

A radial basis function approach to reconstructing the local
volatility surface of European options

James Glover

February 10, 2010



I declare that this dissertation is my own unaided work. It is being submitted for the degree of Master of Science at the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination at any other university

James Glover
10 February 2010

Abstract

A key problem in financial mathematics is modelling the volatility skew observed in options markets. Local volatility methods, which is one approach to modelling skew, requires the construction of a volatility surface to reconcile discretely observed market data and dynamics. In this thesis we propose a new method to construct this surface using radial basis functions. Our results show that this approach is tractable and yields good results. When used in a local volatility context these results replicate the observed market prices. Testing against a skew model with known analytical solution shows that both prices and hedging parameters are accurately reconstructed, with best case average relative errors in pricing of 0.0012. While the accuracy of these results exceeds those reported by spline interpolation methods, the solution is critically dependent upon the quality of the numerical solution of the resultant local volatility PDE's, heuristic parameter choices and data filtering.

Acknowledgements

I would like to thank my supervisor, Professor Ali - without his patience, guidance, help and encouragement this thesis would not have been possible. I would also like to thank my parents. My father for sharing his years of scientific experience, his tireless proofreading, advice and ultimately for cultivating my interest in science from a young age, and my mother for all her encouragement and being my moral support over the years.

Contents

1	Synopsis	7
2	Introduction - Options and option trading in a Black-Scholes world	9
2.1	The Black-Scholes model	10
2.2	Option risks - The Greeks	12
2.2.1	Delta	13
2.2.2	Theta	13
2.2.3	Vega	13
2.2.4	Gamma	14
2.3	The realities of option trading: Implied volatility	15
3	The volatility smile problem	17
3.1	Causes of volatility skew and term structure	19
3.1.1	Structural flaws in Black-Scholes	19
3.1.2	Behaviorial explanations of the skew	19
3.2	Approaches to modelling the volatility surface	20
3.2.1	Jump-diffusion models	20
3.2.2	Stochastic volatility	21
3.2.3	Local volatility models	22
4	Radial basis functions	23
4.1	Radial basis functions	23
4.2	Approximation using radial basis functions	25
4.2.1	The optimal weight vector	26
4.3	Model specification	27
4.3.1	The heuristic approach	27
5	Reconstructing the local volatility surface using radial basis functions	29

5.1	Simplifying assumptions	29
5.2	The procedure	30
5.2.1	The initial weight vector	30
5.2.2	Evaluating the cost function	30
5.3	Implementation considerations	32
5.4	The optimisation algorithms	32
5.4.1	The Nelder-Mead Simplex algorithm	32
5.4.2	Trust region optimisation	33
5.4.3	Implementation of the optimisation algorithms	34
6	Experiments and results	35
6.1	A problem with an analytical solution	35
6.1.1	Stability of the Greeks	41
6.2	A Real-world problem: Reconstructing the volatility surface of S&P 500 index options	45
7	Discussion and future research	59
8	Conclusion	61
A	Source Code	64
A.1	Train a volatility surface	64
A.2	Numerical example cost function	65
A.3	Analytical example cost function	65
A.4	Generate the distance surfaces, used in calculation of the volatility surface	66
A.5	Generate a volatility surface from a distance surface and weights	67
A.6	Crank Nicholson for the generalised Black-Scholes PDE	68
A.7	Train a linear RBF Network, used to generate initial solutions	69
A.8	Find the optimal parameters of a linear radial basis function approximation	70
A.9	CEV model - used in the analytical example	71

List of Figures

2.1	The payoff of a call option at expiry	10
2.2	The payoff of a put option at expiry	10
2.3	The delta profile of a call option	13
2.4	The theta profile of a call option	14
2.5	The vega profile of a call option	14
2.6	The gamma profile of a call option	15
3.1	The volatility skew of S&P index call options	17
3.2	The volatility skews of S&P index options with varying maturities	18
3.3	The volatility surface of S&P index options	18
3.4	Lognormal distributions with and without kurtosis	19
3.5	Characteristic skew shape of Equity Index Options	20
3.6	Characteristic smile shape of a currency options	21
4.1	A 1D Gaussian function with $c = 0$ and $\beta = 1$	24
4.2	A 1D multiquadratic function with $c = 0$ and $\beta = 1$	24
4.3	A set of data points and an over-fitted function	25
4.4	A set of data points with the likely, correct, function	26
6.1	Local volatility surface defined by $\sigma(S, t) = \frac{15}{S}$	37
6.2	Reconstructed local volatility surface using a Gaussian function set and the Nelder-Mead optimisation method	38
6.3	Reconstructed local volatility surface using a multi quadratic function set and the Nelder-Mead optimisation method	39
6.4	Reconstructed local volatility surface using a thin plate spline function set and the Nelder-Mead optimisation method	40
6.5	Reconstructed local volatility surface using a Gaussian function set and the Trust Region optimisation method	41
6.6	Reconstructed local volatility surface using a multi quadratic function set and the Trust Region optimisation method	42

6.7	Reconstructed local volatility surface using a thin plate spline function set and the Trust Region optimisation method	43
6.8	A comparison of the Greeks	44
6.9	Interpolation results of the initial weight vector using a set of Gaussian functions	46
6.10	Interpolation results of the initial weight vector using a set of multi quadratic functions .	46
6.11	Interpolation results of the initial weight vector using a set of thin plate spline functions .	47
6.12	Reconstructed local volatility surface using a Gaussian function set and the Nelder-Mead optimisation method	48
6.13	Reconstructed local volatility surface using a multi quadratic function set and the Nelder-Mead optimisation method	49
6.14	Reconstructed local volatility surface using a thin plate spline function set and the Nelder-Mead optimisation method	50
6.15	Reconstructed local volatility surface using a Gaussian function set and the Trust Region optimisation method	51
6.16	Reconstructed local volatility surface using a multi quadratic function set and the Trust Region optimisation method	52
6.17	Reconstructed local volatility surface using a thin plate spline function set and the Trust Region optimisation method	53
6.18	Interpolation results of the initial weight vector using a set of Gaussian functions with a larger set of centres	54
6.19	Interpolation results of the initial weight vector using a set of multi quadratic functions with a larger set of centres	54
6.20	Interpolation results of the initial weight vector using a set of thin plate spline functions with a larger set of centres	55
6.21	Reconstructed local volatility surface using a Gaussian function set and the Trust Region optimisation method	56
6.22	Reconstructed local volatility surface using a multi quadratic function set and the Trust Region optimisation method	57
6.23	Reconstructed local volatility surface using a thin plate spline function set and the Trust Region optimisation method	58
7.1	Relative error profile of the thin plate spline approach on the numerical example	60

List of Tables

3.1	Loss as a result of a downward jump across various portfolios	20
6.1	Summarised results using the Nelder-Mead algorithm	36
6.2	Summarised results using the Trust-Region algorithm	37
6.3	Implied volatilities for the S&P 500 index during October 2005	45
6.4	Summarised results using the Nelder-Mead algorithm	45
6.5	Summarised results using the Trust Region algorithm	47
6.6	Summarised results using the Trust Region algorithm with an increased function set size	55
7.1	Comparison of radial basis function and spline interpolation approaches for the analytical problem	59

Chapter 1

Synopsis

The Black-Scholes option pricing formula introduced in Black and Scholes [1973] revolutionised the way the financial markets priced, hedge and understood derivatives, and in particular, options. It provided an analytical framework which allowed for the extraction of prices as well as the risks or sensitivity of the price to the various parameters used to determine price.

So influential was the equation that options were said to exist in a “Black-Scholes” world, and often behaved in the idealistic Black-Scholes manner. The real issue was whether this ideal behaviour was justified, as, in the derivation of this formula, Black and Scholes [1973] made several simplifying assumptions. The downside of not modelling behaviour accurately would be significant risk taken on by the buyer or seller of the option, often without knowledge.

Despite what theoretical models may tell us, ultimately the price of financial instruments such as options are determined by the market - that is the price which market participants are willing to buy and sell them for. After the 1987 Black Monday Wall Street crash there was a noticeable change in the trading price of options [Rubinstein 1994]. One of the key pricing parameters in the Black-Scholes model is the volatility - or the expected standard deviation of the underlying assets returns. When this number was backed out from the option prices using the Black-Scholes formula it was noted that it varied across different options with the same underlying, and when plotted on a graph against the strike price of these options formed a characteristic “skew” or “smile” shape. This however, does not seem intuitively correct as the volatility should only depend on the underlying asset. This phenomenon is known as the volatility smile problem and its existence suggests that the Black-Scholes model, in its original form is not structurally able to accurately describe option prices and their dynamics.

The purpose of this research is to present an approach to price options more accurately given the smile problem. Previous solutions can be divided into two categories - both of which relax the assumption made by Black-Scholes that volatility be constant. Stochastic volatility models add an additional source of randomness to the model (volatility itself is allowed to vary stochastically over time). Local volatility models on the other hand make use of a volatility function or surface which describes the volatility over different maturities and underlying asset prices.

The solution presented in this thesis falls in the local volatility realm. Key to the solution of local volatility models is the determination of this local volatility function by calibration using observed market prices. More precisely, this thesis presents an approach to construct a function or surface by calibration of observed prices to market data.

The approach presented makes use of radial basis function sets as interpolators. Radial basis functions are widely used sets of functions which are particularly useful in interpolating data and are widely used in both pattern recognition and surface interpolation. Application of radial basis function interpolation is complicated by the non-linear nature of the problem. The points on the surface are not directly observable

but instead need to be backed out. As the error of the interpolated surface needs to minimise the error between observed and interpolated values - reducing the problem to a non-linear optimisation problem.

This work is largely based on that by Lagnado and Osher [1997a] and Coleman *et al.* [1999]. For these reasons, the tests and results presented align closely with work presented in Lagnado and Osher [1997a] and Coleman *et al.* [1999] for purpose of proper comparison. The results show that the radial basis function approach to reconstruction of the local volatility surface is a valid one, and the framework established in this thesis has significant potential to extract accurate solutions.

The thesis introduces the reader to options and their pricing in theory and practice in Chapter 2 - which will also explore the key assumptions made in the derivation of the Black-Scholes model. The first part of Chapter 3 introduces this smile phenomenon and suggests possible reasons for its existence. This is followed by a discussion on approaches to pricing options in the presence of the smile. A brief introduction to radial basis functions and the basic theory surrounding them is presented in Chapter 4, followed by a full description of the proposed method in Chapter 5. Finally, Chapter 6 presents results and Chapter 7 provides analysis of these results.

Chapter 2

Introduction - Options and option trading in a Black-Scholes world

Financial options have, over the past few years, jumped to prominence in the global financial marketplace. Their ability to transfer and diversify risk, provide leverage and form the base of highly structured products has seen them emerge as critical instruments, actively traded on regulated markets the world over. However, despite the relative sophistication of pricing and risk analysis options have been around for centuries. Haug and Taleb [2008] provides a good discussion on options in a historical setting.

An option is a contract between two parties which gives, as the name suggests, one party the option to either sell or buy an underlying asset at a given price on a given day in the future. That is, the buyer of an option has the right but not the obligation to exercise the option to deliver the underlying asset. Call options give the buyer the right to buy the underlying asset at a certain price (strike price), on a certain date (maturity date) and put options give the option buyer the right to sell stock at the strike price on the maturity date to the counter party.

As an example assume a person purchases from a counter party a 1 year call option with a strike price of x . The person now has the right, but not the obligation, to buy the underlying asset in 1 year at a price of x . Assume that on the maturity date, 1 year later, the underlying asset is valued at more than x . The option holder would be able to exercise the option and take delivery of an instrument valued at greater than x for only x . Hence, if he immediately sold the asset after exercising, he would make a profit of the excess between the asset value and x . However, if the underlying instrument is worth less than the strike price x , the option buyer would not exercise as he would be paying x for an asset worth less than x . He would therefore abandon the option losing only the payment that was made upfront. This example shows that the owner of a call option makes money if the underlying asset is above the strike and goes up in value, and a downside loss of the premium should the underlying asset price be less than the strike at expiry. Figure 2.1 shows the profile of a call option at maturity.

A similar idea holds for a put option, except the holder of the put option makes a profit if the underlying asset price falls below the strike and loses premium if it is above the strike, as indicated in Figure 2.2. This gives rise to one of the more important uses for options - protection. An investor might have shares in a company that are worth x per share. If the investor is worried that the share may go down over the next year he can buy a put with a strike of x and a maturity of 1 year. This contract guarantees that he will be able to sell his shares in the company at x in 1 year even if the share price is lower than x . This and other strategies will be discussed in following chapters as they play a key role in the smile phenomenon that this dissertation explores.

The key question, at the crux of this research is how to value and model the options during their life.

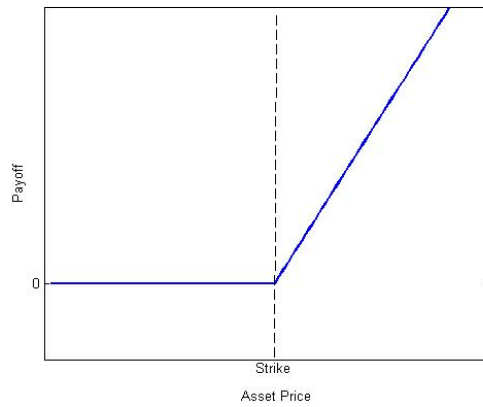


Figure 2.1: The payoff of a call option at expiry

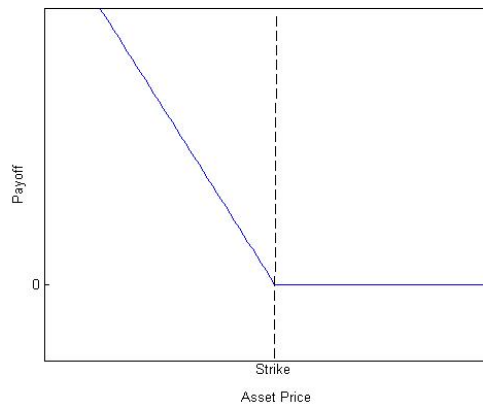


Figure 2.2: The payoff of a put option at expiry

2.1 The Black-Scholes model

Even though option traders have been able to price and manage option risk for centuries, it wasn't until the seminal work of Black and Scholes [1973] that the option world really took off. Black and Scholes [1973] introduced the famous Black-Scholes option valuation model to the world which gave, not only a fair theoretical price based on key assumptions, but also a strategy which allowed an option to be replicated using a portfolio of the underlying asset. This meant that option traders didn't have to hold directional option risk but could hedge it out using the underlying asset and a dynamic strategy.

The derivation given below, taken largely from Hull [2006], gives an insight into the approach and assumptions made by Black and Scholes [1973].

The first of these assumptions is that the underlying asset price is lognormally distributed and follows geometric Brownian motion. That is, given an asset price S , with standard deviation σ and expected drift μ , the asset price follows the stochastic process

$$dS = \mu S dt + \sigma S dz \quad (2.1)$$

over time t , where dz is a Wiener process (a stochastic process with a mean of 0 and a variance of 1). The validity of this assumption will be scrutinised in later chapters.

As the underlying is assumed to follow this process it should suggest the price of an option should be a function of both S and t . One very important result in stochastic calculus, known as Ito's lemma, shows that given a process

$$dx = a(x, t)dt + b(x, t)dz \quad (2.2)$$

a function G of x and t follows the process

$$dG = \left(\frac{\partial G}{\partial x}a + \frac{\partial G}{\partial t} + \frac{1}{2} \frac{\partial^2 G}{\partial x^2} b^2 \right) dt + \frac{\partial G}{\partial x} b dz. \quad (2.3)$$

Using this result and applying it to Equation 2.1 results in

$$df = \left(\frac{\partial f}{\partial S} \mu S + \frac{\partial f}{\partial t} + \frac{1}{2} \frac{\partial^2 f}{\partial S^2} \sigma^2 S^2 \right) dt + \frac{\partial f}{\partial S} S dz, \quad (2.4)$$

where f is the price of an option. Discretising 2.1 and 2.4 gives

$$\Delta S = \mu S \Delta t + \sigma S \Delta z \quad (2.5)$$

and

$$\Delta f = \left(\frac{\partial f}{\partial S} \mu S + \frac{\partial f}{\partial t} + \frac{1}{2} \frac{\partial^2 f}{\partial S^2} \sigma^2 S^2 \right) \Delta t + \frac{\partial f}{\partial S} S \Delta z. \quad (2.6)$$

This leads to the second key assumption of Black and Scholes [1973]. They show the existence of a portfolio of derivatives and underlying assets that would eliminate the Wiener process dz . This portfolio consists of a sale of 1 derivative and the simultaneous purchase of $\frac{\partial f}{\partial S}$ of the underlying asset. The value of this portfolio, Φ , is given by

$$\Phi = -1f + \frac{\partial f}{\partial S} S. \quad (2.7)$$

Key to this idea is that the number of units in the underlying asset needed to maintain the relationship given by Equation 2.7 is a continuous function which varies according to the price of the underlying asset and derivative - this again becomes an important point for discussion in later chapters.

