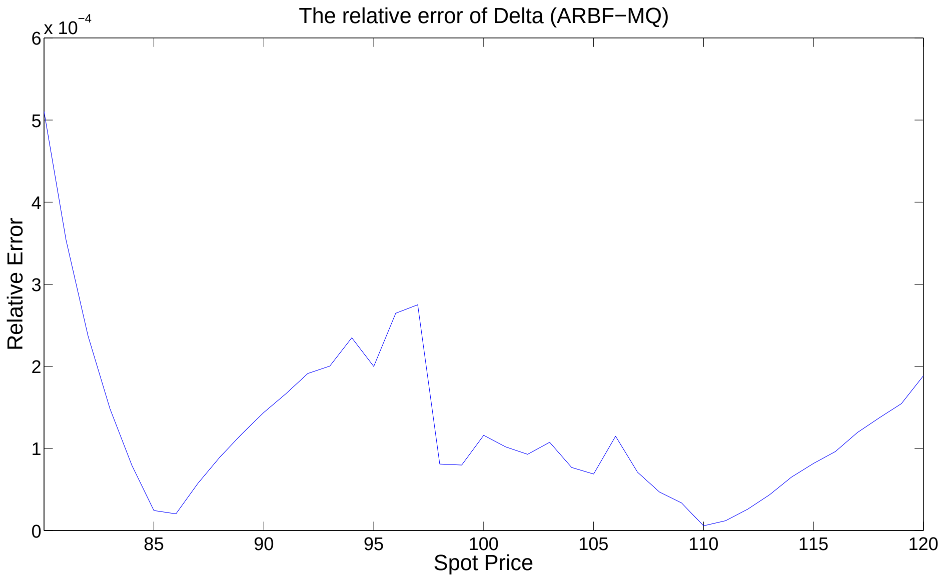

Figure B.5: Relative error of Delta in Parameter Set 1 for ARBF-MQ, t=0

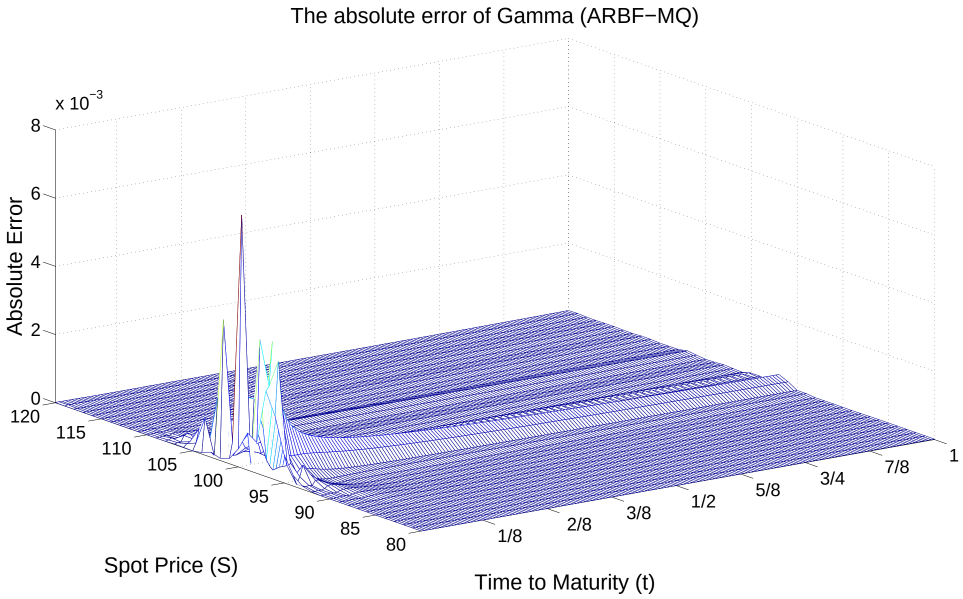

FIGURE B.6: Profile of absolute error of Gamma in Parameter Set 1 for ARBF-MQ.