Discretising Equation 2.7 gives

$$\Delta \Phi = -1\Delta f + \frac{\partial f}{\partial S} \Delta S. \quad (2.8)$$

Substituting in values for ΔS and Δf from Equations 2.5 and 2.6 respectively results in

$$\Delta \Phi = \left(\frac{\partial f}{\partial t} - \frac{1}{2} \frac{\partial^2 f}{\partial S^2} \sigma^2 S^2 \right) \Delta t. \quad (2.9)$$

Now, because the Wiener process (randomness) has been eliminated this portfolio is riskless - it has a defined return. An important principle in finance is that riskless portfolios or instruments should always return the risk free rate, r . r is generally defined as the rate which investors will earn by placing money

on deposit without assuming risk or the rate they will pay to borrow money assuming they pose no risk to the lender (in practice, banks charge a spread on either side so these rates are never equal). Assume the portfolio, Φ will return x with absolute certainty to an investor. If x is greater than r then the investor can borrow money at r , use this to fund the purchase of the portfolio and earn x - resulting in a risk free profit. Similarly if x is lower than r , the investor can sell the portfolio and lose x , and put the money earned from selling the portfolio and put it on deposit and earn r , again resulting in a risk free profit. These riskless opportunities are known as arbitrage. The only value for x where arbitrage wouldn't be possible is if $r = x$. Therefore for no arbitrage to exist in a market (another assumption made by Black-Scholes) riskless portfolios should always return the risk free rate, r .

With this knowledge, our portfolio Φ , over Δt should return

$$\Delta\Phi = r\Phi\Delta t \quad (2.10)$$

Substituting in for Φ and $\Delta\Phi$ from Equations 2.7 and 2.9 respectively gives

$$\left(\frac{\partial f}{\partial t} - \frac{1}{2} \frac{\partial^2 f}{\partial S^2} \sigma^2 S^2 \right) \Delta t = r \left(-f + \frac{\partial f}{\partial S} S \right) \Delta t. \quad (2.11)$$

Rearranging Equation 2.11 results in

$$\frac{\partial f}{\partial t} + rS \frac{\partial f}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} = rf, \quad (2.12)$$

which is the celebrated equation of Black and Scholes [1973].

The boundary conditions can be easily determined by the arguments presented earlier in this chapter. For a call option at expiry the option is worth the difference between the current underlying asset price, S , and the strike price, K , if $S > K$ and 0 otherwise. This translates to a boundary condition of

$$f_T = \max(S_T - K, 0), \quad (2.13)$$

where S_T is the asset price at maturity. Again, a similar argument can be used for a put option resulting in the put boundary condition

$$f_T = \max(K - S_T, 0). \quad (2.14)$$

The Black-Scholes equation shows that inputs required for modeling an option are the underlying asset price, S , the strike K , the maturity T , the risk free rate, r , and the volatility, σ . All these parameters are directly observable except for the volatility. It is this parameter which plays such an important role in the model, that will be the main focus of investigation in this dissertation.

2.2 Option risks - The Greeks

Not only does the Black-Scholes model provide a valuation framework for options over their lifetime but also the risks and effects of changes in the various parameters. The sensitivity of the option value to changes in asset price, time, volatility, risk free rates, various second and higher order sensitivities as well as cross sensitivities, collectively known as Greeks for reasons which will become apparent, allow the option market participant to build up a rich view of the risks faced.

2.2.1 Delta

The delta is probably the most important greek and relates the change in an underlying asset price to the change in the option price. It is defined by

$$\Delta = \frac{\partial f}{\partial S} \quad (2.15)$$

and has the profile shown in Figure 2.3.

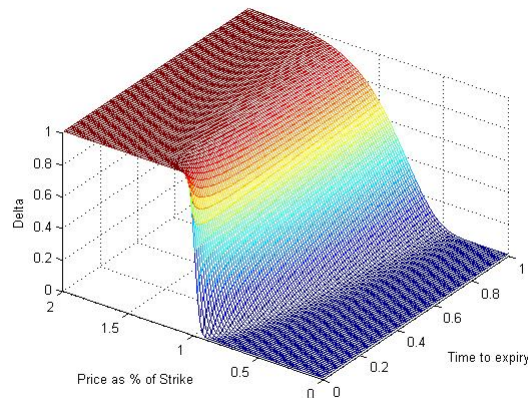


Figure 2.3: The delta profile of a call option

The delta is a key component in the replicating portfolio given by Equation 2.7 which was used to derive the Black-Scholes equation. The delta therefore gives the number of units of the underlying asset required to replicate (and thus hedge) an option at a point in time.

2.2.2 Theta

Theta relates the change in option value to the change in time to expiry. It is often known as option decay as it describes the value the option loses at each time step closer to expiry. It is given by

$$\Theta = \frac{\partial f}{\partial t} \quad (2.16)$$

and has the profile shown in Figure 2.4. The profile shows that options suffer from the greatest time decay when the underlying asset is around the spot and this decay is accelerated the closer the option gets to expiry.

2.2.3 Vega

One of the most important Greeks to options traders is vega which gives the change in option value as volatility of the underlying instrument changes. This importance is due to option traders being essentially traders of volatility, a concept which will be explored in the next section. Vega is given by equation 2.17 and has the profile shown in figure 2.5.

$$V = \frac{\partial f}{\partial \sigma} \quad (2.17)$$

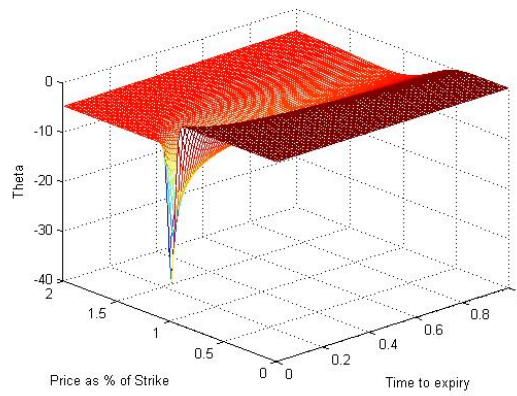


Figure 2.4: The theta profile of a call option

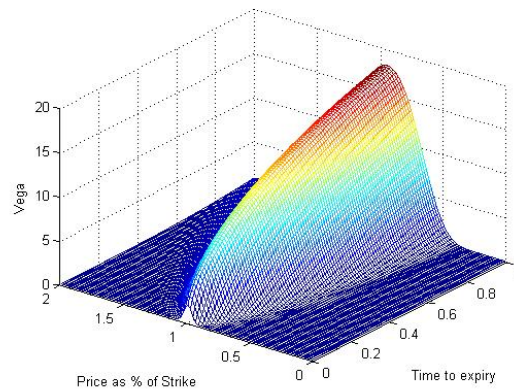


Figure 2.5: The vega profile of a call option

From the profile it can be seen that the closer the underlying asset price to the strike and the further the option is from maturity, the higher the effect of volatility on the price. This observation will become crucial when determining volatility surfaces.

2.2.4 Gamma

Gamma is the most important of the second order Greeks. It relates the change in the delta to the change in the underlying asset price - effectively giving a measure of error in the hedging ratio as the underlying asset price moves. This is critical for a trader who wants to know how sensitive his hedge is to error and what the likely effects to his profit and loss are due to this error. It is a second order derivative given by

$$\Gamma = \frac{\partial}{\partial S} \frac{\partial f}{\partial S}. \quad (2.18)$$

The profile given by Figure 2.6 shows a strong similarity to the Theta - a relationship which will be used to properly understand ideas of volatility trading in the next section.

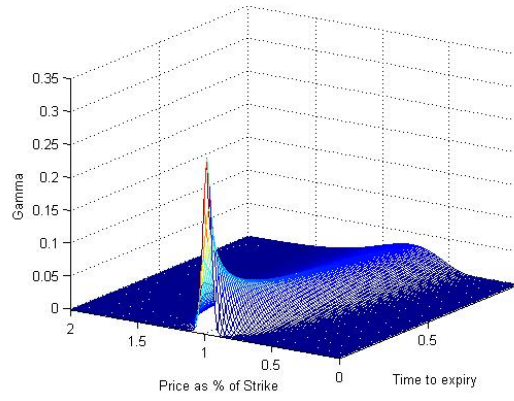


Figure 2.6: The gamma profile of a call option

2.3 The realities of option trading: Implied volatility

The option market can be roughly divided into two sets of participants - price takers and market makers. Price takers are generally large institutional investors like pension, mutual and hedge funds. They use options to put together strategies based on their views on the direction of the underlying asset price movements. Price takers are so called because they will approach market makers who will give them a price which they are prepared to sell and a price at which they are prepared to buy the option at.

While price takers are taking views on the path of the underlying asset price over the life of the option, market makers do not want to be bound by this and therefore need to find a way of hedging out this risk. This can be achieved either by selling or buying options with similar characteristics which offset their positions or by creating a delta hedged portfolio using the dynamic strategy developed by Black-Scholes.

Bossu *et al.* [2005] provide an intuitive understanding of the implications of this delta hedging strategy. They show that a delta hedged portfolio's profit and loss can be broken down into three key elements - profit and loss from gamma (hedging error), profit and loss from Theta or time decay and vega, profit and loss from changing volatility. This gives an equation for daily profit and loss, PL , of

$$PL = \frac{1}{2}\Gamma S^2(\Delta S)^2 + \Theta(\Delta t) + V(\Delta\sigma). \quad (2.19)$$

Now assuming that volatility is constant (the same assumption made by Black and Scholes [1973]) results in

$$PL = \frac{1}{2}\Gamma S^2(\Delta S)^2 + \Theta(\Delta t). \quad (2.20)$$

Bossu *et al.* [2005] provide a derivation showing the relationship between Θ and Γ to be

$$\Theta = -\frac{1}{2}\Gamma S^2\sigma^2. \quad (2.21)$$

Substituting this into Equation 2.20 and rearranging gives

$$PL = \frac{1}{2}\Gamma S^2 \left[\left(\frac{\Delta S}{S} \right)^2 - \sigma^2 \Delta t \right]. \quad (2.22)$$

$\left(\frac{\Delta S}{S}\right)^2$ can be interpreted as the one day variance of the asset price and the second term $\sigma^2 \Delta t$ the daily volatility that the option is valued at. Traders refer to the volatility which is used to price an option as the implied volatility, as it is the volatility implied by the price of an option. Equation 2.22 shows that the profit and loss from a delta hedged portfolio is effectively the difference between realised volatility in the underlying asset and the volatility which was used to price the option at, weighted by the gamma of the option.

It follows that if traders expect realised volatility to be high over the term of the option they will set the implied volatility they use to price options higher and lower if they expect the realised volatility to be lower over the period. As all parameters of option pricing, except the volatility are directly observable it is common practice for traders to simply quote the volatility which they value the option at. Traders, thus, only have control over this volatility variable. As such, the volatility is used to represent risk - a trader who believes there is increased risk in selling an option will simply increase the volatility, and decrease it when there is risk in buying it. Implied volatility is therefore a measure of future expected volatility and option risk - a far cry from simply the standard deviation of the underlying asset prices. This concept is key for understanding the volatility skew.

Chapter 3

The volatility smile problem

In a Black-Scholes world it is expected that when the implied volatilities of different options contracts on the same underlying asset are compared, they would be equal – the model assumes a constant volatility independent of strike price and maturity. In practice however this is not the case. Figure 3.1 shows the implied volatilities of European call options on the S&P index plotted against their corresponding strike prices.

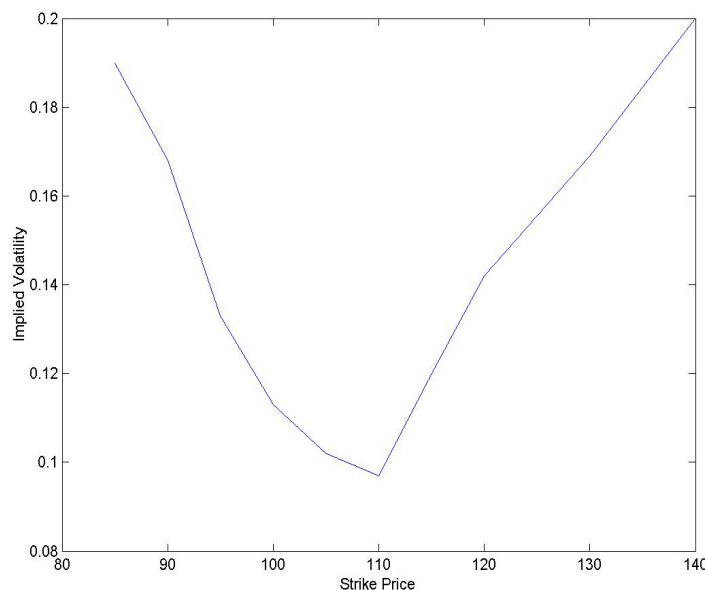


Figure 3.1: The volatility skew of S&P index call options

This characteristic shape, known as the volatility skew or volatility smile, suggests that traders use different implied volatilities for options with different strikes. This has a number of implications. Firstly, as the expected future variance of an underlying assets price should be independent of option strikes, it suggests that the market disagrees with the Black-Scholes model of option risk. Figure 3.1 suggests that option sellers expect higher levels of risk for options, the lower they are struck with respect to the current underlying asset price levels. This phenomenon is not limited to options with differing strikes but, as Figure 3.2 shows, the volatility skew of options with varying maturities also differs. This may be explained by the expectation that future realised variance would certainly change depending on time horizon - something which Black-Scholes does not take into account. This dependence on time, known

as term structure, combines with the skew (dependence on strike price) to form a volatility surface, illustrated in Figure 3.3.

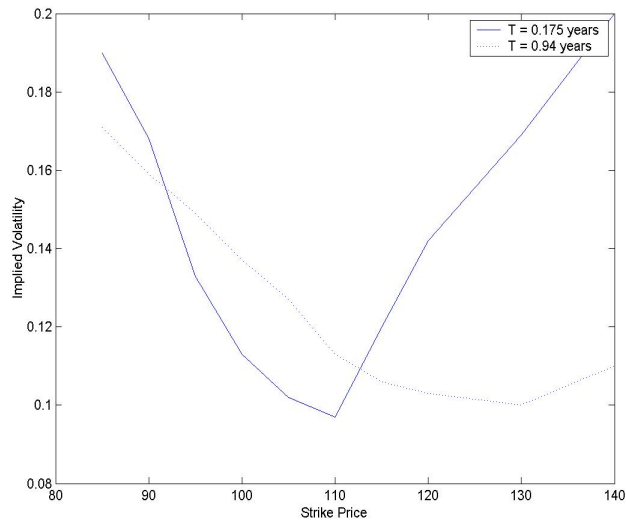


Figure 3.2: The volatility skews of S&P index options with varying maturities

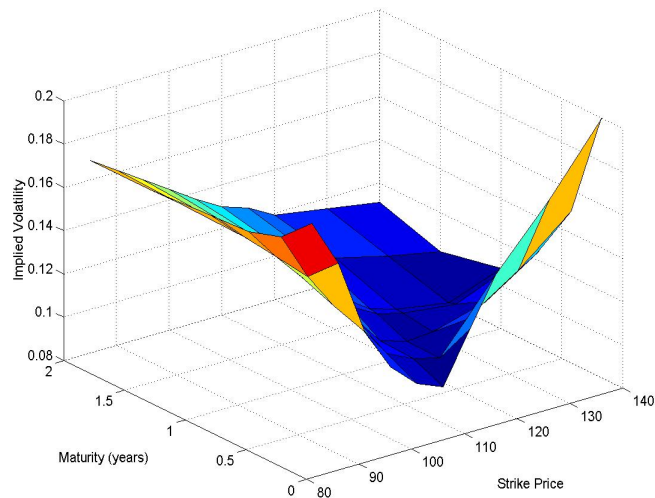


Figure 3.3: The volatility surface of S&P index options

It must be remembered, though, that options are traded in liquid markets and the observations of skew and term structure may also be due to market structure and dynamics. As an example, if there is a very high demand for put options struck lower than the current underlying asset price, economics of supply and demand would dictate that there should be upward price pressure. This chapter will examine both structural flaws in Black-Scholes and the behavioral reasons that give rise to volatility skew and term structure, and show some of the varied approaches used to model this behavior.

3.1 Causes of volatility skew and term structure

3.1.1 Structural flaws in Black-Scholes

Underpinning Black-Scholes are the assumptions made in its derivation. The first of these assumptions is the stochastic process governing asset prices given in Equation 2.1 suggesting lognormal distributions with a constant standard deviation. Investigations, however, have shown that, the assumption that volatility is constant is a flawed one. Volatility is heteroskedastic - volatility itself is volatile with periods of higher volatility and periods of lower volatility [Risk Glossary 2006] and varies stochastically over time. The second problem with the assumed stochastic process is that it does not incorporate the observed leptokurtosis in asset price distributions - taller peaks and fatter tails [Risk Glossary 2006]. Figure 3.4 shows the difference between the lognormal distribution assumed by Black-Scholes, and the more accurate leptokurtic distributions. This can be interpreted to mean that large movements in underlying assets are more likely than Black-Scholes model predicts introducing additional risk to a trader.

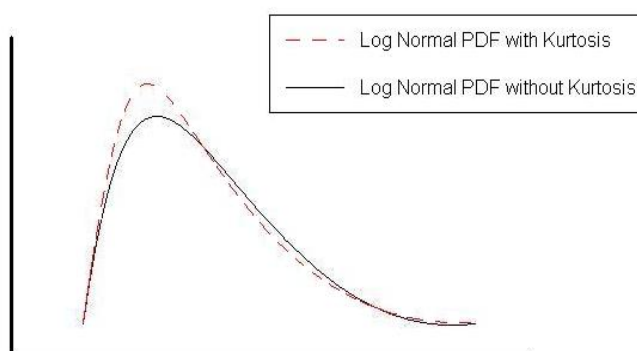


Figure 3.4: Lognormal distributions with and without kurtosis

Another key assumption of Black-Scholes is the ability to hedge an option using a replicating portfolio of the underlying asset (delta hedging). As was shown in Chapter 2, the delta is a continuous function of option value and underlying asset price. This portfolio needs to be adjusted on a continuous basis, which, in practice, is impossible due to transaction costs associated with trading the underlying asset. This inability to constantly maintain the hedged portfolio introduces error into the delta hedged portfolio. This error is magnified by the Gamma, or the change in delta as the underlying asset price moves - something which option traders need to be wary of as stock prices tend to jump around in large discrete moves, resulting in large hedging errors and hence losses. This was highlighted in dramatic fashion in the 1987 stock market crash which shed light on the large model risks involved in option trading.

3.1.2 Behavioral explanations of the skew

Ederington and Guan [2002] suggests that structural problems in the Black-Scholes model can't be the only reason options markets exhibit skew. Rubinstein [1994] shows that the volatility skew only become prominent after the 1987 crash. To properly understand the risk of large movements Table 3.1 shows an example of the effects of a 20% downward jump in the price of an underlying asset on various hedged option portfolios. For each of these portfolios the option trader has sold 100 call options and bought the required units of underlying assets to make sure the portfolios are hedged without error. We assume a volatility of 30%, a risk free rate of 10%, and a maturity of 0.25 years. Assume that the underlying asset price is worth \$100 before the jump and \$80 afterwards. As the option trader has sold the call options they become worth less as the underlying asset price drops and he makes money. However, units in the

underlying asset were purchased to hedge this, and these units lose \$20's in value. Coupled with the hedging error resulting from being unable to instantaneously hedge the position results in a net loss to the option trader. Table 3.1 shows the increase in loss to the options with lower strike. Options traders protect themselves against the increased risk by increasing the implied volatility and hence charging more for options struck lower - producing a skew.

Strike	Gain on options	Loss on hedge	Total loss
\$110	\$74.62	-\$504.60	-\$429.98
\$105	\$183.50	-\$1006.10	-\$822.60
\$100	\$385.31	-\$1659.10	-\$1,273.79
\$95	\$692.60	-\$2271.20	-\$1,578.60

Table 3.1: Loss as a result of a downward jump across various portfolios

The increased fear of crashes after 1987 is a phenomenon known as crashophobia. Not only does this produce skew because of option traders increasing implied volatility but also because it has increased awareness that crashes do happen and thus, there is higher demand for puts which are struck at low prices as protection against crashes. This pressure on supply and demand also pushes up the price giving higher implied volatilities. This gives rise to the characteristic shapes seen in different markets. Option markets where equity indices are the underlying asset usually exhibit skew, as seen in Figure 3.5, because of the need to protect against downward price movements. Foreign exchange markets however can move dramatically in both directions - in any given currency pair either currency is subject to downward shocks, meaning the pair instrument can experience large movements in both directions. These markets exhibit more of a smile like structure seen in Figure 3.6.

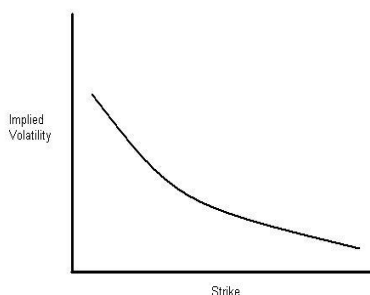


Figure 3.5: Characteristic skew shape of Equity Index Options

3.2 Approaches to modelling the volatility surface

3.2.1 Jump-diffusion models

Merton [1976] was one of the first to provide a framework for valuation of options which can accommodate skew. This was achieved by relaxing the requirement that asset prices move continuously and, instead, provides a model which uses a stochastic process that has discrete jumps in conjunction with continuous changes. This is achieved by defining a new stochastic process given by Equation 3.1 where λ is the average number of discrete jumps per year, k is the average size of those jumps as a percentage of the underlying asset price and dp is a Poisson process:

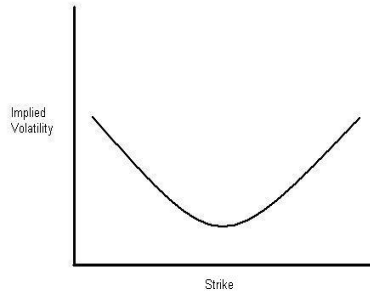


Figure 3.6: Characteristic smile shape of a currency options

$$dS = (r - \lambda k)Sdt + \sigma dz + dp. \quad (3.1)$$

Merton [1976] shows that, for the special case when the size of the jumps are normally distributed the price of an option is given by

$$\sum_{n=0}^{\infty} \left(\frac{e^{-\lambda(1+k)T} (\lambda(1+k)T)^n}{n!} f_n \right) \quad (3.2)$$

where f_n are the corresponding Black-Scholes option prices. Introducing discrete jumps has the effect of introducing leptokurtosis into the distribution. This model is best suited to modeling assets which exhibit a smile rather than skew as both tails of the distribution have increased probability [Hull 2006].

3.2.2 Stochastic volatility

One of the approaches gaining a large following in the financial world is that of stochastic volatility. Stochastic volatility models are two-factor models in that they capture two sources of randomness, the stochastic nature of the underlying asset price and that of the volatility. In general, stochastic volatility models use a stochastic process given in Equation 3.3 where the variance V follows a stochastic process of its own:

$$dS = \mu S dt + \sqrt{V} dz. \quad (3.3)$$

Heston [1993] provides one of the more popular stochastic volatility models. It assumes V follows the process

$$dV = \theta(w - v)dt + \xi\sqrt{V}dB, \quad (3.4)$$

where w is the long term mean of volatility, θ is a measure of the reversion of volatility to w , ξ is the measure of the volatility of volatility or heteroskedascity and dB is a Wiener process which is correlated to dz in Equation with some correlation coefficient. This correlation coefficient is important as it suggests that the sources of randomness which drive both the stock price movements and changes in volatility are correlated.

Similar approaches are taken by Hull and White [1987] and Hagan *et al.* [2002]. The SABR approach of Hagan *et al.* [2002] is quickly becoming a standard approach for modeling volatility skew.

3.2.3 Local volatility models

Dupire [1994] shows that if we have the implied volatility $\sigma(K, t)$ for all possible K and t then we can create a unique process

$$dS = \mu dt + \sigma(S, t)SdW \quad (3.5)$$

where $\sigma(S, t)$ can be uniquely determined from market prices, which, when incorporated into the Black-Scholes model

$$\frac{\partial V}{\partial t} + rS\frac{\partial V}{\partial S} + \frac{1}{2}\sigma(S, t)^2S^2\frac{\partial^2 V}{\partial S^2} = rV \quad (3.6)$$

yields more accurate market prices. In reality however the implied volatility is not available for all possible strikes and maturities and as such the problem of determining $\sigma(K, t)$ from the sparse data becomes ill-posed, which can lead to instability. The result is the reduction of the local volatility surface approach to that of a non-parametric regression problem.

Rubinstein [1994] introduced the implied binomial tree approach, which takes the volatility skew or $\sigma(S)$ into account but suffers because of the failure to incorporate the term structure of volatility into the model. This is addressed in the implied trinomial tree approach of Dupire [1994] but this approach suffers from a lack of robustness because of the ill-posed nature of the problem [Jackson *et al.* 1998].

Lagnado and Osher [1997a] propose an approach to deal with this ill-posed nature. Making use of a technique known as Tykhonov regularization they propose the inverse optimisation problem of finding $\sigma(K, t)$ such that

$$\sum_{i=1}^n (v_i(\sigma(S, t)) - \bar{v}_i)^2 + \lambda \|\nabla \sigma(S, t)\| \quad (3.7)$$

is minimised, where vectors v and \bar{v} are the the calculated price of the option and the observed market price respectively, λ is a constant and n is the number of data points. They use a gradient-descent optimisation method to minimise the function in Equation 3.7 which results in severe limitations. Both Jackson *et al.* [1998] and Coleman *et al.* [1999] highlight the large computational expense of solving this large scale optimisation problem. It is for this reason that both Jackson *et al.* [1998] and Coleman *et al.* [1999] propose the use of splines to fit a smooth surface to the available data, reducing the scale of the optimisation problem significantly.

Chapter 4

Radial basis functions

Nonparametric regression attempts to find a suitable continuous function which approximates the set of known data points. That is, a function $S(x) : \mathfrak{R}^n \mapsto \mathfrak{R}$ is found such that $S(x^i) \approx y_i$ for $i = 1 \dots n$.

One popular approach to constructing such a function is to take a linear combination of basis functions such that:

$$S(x) = \sum_{i=1}^m w_i \varphi_i(x). \quad (4.1)$$

Basis functions of a function space are the functions which can, in linear combination, represent any function in that functional space. This approach to regression essentially reduces the problem down to finding the basis functions $\varphi_i(x)$ and their associated weights, w_i . This thesis focuses on the choice of a specific basis set called radial basis functions to approximate the local volatility function.

4.1 Radial basis functions

Radial basis functions are functions which take the form of Equation 4.2.

$$\varphi(\mathbf{x}, \mathbf{c}) = \varphi(\|\mathbf{x} - \mathbf{c}\|). \quad (4.2)$$

That is, the value of the function depends only on the distance of the point \mathbf{x} from the center point of the function, \mathbf{c} . The distance function is usually defined as the Euclidean norm [Baxter 1992].

There are three common types of radial basis functions – the Gaussian (4.3), the multiquadratic (4.4) and the thin plate spline (4.5):

$$\varphi(\|\mathbf{x} - \mathbf{c}\|) = e^{-\beta\|\mathbf{x} - \mathbf{c}\|^2}, \quad (4.3)$$

$$\varphi(\|\mathbf{x} - \mathbf{c}\|) = (\|\mathbf{x} - \mathbf{c}\|^2 + \beta^2)^{\frac{1}{2}}, \quad (4.4)$$

$$\varphi(\|\mathbf{x} - \mathbf{c}\|) = \|\mathbf{x} - \mathbf{c}\|^2 \log(\|\mathbf{x} - \mathbf{c}\|). \quad (4.5)$$

The choice of radial basis function type is largely problem dependent as each has unique characteristics which may make it more suitable for some problems. The Gaussian function is said to be locally responsive as the value of the function decays quickly to 0 as it gets further from the center, c [Orr 1996]. This behavior is represented in Figure 4.1 which shows a one dimensional Gaussian function with $\beta = 1$ and $c = 0$. Franke [1982] shows that approximations using Gaussian functions is highly dependent on the value of the width parameter β , which needs to be found.

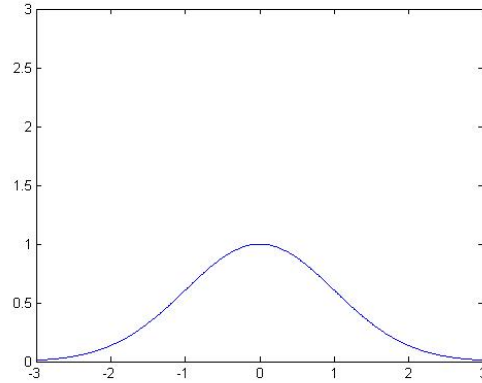


Figure 4.1: A 1D Gaussian function with $c = 0$ and $\beta = 1$

Multiquadratics are globally responsive – their value does not decay to 0 as distance from the center, c , increases [Orr 1996]. This is illustrated in Figure 4.2 which shows a one dimensional multiquadratic function with $\beta = 1$ and $c = 0$. Franke [1982] indicates that approximations using multiquadratic functions are less sensitive to the value of the width parameter β .

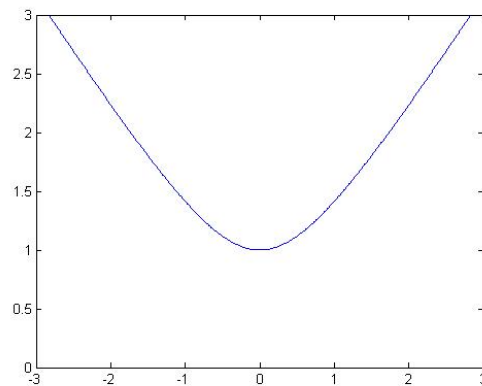


Figure 4.2: A 1D multiquadratic function with $c = 0$ and $\beta = 1$

The thin plate spline function is a commonly used function for two dimensional regression – due mainly to it’s physical interpretation. The function is a solution to the biharmonic equation $(\nabla^2)^2 = 0$ and physically represents a surface which passes through a given set of data points with the minimum amount of “bending energy” – a highly desirable property for smooth surfaces. An additional advantage of the thin plate spline is that there is no width parameter which needs to be found. This reduces the complexity and increases efficiency of the problem significantly.

For the purposes of this research all three of these radial basis function types will be compared and contrasted as to their suitability in finding an accurate approximation of the local volatility surface.

4.2 Approximation using radial basis functions

It is clear from earlier discussion that there are several parameters which need to be found when constructing a linear approximation model. Of key importance is the weight vector, w . The most common approach to finding the weight vector is the method of least squares. In least squares the weight vector is chosen such that it minimises the sum of the squared errors between the known values and those predicted by the model. That is, given the model

$$S(x) = \sum_{j=1}^m w_j \varphi_j(x), \quad (4.6)$$

and known data points

$$(x_i, f_i) \text{ for } i = 1 \dots n, \quad (4.7)$$

w is found to minimise

$$\sum_{i=1}^n [f_i - S(x_i)]^2, \quad (4.8)$$

which, after substitution, gives

$$\sum_{i=1}^n \left[f_i - \sum_{j=1}^m w_j \varphi_j(x_i) \right]^2. \quad (4.9)$$

The problem with this approach is that no assumption is made about the requirements of the function between the known data points. As the system may be over specified this leads to the possibility of over-fitting. Figure 4.3 shows a curve set to a set of data points. Even though it is evident that the data points are more likely to represent a linear relationship, as shown in Figure 4.4, the function in Figure 4.3 may be a better solution to the standard least squares approach.

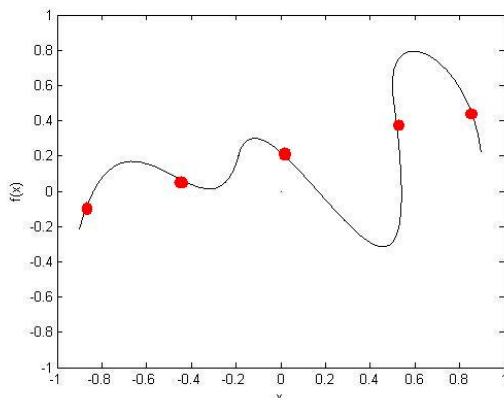


Figure 4.3: A set of data points and an over-fitted function

The problem of overfit can be solved using an approach known as ridge regression developed by Hoerl [1962]. This approach has been shown to be equivalent to Tikhonov regularisation [Tikhonov 1963] which is used by Lagnado and Osher [1997b] in their approach to recovering the local volatility surface.