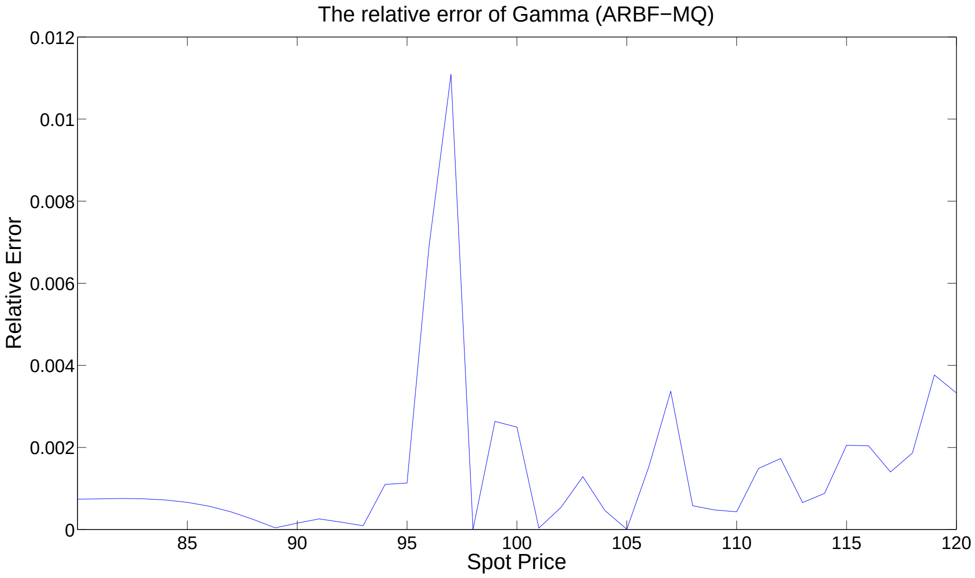

FIGURE B.7: Relative error of Gamma in Parameter Set 1 for ARBF-MQ, t=0.

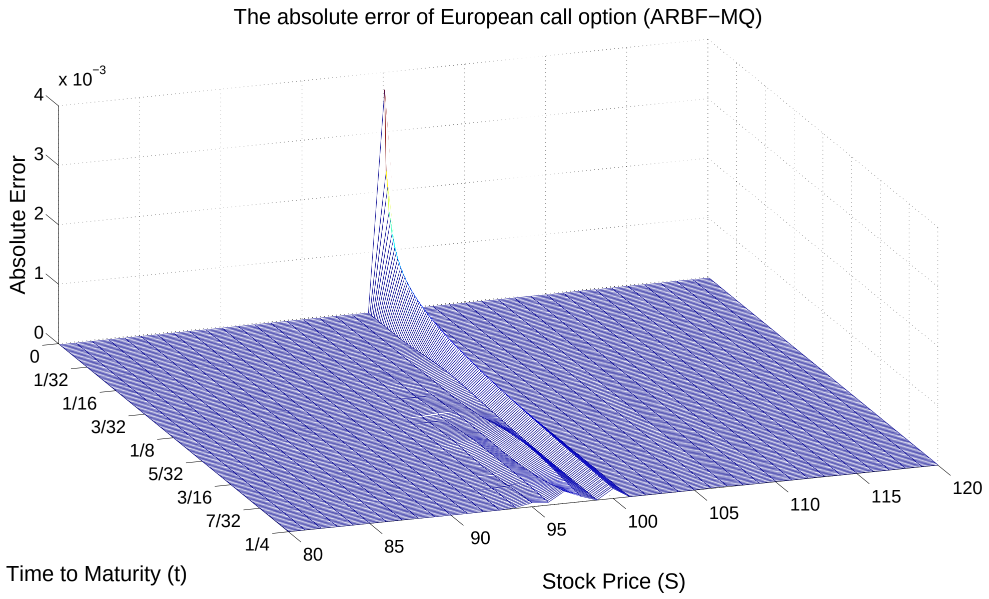

Figure B.8: Profile of absolute error of European call option in Parameter Set 2 with ARBF-MQ, uniform time step.

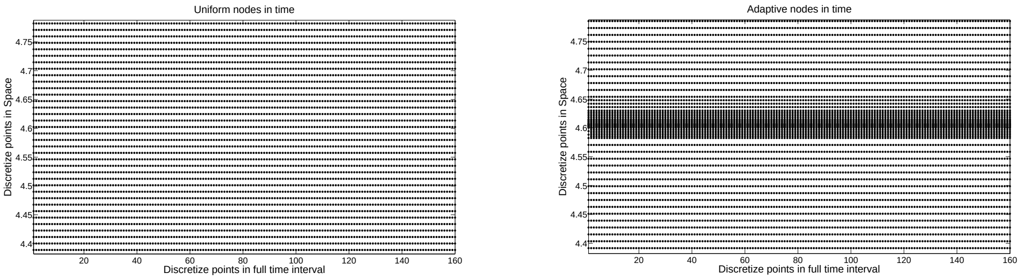

Figure B.9: Uniform and adaptive nodes distribution for Parameter Set 2.