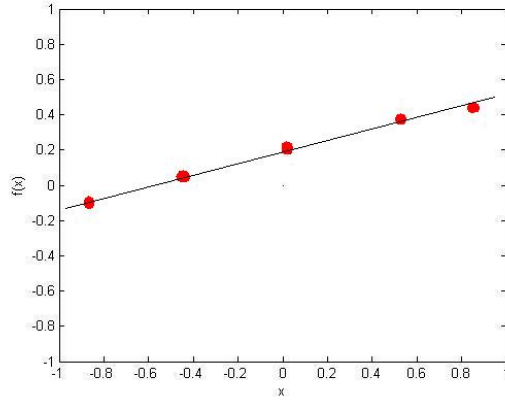


Figure 4.4: A set of data points with the likely, correct, function

The idea is to add an additional term to the regression problem, given in Equation 4.9, which penalises large weights – the effect being the reduction in variance of the surface. This results in the minimisation of Equation 4.10 over \mathbf{w} :

$$\sum_{i=1}^n \left[f_i - \sum_{j=1}^m w_j \varphi_j(x_i) \right]^2 + \lambda \sum_{j=1}^m w_j^2. \quad (4.10)$$

4.2.1 The optimal weight vector

To solve this minimisation problem, the standard approach of setting the first derivative to zero and solving for w , is used [Orr 1996]:

$$\frac{d}{d\mathbf{w}} \left[\sum_{i=1}^n \left(f_i - \sum_{j=1}^m w_j \varphi_j(x_i) \right)^2 + \lambda \sum_{j=1}^m w_j^2 \right] = 0. \quad (4.11)$$

To make the solution of this equation easier the derivative of the j th equation can be found

$$2 \sum_{i=1}^n (f_i - w_j \varphi_j(x_i)) \varphi_j(x_i) + 2\lambda w_j = 0. \quad (4.12)$$

Rearranging results in

$$\sum_{i=1}^n f_i \varphi_j(x_i) - \sum_{i=1}^n w_j \varphi_j(x_i) \varphi_j(x_i) + \lambda w_j = 0. \quad (4.13)$$

This implies

$$\begin{aligned} \varphi_1^T \mathbf{f} - \varphi_1^T \varphi_1 w_1 + \lambda w_1 &= 0, \\ \varphi_2^T \mathbf{f} - \varphi_2^T \varphi_2 w_2 + \lambda w_2 &= 0, \\ &\vdots \\ \varphi_m^T \mathbf{f} - \varphi_m^T \varphi_m w_m + \lambda w_m &= 0. \end{aligned} \quad (4.14)$$

With

$$\Phi = \begin{bmatrix} \varphi_1(x_1) & \varphi_2(x_1) & \cdots & \varphi_m(x_1) \\ \varphi_1(x_2) & \varphi_2(x_2) & \cdots & \varphi_m(x_2) \\ \varphi_1(x_3) & \varphi_2(x_3) & \cdots & \varphi_m(x_3) \\ \vdots & & \ddots & \vdots \\ \varphi_1(x_n) & \varphi_2(x_n) & \cdots & \varphi_m(x_n) \end{bmatrix} \quad (4.15)$$

$$\varphi_j = \begin{bmatrix} \varphi_j(x_1) \\ \varphi_j(x_2) \\ \varphi_j(x_3) \\ \vdots \\ \varphi_j(x_n) \end{bmatrix}, \mathbf{f} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_n \end{bmatrix} \text{ and } \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix}$$

we get

$$\begin{bmatrix} \varphi_1^T \\ \varphi_2^T \\ \varphi_3^T \\ \vdots \\ \varphi_m^T \end{bmatrix} \mathbf{f} - \Phi^T \Phi \mathbf{w} + \lambda \mathbf{w} = 0 \quad (4.16)$$

given that $\varphi_i \varphi_j = 0, i \neq j$. Rearranging results in

$$\Phi^T \mathbf{f} = \Phi^T \Phi \mathbf{w} - \lambda \mathbf{w} = (\Phi^T \Phi - \lambda I) \mathbf{w}. \quad (4.17)$$

Solving for \mathbf{w} , the optimal weight vector, gives

$$\mathbf{w} = (\Phi^T \Phi - \lambda I_m)^{-1} \Phi^T \mathbf{f}. \quad (4.18)$$

4.3 Model specification

Finding the optimal weight vector depends on knowledge of the set of functions, their parameters and the regularisation parameter λ . The non-linear relationship between \mathbf{w} and the model specifying set leaves two choices - impose a set of functions, their parameters and λ on the system using heuristics and knowledge of the problem, or, use a non-linear optimisation algorithm to find the optimal set.

4.3.1 The heuristic approach

Taking a heuristic approach requires the development of an intuitive feeling of how model parameters affect the regression, and a clear understanding of the geometry of the specific regression problem being solved.

The first decision that needs to be taken is the choice of radial basis function type. Intuitively the thin plate spline seems to be the natural choice as it's physical interpretation matches the problem of volatility surface reconstruction. Franke [1982] finds that thin plate splines perform well, showing minimum error

and significant smoothness, on 2 dimensional surface reconstruction problems. The additional advantage of thin plate splines is that it requires one less parameter to be found - there is no width parameter β .

Franke [1982] also found the multi quadratic to be successful at interpolating 2 dimensional surfaces. Both Baxter [1992] and Franke [1982] shows the multi-quadratics relative insensitivity to the choice of centers, \mathbf{c} .

This is in contrast to the most popular of radial basis function types - the gaussian. Franke [1982] finds it very sensitive to the choice of \mathbf{c} , a view which is supported by Baxter [1992]. The problem with the Gaussian is that it is bounded (it decays to 0 very quickly) and, as described earlier, only locally responsive while both thin plate splines and multi-quadratics are unbounded and globally responsive. This means that it may be necessary to have a large set of Gaussian functions to adequately cover and describe the full problem domain.

Part of this research, however, is to investigate which of these functions is more appropriate for recovery of the volatility surface and, as such, all will be tested and compared.

Chapter 5

Reconstructing the local volatility surface using radial basis functions

The approach taken in this dissertation follows that of Coleman *et al.* [1999] closely, except radial basis functions are used instead of splines to represent the local volatility function. That is, a function

$$\sigma(S, t) = \sum_{j=1}^m w_j h_j(\mathbf{x}) \quad (5.1)$$

with h a set of m radial basis functions and w_j a set of corresponding weights, is found that satisfies

$$\min_{\sigma(S,t)} \sum_{i=1}^n [v_i(\sigma(S, t)) - v_{BS}(f_i)]^2. \quad (5.2)$$

$v_{BS}(\mathbf{f})$ in Equation 5.2 is the set of n observed Black-Scholes prices and $v_i(\sigma(S, t))$ is the option price at S and t given by

$$\frac{\partial V}{\partial t} + rS \frac{\partial V}{\partial S} + \frac{1}{2} \sigma(S, t)^2 S^2 \frac{\partial^2 V}{\partial S^2} = rV. \quad (5.3)$$

Should the generated surfaces suffer from overfit and become unstable Tikhonov regularisation can be used resulting in the problem

$$\min_{\sigma(S,t)} \sum_{i=1}^n [v_i(\sigma(S, t)) - f_i]^2 + \lambda \sum_{j=1}^m w_j. \quad (5.4)$$

Equations 5.2 and 5.4 are non-linear least-squares minimisation problems which pose significant computational challenges.

5.1 Simplifying assumptions

A number of simplifying assumptions and heuristics are used to reduce the scale of the optimisation presented in Equation 5.2. Firstly we assume that the set of m radial basis functions, \mathbf{h} in Equation 5.1 are known, and that their positioning (centers) are fixed. A range of function sets are used to try and establish the best choice for the local volatility problem. The result of this assumption is to reduce

the optimisation problem to find the optimal weight vector \mathbf{w} . Secondly, we assume that, if Tikhonov regularisation is needed, the regularisation parameter, λ , is chosen using trial and error - a procedure which finds the optimal λ given the non-linear nature of the problem is out of the scope of this thesis.

5.2 The procedure

An outline for recovering the local volatility function is presented below. Procedures for each of the key steps will be covered in detail.

Given the function set \mathbf{h} and observed market data \mathbf{f} we need to:

- Find an initial weight vector \mathbf{w}_0

Then iteratively,

- Evaluate the cost function given in Equation 5.2
- Using an optimisation algorithm update the optimal weight vector
- Repeat until convergence

5.2.1 The initial weight vector

Experimentation found that the whole procedure was very sensitive to initial choice of weight vector. To solve this and generate a surface of realistic volatilities a simple method is used. We find an initial \mathbf{w}_0 such that the implied volatilities of observed market data \mathbf{f} are interpolated using the appropriate radial basis function set \mathbf{h} . That is, we solve

$$\min_{\mathbf{w}_0} \sum_{i=1}^n \left[f_i - \sum_{j=1}^m w_{0j} h_j(x_i) \right]^2 + \lambda \sum_{j=1}^m w_{0j}^2 \quad (5.5)$$

This provides an initial weight vector w_0 that gives a volatility surface that is reasonable. Equation 5.5 is a linear least squares optimisation problem, and as such is solved using methods set out in Chapter 4.

5.2.2 Evaluating the cost function

Key to the evaluation of the cost function

$$CF = \min_{\sigma(S,t)} \sum_{i=1}^p [v_i(\sigma(S,t)) - f_i]^2 + \lambda \sum_{i=1}^n w_i \quad (5.6)$$

is the solution of option prices $v_i(\sigma(S,t))$ for each observation point. This is done by solving

$$\frac{\partial v}{\partial t} + rS \frac{\partial v}{\partial S} + \frac{1}{2} \sigma(S,t)^2 S^2 \frac{\partial^2 v}{\partial S^2} = rv \quad (5.7)$$

This research takes a finite difference approach, in particular, the Crank-Nicolson method, to solve Equation 5.7. This was chosen due to its relative simplicity, and, its good convergence properties. The Crank-Nicolson scheme has an error term of $O(h^2)$ [Chistensen and Munk 2004]. To achieve this the solution

space is first discretised into N spatial and M temporal points of size k and h respectively. That is, the time interval $[0; T]$ is discretised by

$$t_i = ih \forall i \in [0 \dots M] \quad (5.8)$$

and

the space interval $[0; S_{max}]$ is discretised by

$$S_j = jk \forall j \in [0 \dots N] \quad (5.9)$$

Using central differences we can define

$$\frac{\partial v}{\partial S} \simeq \frac{v(S+k, t) - v(S-k, t)}{2k} = \frac{v_{i,j+1} - v_{i,j-1}}{2k} \quad (5.10)$$

and

$$\frac{\partial^2 v}{\partial S^2} \simeq \frac{v(S+k, t) - 2v(S, t) + v(S-k, t)}{k^2} = \frac{v_{i,j+1} - 2v_{i,j} + v_{i,j-1}}{k^2} \quad (5.11)$$

Crank-Nicolson uses an average of the forward and backward differences for the time derivative $\frac{\partial v}{\partial t}$. The forward difference is given by

$$\frac{\partial v}{\partial t} \simeq \frac{v(S, t+h) - v(S, t)}{h} = \frac{v_{i+1,j} - v_{i,j}}{h} \quad (5.12)$$

and the backward difference is given by

$$\frac{\partial v}{\partial t} \simeq \frac{v(S, t) - v(S, t-h)}{h} = \frac{v_{i,j} - v_{i-1,j}}{h} \quad (5.13)$$

Substituting these approximations into Equation 5.7 results in

$$a_{i,j}v_{i,j-1} + b_{i,j}v_{i,j} + c_{i,j}v_{i,j+1} = -a_{i,j}v_{i+1,j-1} + (4 - b_{i,j})v_{i+1,j} - c_{i,j}v_{i+1,j+1}, \quad (5.14)$$

where

$$\begin{aligned} a_{i,j} &= \frac{1}{2}h \left[rj - \sigma^2(S_j, t_i)j^2 \right], \\ b_{i,j} &= 2 + h \left[r + \sigma^2(S_j, t_i)j^2 \right], \\ c_{i,j} &= -\frac{1}{2}h \left[rj + \sigma^2(S_j, t_i)j^2 \right]. \end{aligned} \quad (5.15)$$

Equation 5.14 can be expressed in matrix notation as

$$\begin{bmatrix} b_{i,1} & c_{i,1} & 0 & 0 & 0 & \cdots & 0 \\ a_{i,2} & b_{i,2} & c_{i,2} & 0 & 0 & \cdots & 0 \\ 0 & a_{i,2} & b_{i,2} & c_{i,2} & 0 & \cdots & 0 \\ \vdots & & & \ddots & & & \vdots \\ \vdots & & & & \ddots & & \vdots \\ 0 & \cdots & 0 & 0 & a_{i,M-2} & b_{i,M-2} & c_{i,M-2} \\ 0 & \cdots & 0 & 0 & 0 & a_{i,M-1} & b_{i,M-1} \end{bmatrix} \begin{bmatrix} v_{i,1} \\ v_{i,2} \\ v_{i,3} \\ \vdots \\ \vdots \\ v_{i,M-2} \\ v_{i,M-1} \end{bmatrix} = \begin{bmatrix} d_{i+1,1} \\ d_{i+1,2} \\ d_{i+1,3} \\ \vdots \\ \vdots \\ d_{i+1,M-2} \\ d_{i+1,M-1} \end{bmatrix} \quad (5.16)$$

where

$$\begin{aligned}
d_{i+1,j} &= v_{i+1,j} \text{ for } j = 2, 3, \dots, M - 2, \\
d_{i+1,1} &= v_{i+1,1} - a_{i,1}v_{i,0}, \\
d_{i+1,M-1} &= v_{i+1,M-1} - c_{M-1}v_{i,M}.
\end{aligned} \tag{5.17}$$

From this, it can be seen that the solution of Equation 5.7, and hence $v(\sigma(S, t))$, requires the solution of $N, M - 1 \times M - 1$ systems of equations. As the cost function requires n solutions, this poses a significant computational challenge, and efficient implementation is key.

5.3 Implementation considerations

A naive evaluation of the cost function would require nN inversions of an $(M - 1) \times (M - 1)$ matrix. This, however, can be reduced in a number of ways. Firstly, a solution of Equation 5.16 yields the value of an option for a given strike, over a number of different underlying prices and times. This means that we only need to generate a solution for each unique strike in our cost function and then use a simple table lookup to retrieve the correct price.

Secondly, and more importantly, we can take advantage of the tridiagonal structure of the coefficient matrix of Equation 5.16. The Thomas algorithm provides a solution which is linear in run time complexity. Given the system

$$\begin{bmatrix} b_1 & c_1 & \cdots & 0 \\ a_2 & b_2 & c_2 & \vdots \\ \vdots & & \ddots & \\ 0 & \cdots & a_n & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} \tag{5.18}$$

the solution matrix \mathbf{x} is given by

$$\begin{aligned}
x_n &= d'_n \\
x_i &= d'_n - c'_n x_{i+1} \text{ for } i = n - 1, n - 2, \dots, 1
\end{aligned} \tag{5.19}$$

where

$$\begin{aligned}
c'_i &= \begin{cases} \frac{c_1}{b_1} & i = 1, \\ \frac{c_i}{b_i - c'_{i-1}a_i} & i = \{2, 3, \dots, n - 1\}, \end{cases} \\
d'_i &= \begin{cases} \frac{d_1}{b_1} & i = 1, \\ \frac{d_i - d'_{i-1}a_i}{b_i - c'_{i-1}a_i} & i = \{2, 3, \dots, n - 1\}. \end{cases}
\end{aligned} \tag{5.20}$$

5.4 The optimisation algorithms

5.4.1 The Nelder-Mead Simplex algorithm

The basic idea behind the Nelder-Mead, or Simplex algorithm is the creation and evolution of a simplex of points on the cost function surface to find the minimum. A simplex is a polotype with $n + 1$ vertices

in n dimensions. The vertices of this polytope are evaluated and adjusted using several simple rules depending on the geometry of the function being searched.

The first stage of the Nelder-Mead algorithm is creating the simplex. Suppose the function $f(\mathbf{x})$ over \mathbf{x} where \mathbf{x} is a vector of length n . The Nelder-Mead algorithm generates a simplex of $n + 1$ points and orders them such that

$$f(x_1) \leq f(x_2) \leq \dots \leq f(x_{n+1}). \quad (5.21)$$

The simplex is then evolved using a simple reflection operation. A new vertex, x_r , is created by

$$x_r = x_0 + \alpha(x_0 - x_{n+1}) \quad (5.22)$$

where x_0 is the centre of gravity of the n best points and α is a reflection coefficient, usually chosen to be 1. $f(x_r)$ is then evaluated, and, if it is found to be better than $f(x_{n+1})$ but worse than $f(x_1)$, i.e., $f(x_1) \leq f(x_r) \leq f(x_{n+1})$ then $f(x_r)$ replaces $f(x_{n+1})$ and the process is repeated. If, however, $f(x_r) \leq f(x_1)$, an expansion operation is performed, and if $f(x_{n+1}) \leq f(x_r)$, a contraction operation is performed.

The intuitive idea behind an expansion is that, as a good solution was found along the direction of the reflection, the search should be continued along that line in hope of improving further. The expansion operation is thus given by

$$x_e = x_0 + \gamma(x_0 - x_{n+1}) \quad (5.23)$$

where γ is the expansion parameter, which is usually chosen to be 2. If $f(x_e) > f(x_r)$, a new simplex is created by replacing $f(x_{n+1})$ by $f(x_e)$ and the whole process is repeated on this new simplex.

Contraction assumes that the algorithm has stepped over a good solution, and needs to shorten the step to accommodate. The contraction operation is given by

$$x_c = x_0 + \rho(x_0 - x_{n+1}) \quad (5.24)$$

where ρ is the contraction coefficient, usually given by $\frac{1}{2}$. Again, this is evaluated against $f(x_r)$ and replaces it in the simplex if better.

Finally if all of these fail to produce a better point, the whole simplex is shrunk by

$$x_i = x_1 + \delta(x_i - x_1) \quad \forall i \in \{2, 3, \dots, n + 1\} \quad (5.25)$$

with the shrink coefficient, δ . The algorithm repeated on this new simplex. The whole process is repeated until convergence is achieved.

5.4.2 Trust region optimisation

Trust region methods, instead of optimising over a complex cost function $f(x)$, approximate the function over a region around a point x (called the trust region) and attempt to find a step \mathbf{s} such that $f(\mathbf{x} + \mathbf{s}) < f(\mathbf{x})$ to minimise this new approximated function. Traditionally, the approximated function, q is chosen to be the first two terms of a Taylor series expansion around x . Thus we try to find a step \mathbf{s} such that

$$\begin{aligned}
q(\mathbf{s}) &\approx \mathbf{s}^T \mathbf{g} + \frac{1}{2} \mathbf{s}^T H \mathbf{s} \\
\text{subject to } &\|D\mathbf{s}\| \leq \Delta
\end{aligned}
\tag{5.26}$$

is minimized, where \mathbf{g} is the gradient of f at \mathbf{x} , H is the Hessian matrix, D is a diagonal scaling matrix and Δ is a positive scalar. The function being minimized is quadratic, which is easily minimized. A further enhancement can be made by using an observation made by Byrd *et al.* [1988] that optimising over only 2 dimensions of \mathbf{s} is often sufficient to arrive at the minimum. In Byrd *et al.* [1988] and Mathworks [2008] s_1 is chosen to be the direction of $-\mathbf{g}$ and s_2 is chosen to be the solution to $H.s_2 = -\mathbf{g}$. The result of these approximations is to reduce the minimization problem to a two-dimensional quadratic approximation over a region. Once \mathbf{s} has been found $f(\mathbf{x} + \mathbf{s})$ can be evaluated and if it is shown to be a better solution, \mathbf{x} is updated to be $\mathbf{x} + \mathbf{s}$.

The size of the trust region about \mathbf{x} is controlled through Δ . The implementation used in this research uses the Preconditioned Conjugate Gradient approach of Coleman and Verma [1998].

5.4.3 Implementation of the optimisation algorithms

For purposes of efficiency and simplicity it was decided that implementations in the Matlab Optimisation Toolbox would be used. A full description of the exact implementation and specification of algorithms and heuristics can be found in Mathworks [2008]. Tolerances for the optimisation algorithm were set at 10^{-8} for the change in function value. In addition to this there was a limit of 10000 iterations set to prevent considerable runtimes.

Chapter 6

Experiments and results

The methods developed in Chapter 5 are tested on two test problems - one with an analytical solution and the other a practical, real-world example. This chapter presents results from both these problems. For comparison purposes, we use the same test problems presented in Coleman *et al.* [1999], Lagnado and Osher [1997a] and Kim *et al.* [2006]. Again for consistency several measures of performance are used. Firstly, the average absolute error at each of the n known data points in pricing given by

$$AverageError = \frac{1}{n} \sum_{i=1}^n |v_i(\sigma) - f_i|, \quad (6.1)$$

where $v(\sigma)$ is the price at data point i and f_i is the observed Black-Scholes price. The maximum absolute error observed at the data points (Equation 6.4) is also given:

$$MaxError = Max (|v_i(\sigma) - f_i|), \forall i = 1 \dots n. \quad (6.2)$$

Relative errors are given by

$$AverageError = \frac{1}{n} \sum_{i=1}^n \frac{|v_i(\sigma) - f_i|}{|f_i|}, \quad (6.3)$$

and

$$MaxError = Max \left(\frac{|v_i(\sigma) - f_i|}{|f_i|} \right), \forall i = 1 \dots n. \quad (6.4)$$

Examination of the behavior of the key Greeks are presented to give an insight into the stability of the approach, along with a simple visual inspection of the reconstructed surface to judge smoothness. For purposes of performance evaluation the number of optimisation algorithm iterations is presented, and, given the rough complexity analysis presented in Chapter 5, the number of calls to the cost function.

6.1 A problem with an analytical solution

Cox and Ross [1976] shows that if an underlying asset follows the process

$$dS = \mu dt + \sigma(S, t) S dW \quad (6.5)$$

where

$$\sigma(S, t) = \sigma_0 S^{\frac{\beta-2}{2}}, \quad (6.6)$$

then the value of a call option on the underlying asset is given by

$$C = S_t e^{-rt} Q\left(2y, 2 + \frac{2}{2-\beta}, 2x\right) - K e^{-rt} \left(1 - Q\left(2x, \frac{2}{2-\beta}, 2y\right)\right) \quad (6.7)$$

where

$$\begin{aligned} y &= \bar{k} k^{2-\beta}, \\ x &= \bar{k} S_t^{2-\beta} e^{(r-1)(2-\beta)t}, \\ \bar{k} &= \frac{2r}{\sigma_0^2(2-\beta)[e^{r(2-\beta)t} - 1]}, \end{aligned} \quad (6.8)$$

and $Q(z; v, k)$ is the complimentary noncentral Chi-square distribution function.

Keeping with Lagnado and Osher [1997a] and Coleman *et al.* [1999] we set $\sigma_0 = 15$ and $\beta = 0$ resulting in the local volatility function

$$\sigma(S, t) = \frac{15}{S}. \quad (6.9)$$

Using this, option values are calculated on the strikes $K = [80; 84; 88; 92; 96; 100; 104; 108; 112; 116; 120]$ and maturities $T = [0; 1]$. An initial underlying price of 100 and a risk free rate of 0.05 is used. Approximate surfaces using the various function sets are generated such that $\sigma(K, T) = 0.15$ to get initial weight vectors with centres placed at each of the observation points.

The radial basis function approach is then run using discretisation parameters of $h = 0.01$ and $k = 1$. Results using the Nelder-Mead optimisation, with Gaussian, multiquadratic and thin plate spline function sets are presented in Table 6.1.

Radial Basis Function	Average Absolute Error	Maximum Absolute Error	Average Relative Error	Maximum Relative Error	Iterations	Cost Function Evaluations	Lambda
Gaussian	1.1779	2.2212	0.3771	0.9972	5850	7719	10
Multi quadratic	0.0043	0.0085	0.0031	0.0312	8093	10000	0.2
Thin plate spline	0.0011	0.0040	0.0012	0.0175	3659	4698	0.01

Table 6.1: Summarised results using the Nelder-Mead algorithm

It is clear from these results that the thin plate spline performs well closely followed by the multiquadratic while the Gaussian performs poorly. These results are backed up by Figures 6.2, 6.3 and 6.4 which show smooth well defined surfaces which closely resemble the known volatility function (given in Figure 6.1) for both the thin plate spline and the multiquadratic and a very unstable surface for the Gaussian.

The problem is rerun using the Trust Region optimisation method. Results are presented in Table 6.2.

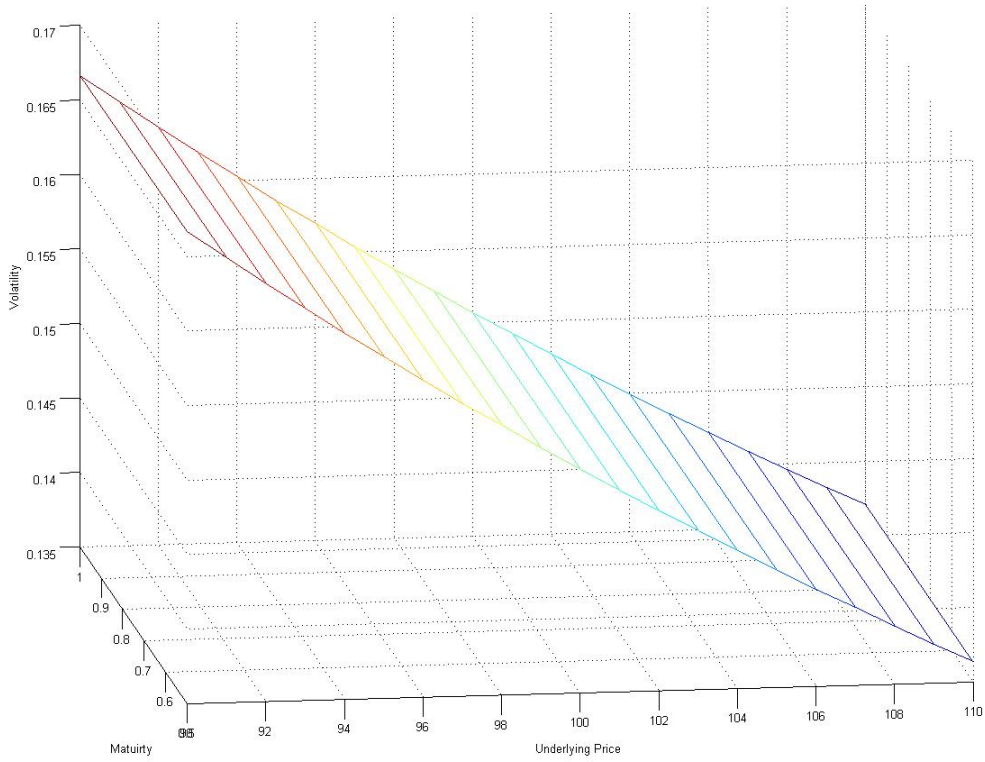


Figure 6.1: Local volatility surface defined by $\sigma(S, t) = \frac{15}{S}$

Radial Basis Function	Average Absolute Error	Maximum Absolute Error	Average Relative Error	Maximum Relative Error	Iterations	Cost Function Evaluations	Lambda
Gaussian	0.8529	1.7064	0.3024	0.9999	52	1219	10
Multi quadratic	0.0007	0.0052	0.0012	0.0228	21	506	0.2
Thin plate spline	0.0012	0.0038	0.0012	0.0165	5	138	0.01

Table 6.2: Summarised results using the Trust-Region algorithm

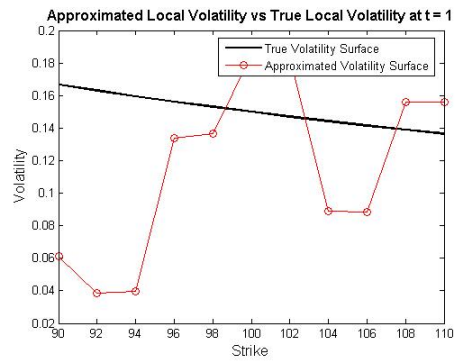
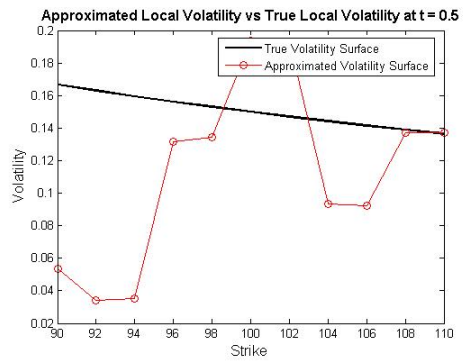
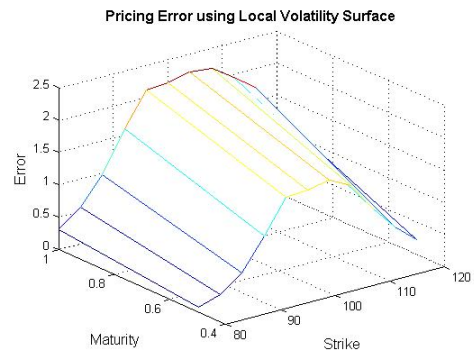
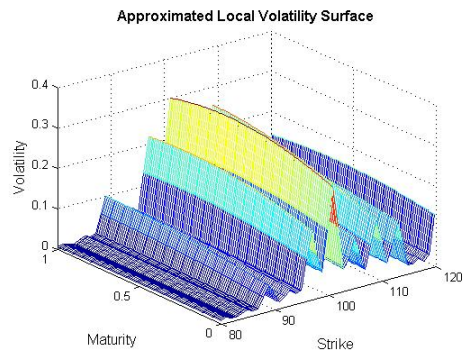


Figure 6.2: Reconstructed local volatility surface using a Gaussian function set and the Nelder-Mead optimisation method

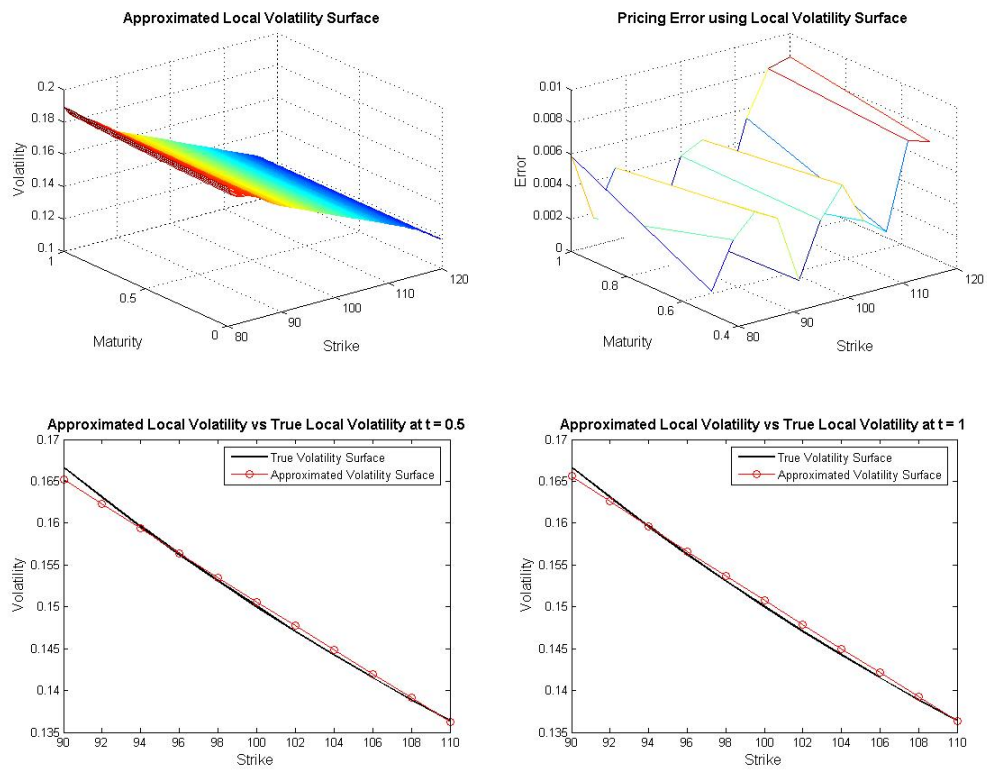


Figure 6.3: Reconstructed local volatility surface using a multi quadratic function set and the Nelder-Mead optimisation method

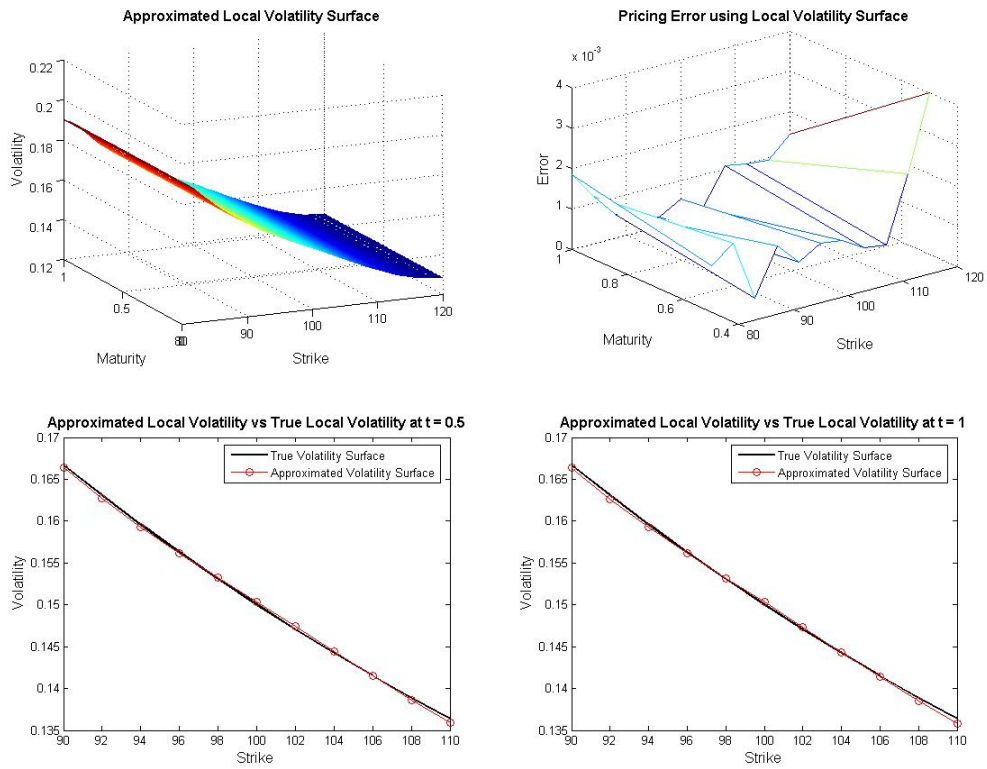


Figure 6.4: Reconstructed local volatility surface using a thin plate spline function set and the Nelder-Mead optimisation method

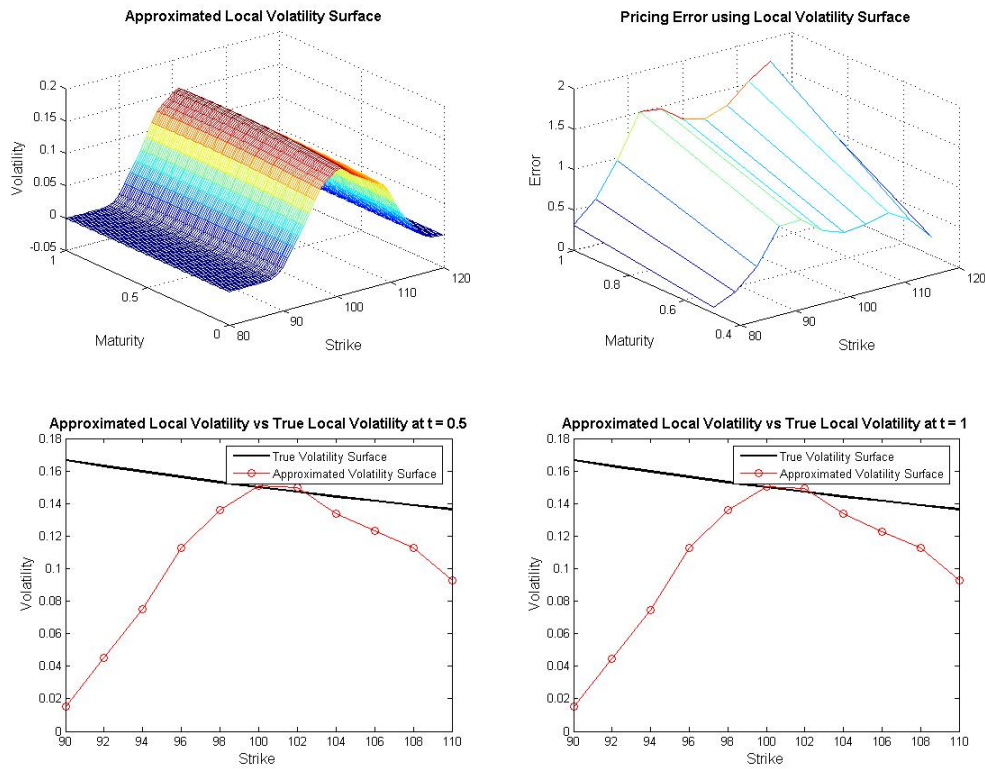


Figure 6.5: Reconstructed local volatility surface using a Gaussian function set and the Trust Region optimisation method

Again, the thin plate spline and multiquadratics are the best performers, showing very accurate results, with the Gaussian a distant third. The respective surfaces are given in Figures 6.5, 6.6 and 6.7. Significantly, the accuracy has been increased and the number of function evaluations has decreased, showing that the trust region method achieves faster convergence, and, as expected is more suited to this problem.