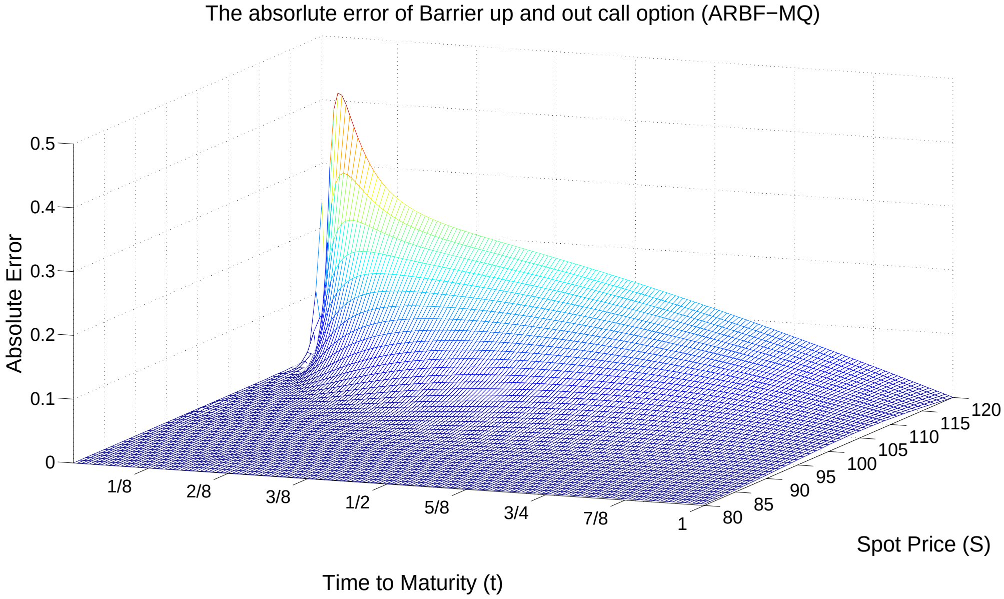

Figure B.10: Profile of absolute error of barrier up and out call option in Parameter Set 1 for ARBF-MQ.

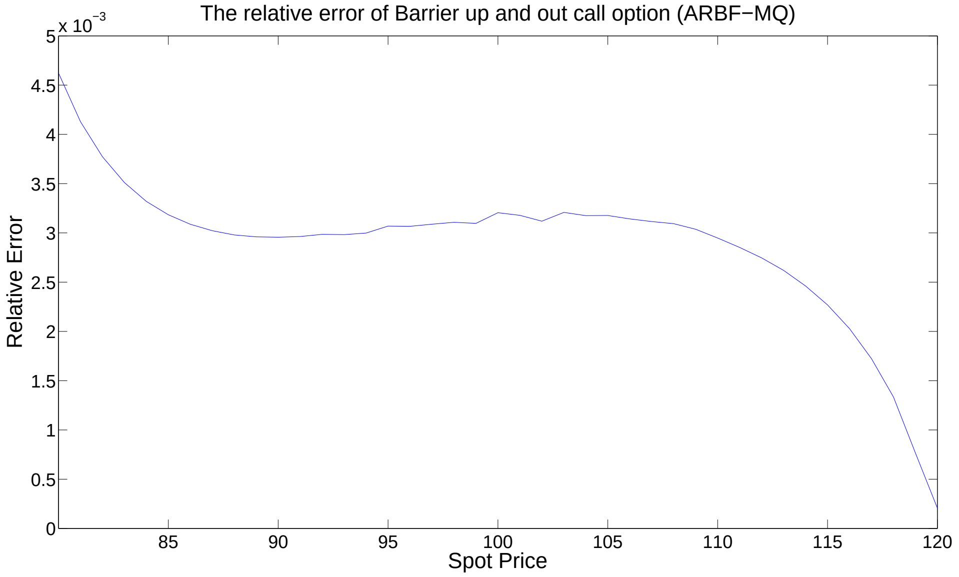

Figure B.11: Surface of absolute error of barrier up and out call option in Parameter Set 1 for ARBF-MQ, at t=0.

Bibliography

[1] I. Babuska and W. Rheinboldt. Error estimates for adaptive finite element computations. *SIAM Journal on Numerical Analysis*, 15(4):736–754, 1978.

[2] K.M. Ball. Invertibility of Euclidean distance matrices and radial basis interpolation. Cat report no.201, Texas A & M University, College Station, 1989.

[3] R.K. Beatson, J.B. Cherrie, and G.N. Newsam. Fast evaluation of radial basis functions:methods for generalized multiquadrics in $\mathbb{R}^n$. *SIAM Journal on Scientific Computing*, 23(5):1549–1571, 2002.