6.1.1 Stability of the Greeks

As discussed in earlier chapters, the Greeks, or sensitivities of the option to changes in various parameters are of key importance to traders and risk managers. It is therefore essential that any model that is used for option pricing and risk investigation be able to produce stable and accurate sensitivities. Figure 6.8 shows a comparison of four of the key Greeks: delta, gamma, vega and theta using the local volatility model generated using the thin plate spline function set and Trust Region optimisation with the Greeks of the analytical solution. In both instances, Greeks were determined using central differences. This shows how accurately the sensitivities are reproduced by the approach taken in this thesis, and further reinforces the radial basis function approach a valid one.

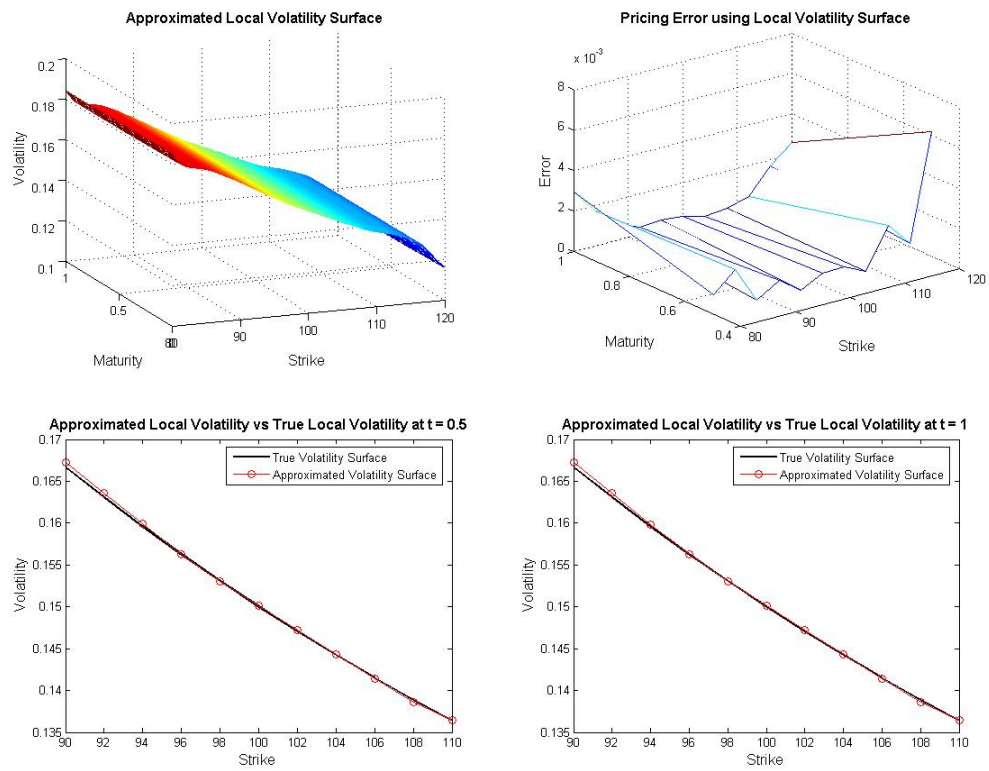


Figure 6.6: Reconstructed local volatility surface using a multi quadratic function set and the Trust Region optimisation method

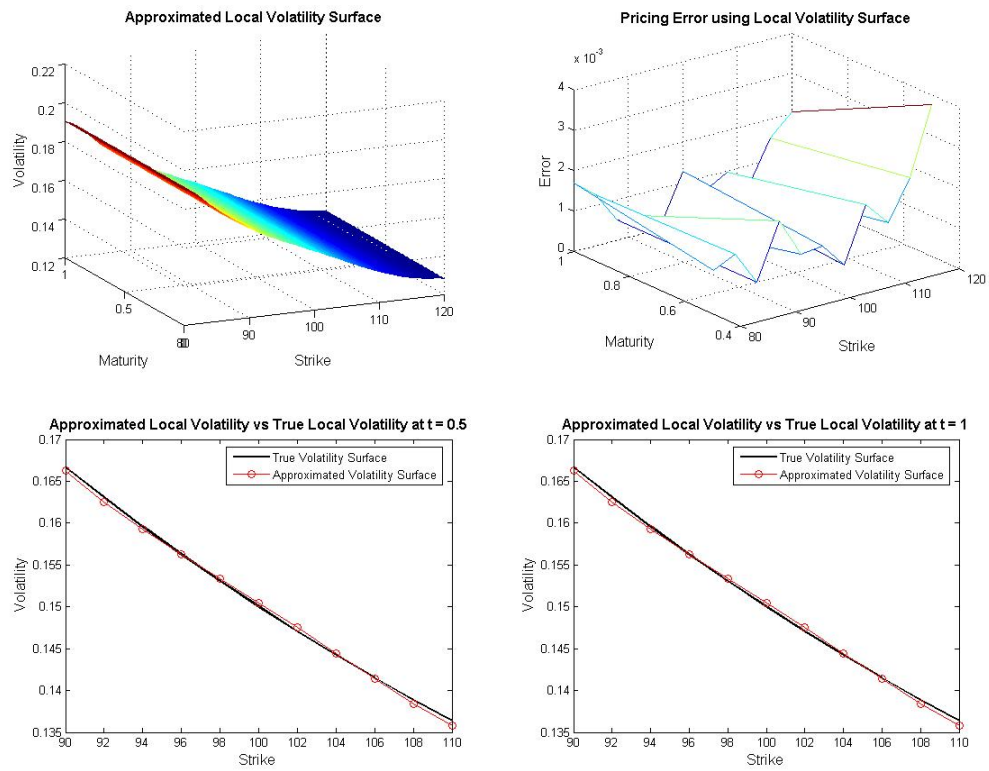
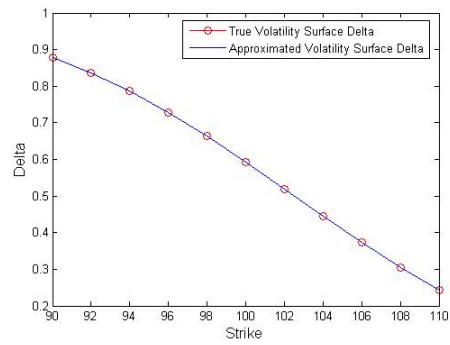
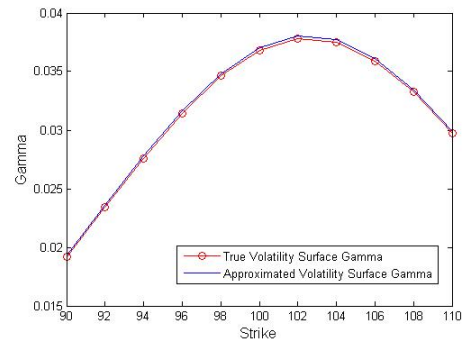


Figure 6.7: Reconstructed local volatility surface using a thin plate spline function set and the Trust Region optimisation method

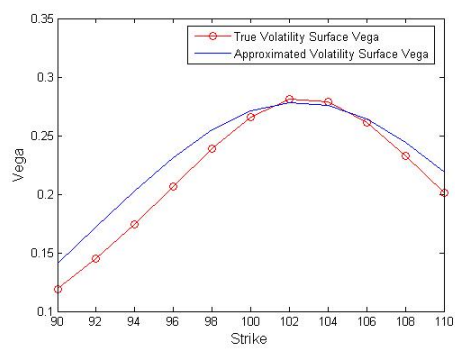
Approximated Local Volatility Delta vs True Local Volatility Delta



Approximated Local Volatility Gamma vs True Local Volatility Gamma



Approximated Local Volatility Vega vs True Local Volatility Vega



Approximated Local Volatility Theta vs True Local Volatility Theta

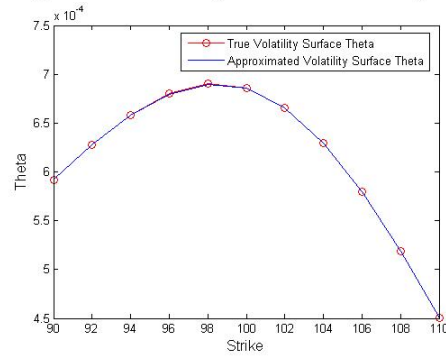


Figure 6.8: A comparison of the Greeks

6.2 A Real-world problem: Reconstructing the volatility surface of S&P 500 index options

Following Coleman *et al.* [1999] the practical problem of reconstructing the volatility surface of options on the S&P 500 index is tackled. Observed implied volatilities, from October 2005, are given in Table 6.3. An index level (underlying asset price) of \$590 and a risk free rate of 6% is used.

Maturity (in years)	Strike									
	\$501.5	\$531	\$560.5	\$590	\$619.5	\$649	\$678.5	\$708	\$767	\$826
0.175	0.190	0.168	0.133	0.113	0.102	0.097	0.120	0.142	0.169	0.200
0.425	0.177	0.155	0.138	0.125	0.109	0.103	0.100	0.114	0.130	0.150
0.695	0.172	0.157	0.144	0.133	0.118	0.104	0.100	0.101	0.108	0.124
0.94	0.171	0.159	0.149	0.137	0.127	0.113	0.106	0.103	0.100	0.110
1	0.171	0.159	0.150	0.138	0.128	0.115	0.107	0.103	0.099	0.108
1.5	0.169	0.160	0.151	0.142	0.133	0.124	0.119	0.113	0.107	0.102
2	0.169	0.161	0.153	0.145	0.137	0.130	0.126	0.119	0.115	0.111

Table 6.3: Implied volatilities for the S&P 500 index during October 2005

This problem is first attempted using a subset of the known volatilities at the points $t = [0.175; 0.695; 1; 2]$ and $S = [531; 590; 649; 708; 826]$. The discretisation parameters used for the Crank-Nicholson scheme are $h = 0.01$ and $k = 1$. This results in 5 Crank-Nicholson solutions on the grid size 200×2478 in the worst case, per evaluation of the cost function.

Results of generation of the initial weight vector w using the heuristic described in Chapter 5 are shown for Gaussian, multi quadratic and thin plate spline function sets in Figures 6.9, 6.10 and 6.11 respectively. The figures show that the multi quadratic appears to provide the most stable initial condition.

Optimisation was first performed using the Simplex algorithm described in Chapter 5. Even though the multi quadratic has the best initial solution, the thin plate spline outperforms it in terms of both accuracy and performance. The Gaussian function performs significantly more poorly than either multi quadratics or thin plate splines in both accuracy and performance - it showed the highest error and took the longest to converge. Table 6.4 summaries the results. Figures 6.12, 6.13 and 6.14 show the generated local volatility surfaces of the Gaussian, multi quadratic and thin plate splines respectively. Visual inspection reveals that the surfaces produced by the multi quadratic and thin plate spline function sets are smooth, and similar with some noticable differences at the boundaries. The Gaussian surface appears unstable - something backed up by the poor performance. Following Coleman *et al.* [1999] a constant volatility approach is also tested where the volatility is set to the average of the observed volatilities and prices generated using the Crank-Nicholson scheme. The average pricing error for the constant volatility was 2.1772 with a maximum error of 5.3871. All three function sets in our example are more accurate than the constant volatility approach.

Radial Basis Function	Average Absolute Error	Maximum Absolute Error	Average Relative Error	Maximum Relative Error	Iterations	Cost Function Evaluations	Lambda
Gaussian	0.5420	2.3598	0.22701	1	7939	10000	0
Multi quadratic	0.1797	0.7816	0.2045	1	7927	10000	0
Thin plate spline	0.1430	0.5573	0.1741	1	7861	10000	0

Table 6.4: Summarised results using the Nelder-Mead algorithm

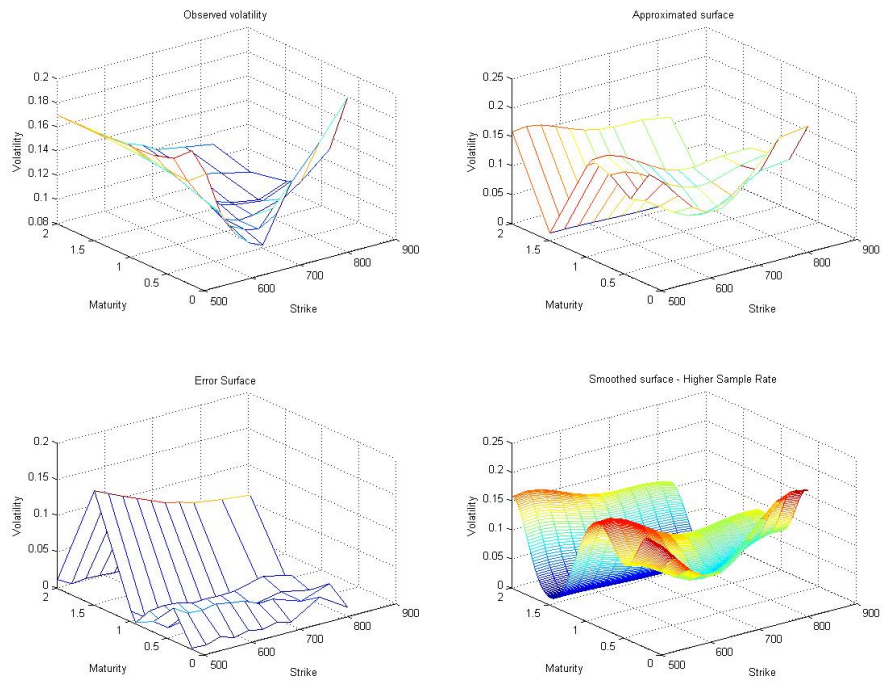


Figure 6.9: Interpolation results of the initial weight vector using a set of Gaussian functions

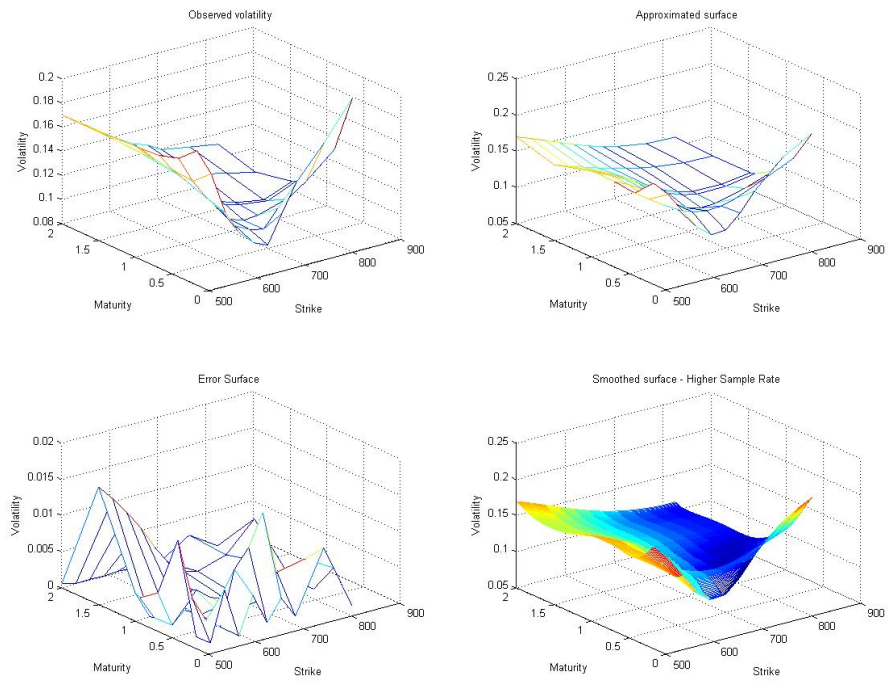


Figure 6.10: Interpolation results of the initial weight vector using a set of multi quadratic functions

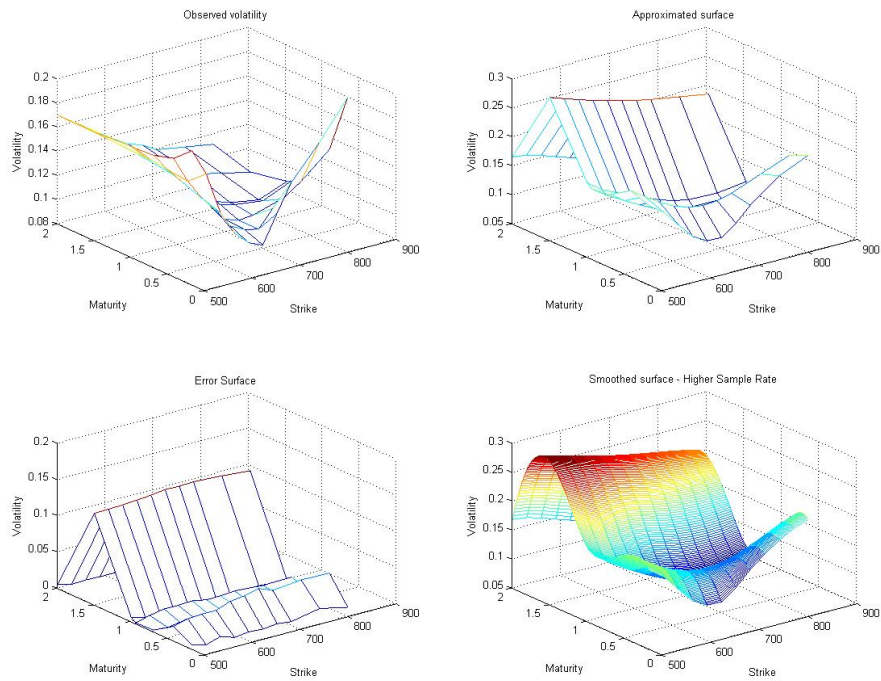


Figure 6.11: Interpolation results of the initial weight vector using a set of thin plate spline functions

The same problem was rerun, using the same parameters, but using the Trust Region optimisation algorithm described in Chapter 5. The results show a significant increase in accuracy and, more significantly, performance. Again, the thin plate spline function set proved the more accurate, closely followed by the multi quadratic set. The Gaussian set again performed poorly in comparison to the other two. Table 6.5 summarises the results and Figures 6.15, 6.16 and 6.17 show the generated local volatility surfaces. Again visual inspection reveals smooth, stable surfaces.

Radial Basis Function	Average Absolute Error	Maximum Absolute Error	Average Relative Error	Maximum Relative Error	Iterations	Cost Function Evaluations	Lambda
Gaussian	0.3896	1.5536	0.1970	1	476	10017	0
Multi quadratic	0.1484	0.5050	0.1919	1	123	2604	0
Thin plate spline	0.1296	0.5272	0.1751	1	476	10017	0

Table 6.5: Summarised results using the Trust Region algorithm

To examine the effect of size of the set of centers and their positioning we rerun the problem using an increase function set size. We place a function center at every known data point - that is the sets $t = [0.175; 0.425; 0.695; 0.94; 1; 1.5; 2]$ and $S = [501.50; 531; 560.5; 590; 619.5; 649; 678.5; 708; 767; 826]$. The discretisation parameters used for the Crank-Nicholson scheme are again $h = 0.01$ and $k = 1$. This results in 10 Crank-Nicholson solutions on the grid size 200×2478 in the worst case, per evaluation of the cost function. Figures 6.18, 6.19 and 6.20 show results of the creation of the initial weight vectors of the Gaussian, multiquadratic and thin plate spline function sets respectively.

Results showed an expected degradation in performance, but significant improvement of the accuracy.

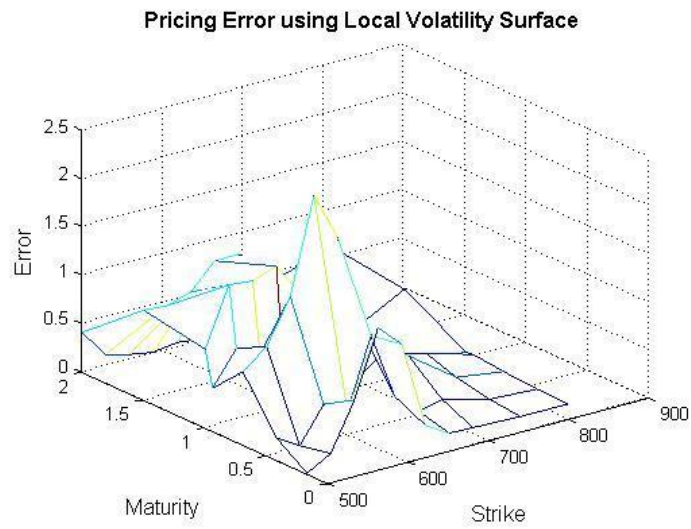
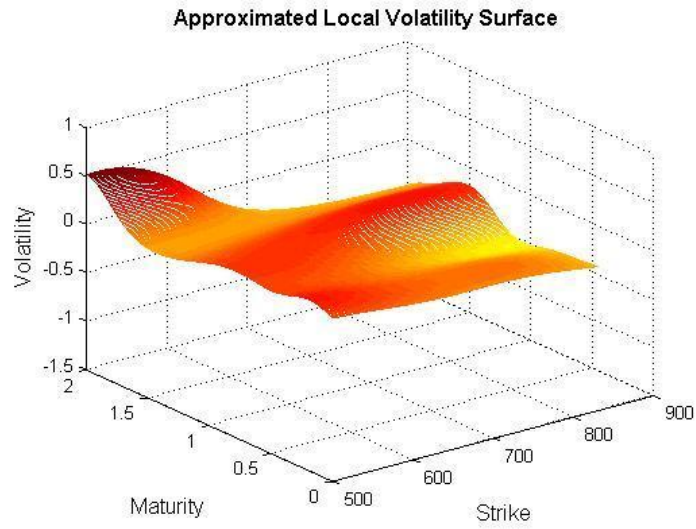


Figure 6.12: Reconstructed local volatility surface using a Gaussian function set and the Nelder-Mead optimisation method

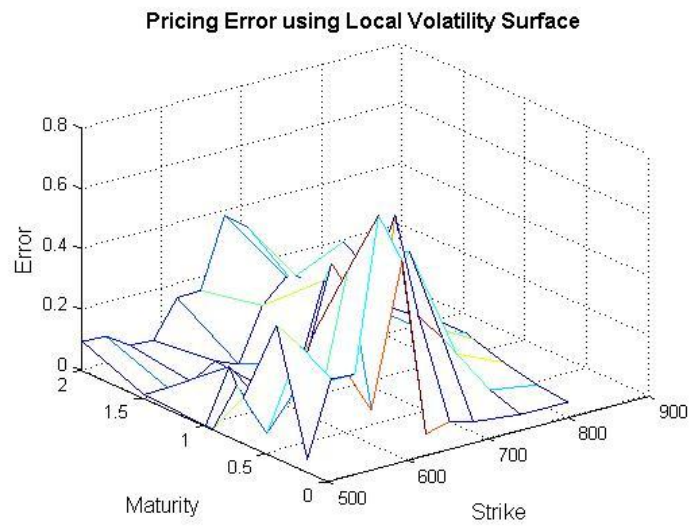
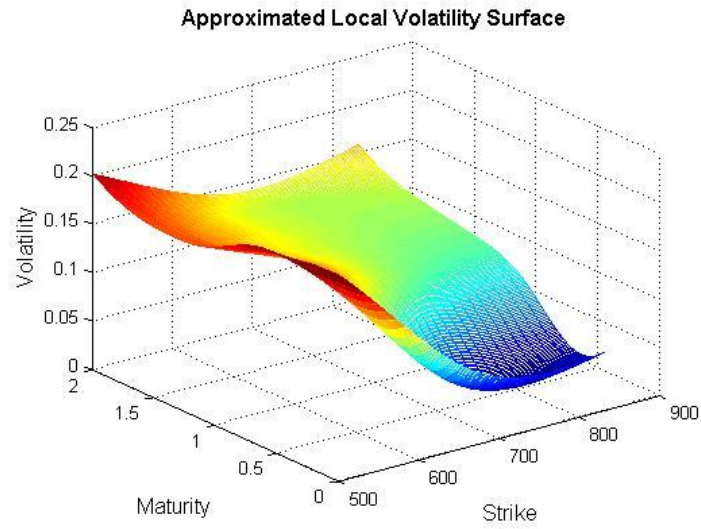


Figure 6.13: Reconstructed local volatility surface using a multi quadratic function set and the Nelder-Mead optimisation method

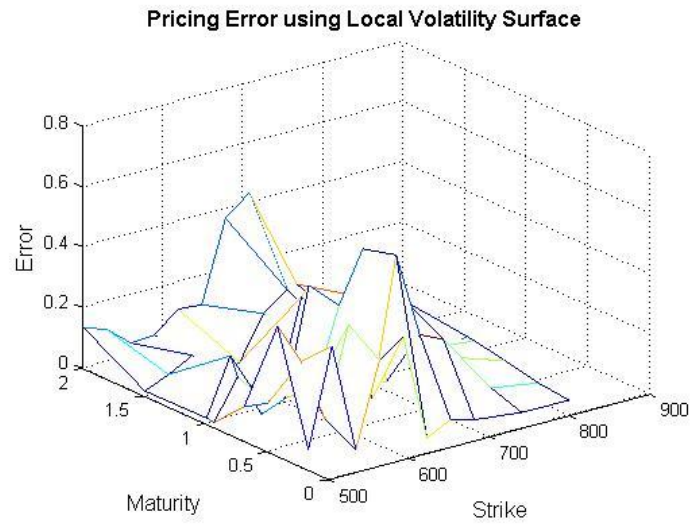
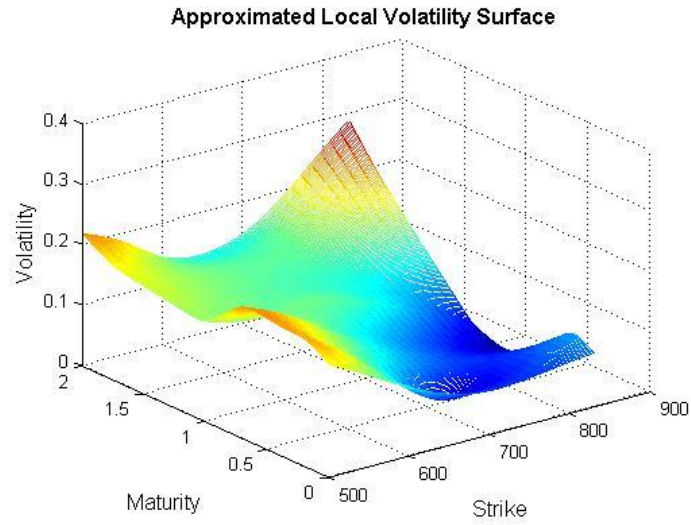


Figure 6.14: Reconstructed local volatility surface using a thin plate spline function set and the Nelder-Mead optimisation method

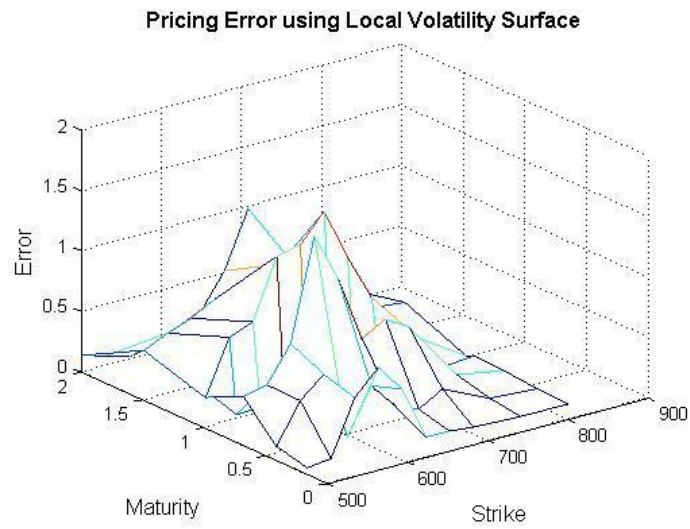
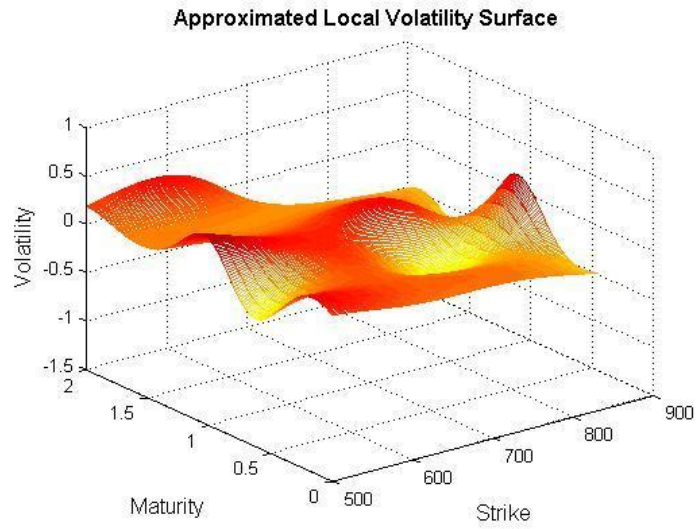


Figure 6.15: Reconstructed local volatility surface using a Gaussian function set and the Trust Region optimisation method

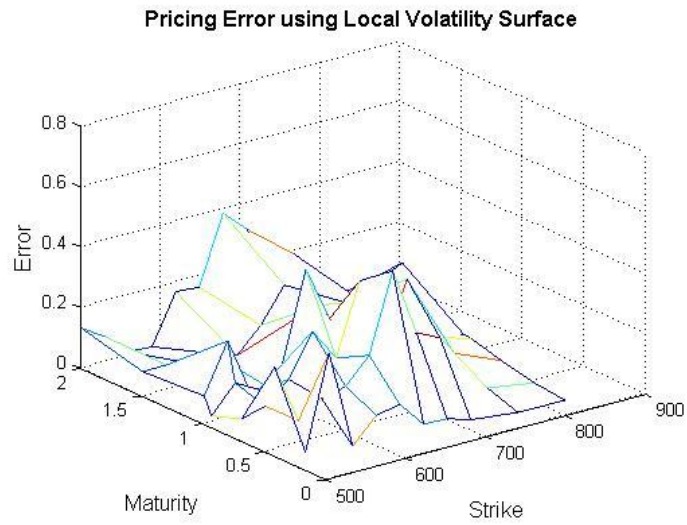
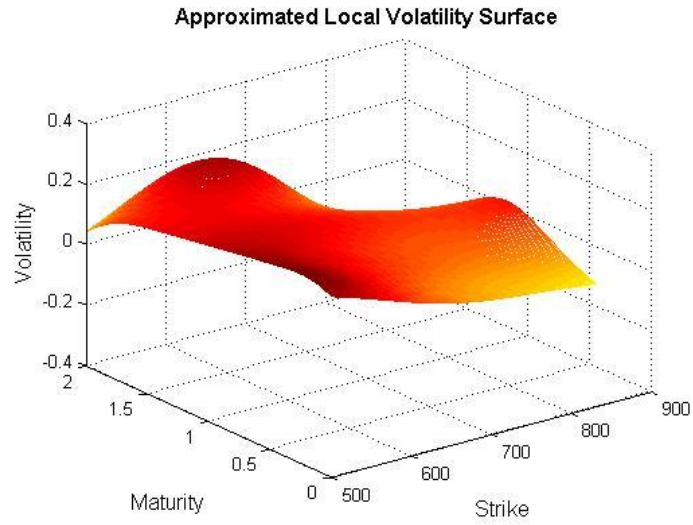


Figure 6.16: Reconstructed local volatility surface using a multi quadratic function set and the Trust Region optimisation method