[4] R.K. Beatson and G.N. Newsam. Fast evaluation of radial basis functions:I. *Computers and Mathematics with Applications*, 24(12):7–19, 1992.

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

[6] J. Behrens, A. Iske, and M. Kaser. Adaptive meshfree method of backward characteristics for nonlinear transport equation. *Technische Universitat Munchen, Germany*, 2001.

[7] J. Behrens, A. Iske, and St. Pohn. Effective node adaption for grid-free semi-lagrangian advection. *Detecte Modelling and Discrete Algorithms in Continuum Mechanics,*, pages 110–119, 2001.

[8] J. Berton and R. Eymard. Finite volume methods for the valuation of American options. *Mathematical Modelling and Numerical Analysis*, 40:311–330, 2006.

[9] F. Black and M. Scholes. The pricing of corporate liabilities. *Journal of Political Economy*, 81(3):637–654, 1973.

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

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

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

[13] H.J. Bungartz, A. Heinecke, D. Puger, and S. Schraufstetter. Option pricing with a direct adaptive sparse grid approach. *Journal of Computational and Applied Mathematics*, 236:3741–3750, 2012.

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

[15] P. Carr, R. Jarrow, and R. Myneni. Alternative characterizations of American put options. *Mathematical Finance*, 2:87–106, 1992.

[16] F. Carsten and R. Schaback. Convergence order estimates of meshless collocation methods using radial basis functions. *Advances in Computational Mathematics*, 8(4):381–399, 1998.

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

[18] R. Chan. Pricing options under jump-diffusion models by adaptive radial basis functions. *Working Paper. Bath, U. K.: Department of Economics, University of Bath*, 2010.

[19] R. Charles, T.A. Driscoll, B. Fornberg, and G. Wright. Observations on the behavior of radial basis function approximations near boundaries. *Computers and Mathematics with Applications*, 43(3):473–490, 2002.

[20] R. Charles, B. Fornberg, and G. Wright. Stable computation of multiquadric interpolant for all value of shape parameter. *Computers and Mathematics with Applications*, 47(5-6):497–523, 2004.

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

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

[23] J. Cox, S. Ross, and M. Rubinstein. Option pricing: A simplified approach. *Journal of Financial Economics*, 7:229–263, 1979.

[24] O. Davydov and D.T. Oanh. Adaptive meshless centres and rbf stencils for poisson equation. *Journal of Computational Physics*, 230(2):287–304, 2011.

[25] J. Dewynne, S. Howison, and P. Wilmott. Option pricing: Mathematical models and computation. *Oxford: Oxford University Press*, 1993.

[26] J. Dewynne, S. Howison, and P. Wilmott. The mathematics of financial derivatives. *Cambridge: Cambridge University Press*, 1995.

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

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

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

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

[31] J. Duchon. Splines minimizing rotation invariant semi-norms in Sobolev spaces. *Constructive Theory of Functions of Several Variables, W.Schempp and K. Zeller*, pages 85–100, 1977.

[32] G.E. Fasshauer. *Meshfree approximation methods with matlab.* World Scientific, 2007.

[33] G.E. Fasshauer and C. Frank. Solving partial differential equations by collocation with radial basis functions. *Applied Mathematics and Computation*, 93(1):73–82, 1998.

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

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

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

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

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

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

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

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

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

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

[44] T. Gerstner and M. Griebel. *Numerical integration using sparse grids.* Spring, 1998.

[45] R. Geske and K. Shastri. Valuation by approximation: A comparsion of alternative option valaution techniques. *Journal of Financial and Qualitative Analysis*, 20(1):45–71, 1985.

[46] P. Glasserman. *Monte Carlo methods in financial engineering.* Springer, New York, 2004.

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

[48] D. Grant, G. Vora, and D. Weeks. Simulation and the early exercise option problem. *Journal of Financial Engineering*, 5(3), 1996.

[49] T. Gutzmer and A. Iske. Detection of discontinuities in scattered data approxiamtion. *Numerical Algorithm*, 16(2):155–170, 1997.

[50] A. Heryudono, E. Larsson, and A. Safdari-Vaighani. A radial basis function partition of unity collocation method for convection diffusion equations arising in financial applications. *SIAM Journal on Scientific Computing*, 2014.

[51] F.J. Hickernell and Y.C. Hon. Radial basis function approximation of the surface wind field from scattered data. *International Journal of Applied Science and Computations*, 4(3):221–247, 1998.

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

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

[54] Y.C. Hon, R. Schaback, and X. Zhou. An adaptive greed algorithm for solving large rbf collocation problems. *Numerical Algorithms*, 32:13–25, 2003.

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

[56] C. Hu and C.W. Shu. A discontinuous galerkin finite element method for hamilton–jacobi equations. *SIAM Journal on Scientific Computing*, 21(2):666–690, 2006.

[57] H.Z. Huang, M.G. Subrahmanyam, and G.G. Yu. A recursive integration method. *Review of Financial Studies*, 9(1):277–300, 1996.

[58] S. Ikonen and J. Toivanen. Operator splitting methods for American option pricing. *Applied Mathematics Letters*, 17(7):809–814, 2004.

[59] A. Iske. Multiresolution methods in scattered data modelling. *Lecture Notes in Computational Science and Engineering, Springer-Verlag Berlin Heidelberg*, 37:83–102, 2004.

[60] A. Iske. On the construction of kernel-based adaptive particle methods in numerical flow simulation. *Springer-Verlag Berlin Heidelberg*, 120:197–221, 2013.

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

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

[63] O. Karakashian and F. Pascal. A posteriori error estimates for a discountinuous Galerkin approximation of second-order elliptic problems. *SIAM Journal on Numerical Analysis*, 46(6):2374–2399, 2003.

[64] Y.K. Kwok and L. Wu. A front-fixing finite difference method for the valuation of American options. *SIAM Journal on Scientific Computing*, 30(4):2158–2180, 2008.

[65] E. Larsson and B. Fornberg. Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions. *Computers and Mathematics with Applications*, 49:103–130, 2005.

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

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

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

[69] G. Liu. *Mesh Free Methods: Moving Beyond the Finite Element Method.* CRC Press, 2003.

[70] P. Lotstedt, J. Persson, L.V. Sydow, and Tysk J. Space time adaptive finite difference method for european multi-asset options. *Computers and Mathematics with Applications*, 53(8):1159–1180, 2007.

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

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

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

[74] W. Margrabe. The value of an option to exchange one asset for another. *Journal of Finance*, 33:177–186, 1978.

[75] S. Memon. Finite element method for American option pricing: A penalty approach. *International Journal of Numerical Analysis and Modeling, Series B*, 3(3):345–370, 2012.

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

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

[78] F.J. Narcowich and J.D. Ward. Norms of inverses and condition numbers for matrices associated with scattered data. *Journal of Approximation Theory*, 64(1):69–94, 1991.

[79] F.J. Narcowich and J.D. Ward. Norm estimates for the inverses of a general class of scattered data radial function interpoation matrices. *Journal of Approximation Theory*, 69(1):84–109, 1992.

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

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

[82] M.J.D. Powell. Radial basis functions for multivariable interpolation: a review, numerical analysis. *Longman Scientific and Technical*, pages 223–241, 1987.

[83] M.J.D. Powell. The theory of radial basis function approximation, advances in numerical analysis III. *Oxford: Clarendon Press*, page 105, 1992.

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

[85] F.P. Preparata and M.I. Shamos. *Computational Geometry.* Springer, New York, 1985.

[86] C. Reisinger and G. Wittum. Efficient hierarchical approximation of high-dimensional option pricing problems. *SIAM Journal on Scientific Computing*, 29(1):440–458, 2006.

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

[88] H.-G Roos, M. Stynes, and L. Tobiska. *Numerical methods for singularly perturbed differential equations.* Springer, 1996.

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

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

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

[92] A. Schmidt and K.G. Siebert. *Design of adaptive finite element software.* Springer, 2005.

[93] Q. Shen. A meshless method of lines for the numerical solution of kdv equation using radial basis functions. *Engineering Analysis with Boundary Elements*, 33(10):1171–1180, 2009.

[94] G.I. Shishkin. Grid approximation of singularly perturbed boundary value problem with convective ter. *Journal of Numerical Analysis*, 5:173–187, 1990.

[95] S. Song and L. Tang. A tvd-type method for 2d scalar Hamilton Jacobi equations on unstructured meshes. *Journal of Computational and Applied Mathematics*, 195(1-2):182–191, 2006.

[96] S.E. Stead. Estimation of gradients from scattered data. *Journal of Mathematics*, 14(1):265–279, 1984.

[97] L.V. Sydow, L.J. Hook, E. Larsson, E.and Lindstrom, S. Milovanovic, J. Persson, V. Shcherbakov, Y. Shpolyanskiy, S. Siren, J. Toivanen, J. Walden, M. Wiktorsson, M.B. Giles, J. Levesley, J. Li, C.W. Oosterlee, M.J. Ruijter, A. Toropov, and Y. Zhao. BENCHOP-the BENCHmarking project in option pricing. *International Journal of Computer Mathematics*, forthcoming 2015.

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

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

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