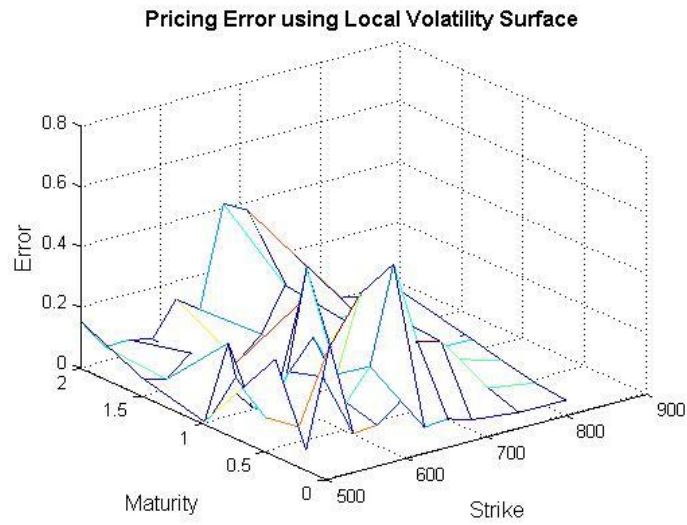
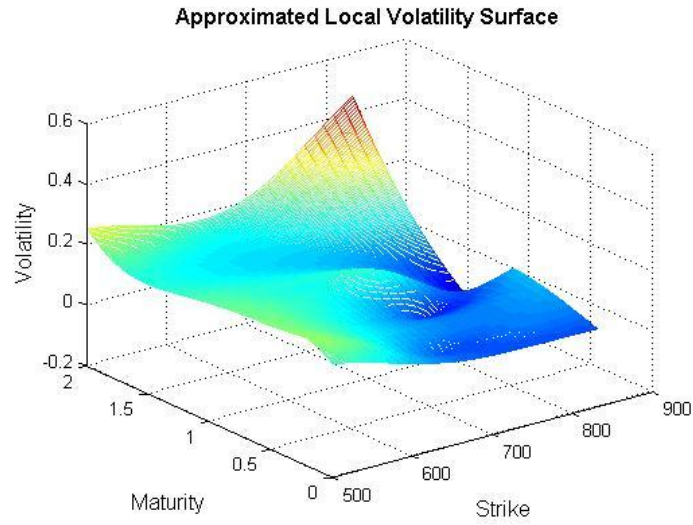


Figure 6.17: Reconstructed local volatility surface using a thin plate spline function set and the Trust Region optimisation method

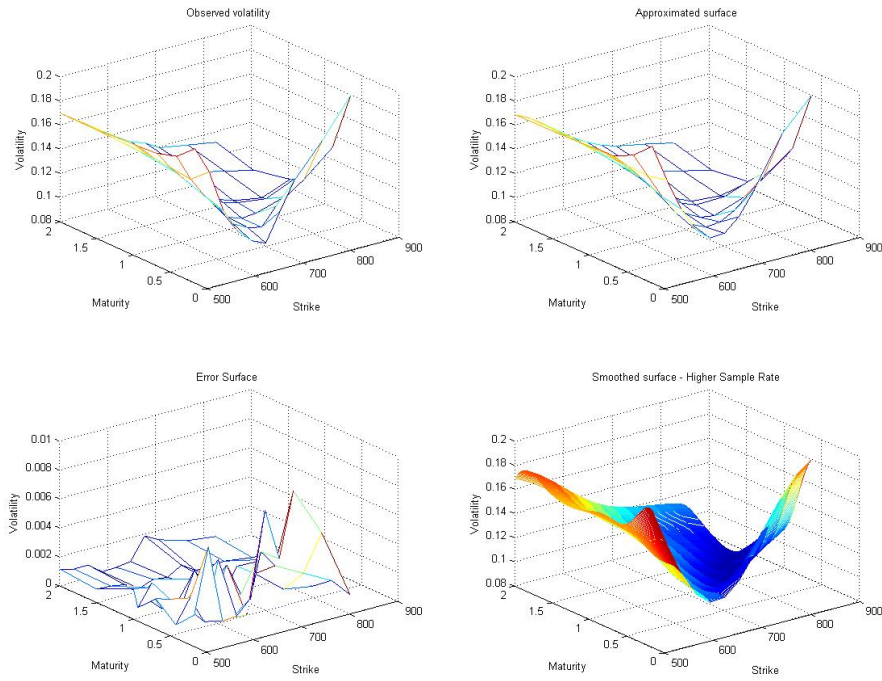


Figure 6.18: Interpolation results of the initial weight vector using a set of Gaussian functions with a larger set of centres

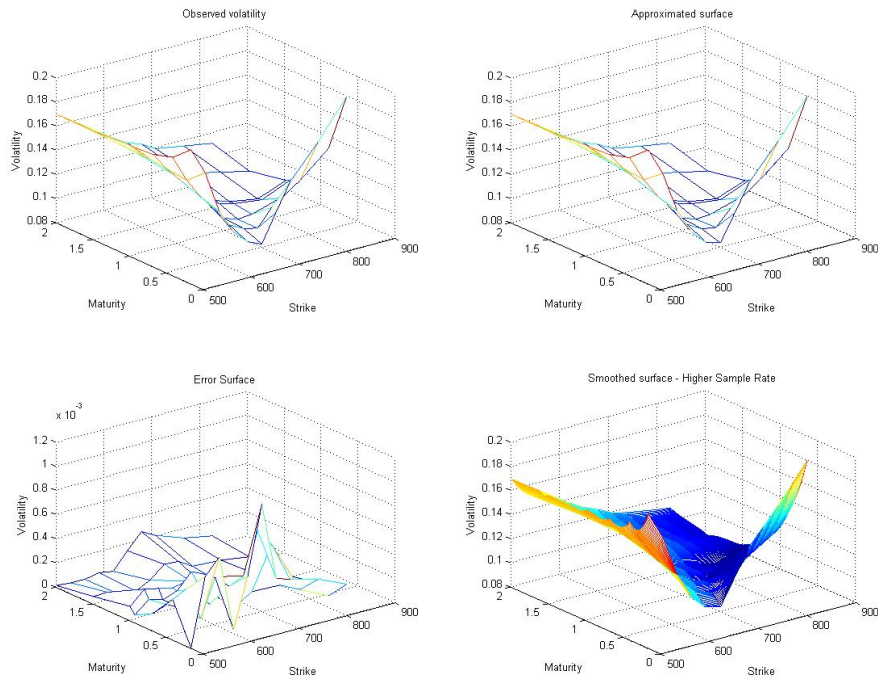


Figure 6.19: Interpolation results of the initial weight vector using a set of multi quadratic functions with a larger set of centres

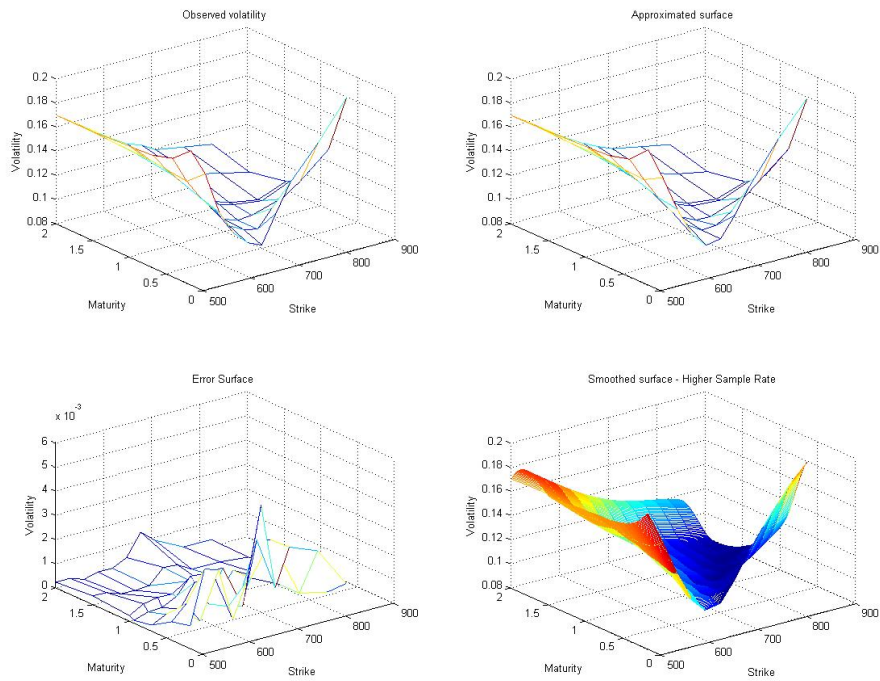


Figure 6.20: Interpolation results of the initial weight vector using a set of thin plate spline functions with a larger set of centres

Again the thin plate spline set showed the best performance. Significantly, the Gaussian set was much more in comparable with the thin plate spline and multi quadratic sets. This suggests that the performance of the approach is highly dependent on the function set and its placement. The results are summarised in Table 6.6.

Visual examination of Figures 6.21, 6.22 and 6.23 reveal that all sets are smooth and seem stable. While the Gaussian and thin plate spline function sets required no smoothing, the multi quadratic approach was over determined with the larger function set, and as a result required a smoothing factor to avoid overfitting.

Radial Basis Function	Average Absolute Error	Maximum Absolute Error	Average Relative Error	Maximum Relative Error	Iterations	Cost Function Evaluations	Lambda
Gaussian	0.1149	0.4100	0.1551	1	140	10011	0
Multi quadratic	0.1432	0.4955	0.1832	1	140	10011	0.4
Thin plate spline	0.1027	0.4108	0.1612	1	140	10011	0

Table 6.6: Summarised results using the Trust Region algorithm with an increased function set size

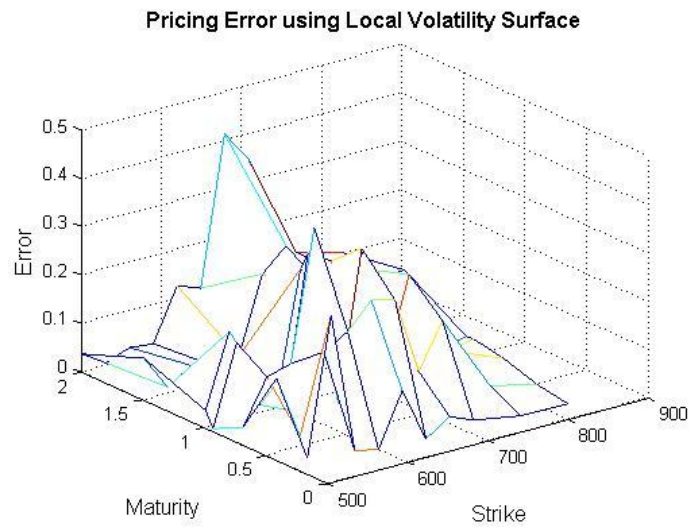
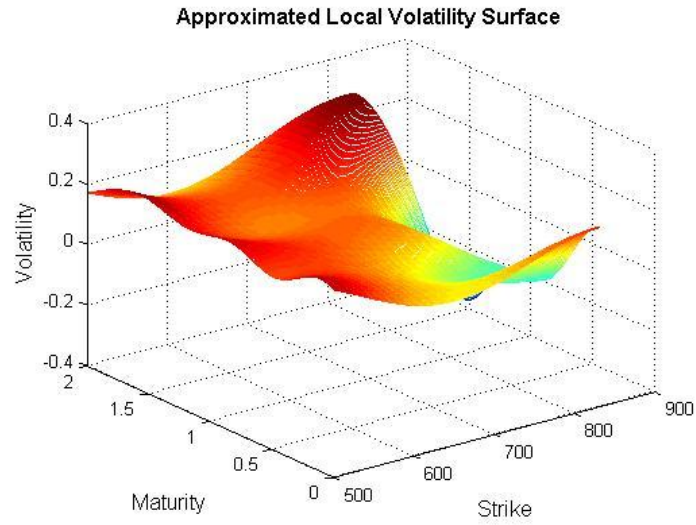


Figure 6.21: Reconstructed local volatility surface using a Gaussian function set and the Trust Region optimisation method

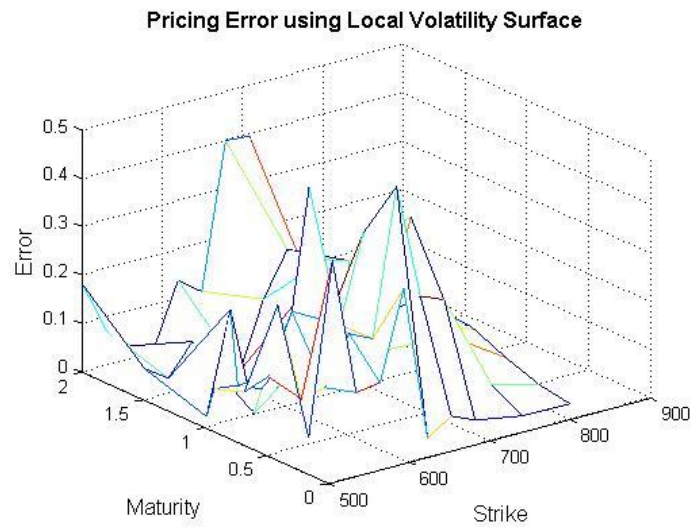
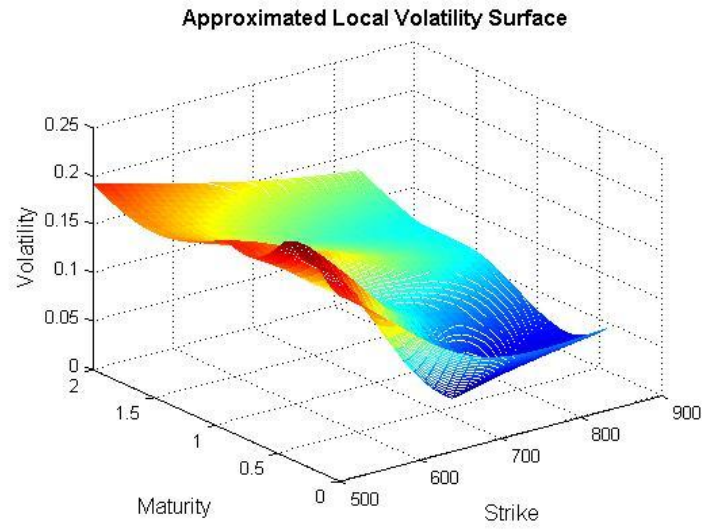


Figure 6.22: Reconstructed local volatility surface using a multi quadratic function set and the Trust Region optimisation method

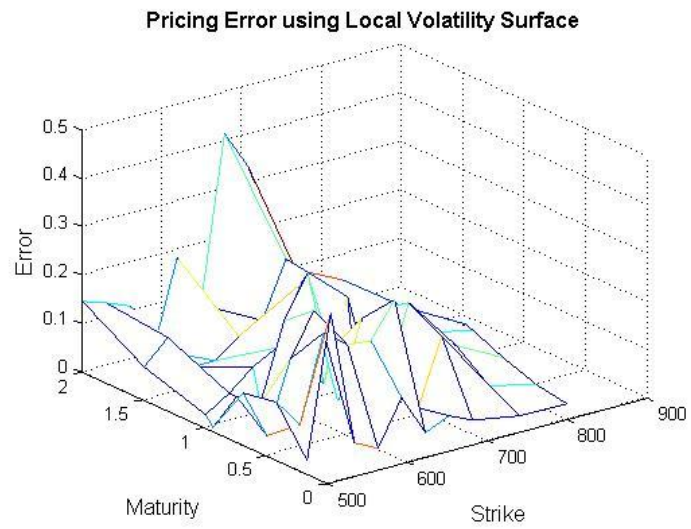
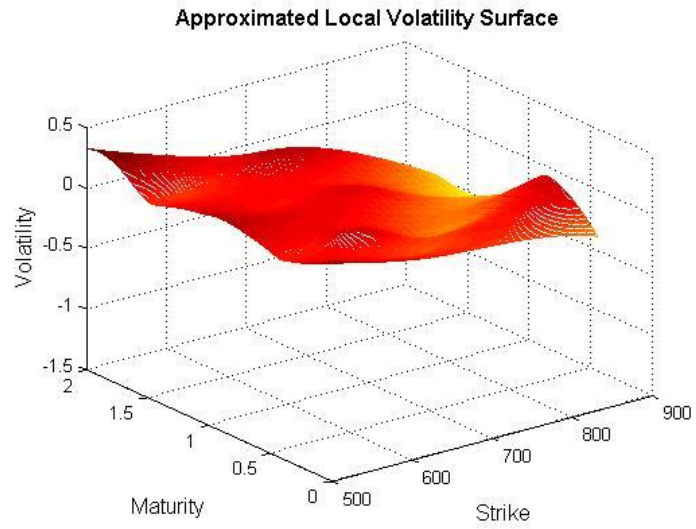


Figure 6.23: Reconstructed local volatility surface using a thin plate spline function set and the Trust Region optimisation method

Chapter 7

Discussion and future research

Results presented in Chapter 6 show the radial basis function approach to be both acceptably accurate and tractable. Our results show that both thin plate splines and multiquadratics perform well, while Gaussian function sets generally perform poorly - this finding backs up observations by Franke [1982]. Both the analytical and realistic “real world” problem compare favourably with results presented by Coleman *et al.* [1999] and Lagnado and Osher [1997a]. Table 7.1 below compares the analytical problem results with those of Coleman *et al.* [1999] and shows that more accurate results have been achieved using the radial basis function approach. Comparison with the numerical test is difficult, as there is no analytical solution available.

Method	Average Relative Error	Max Relative Error
Gaussian RBF (Nelder-Mead)	0.37708	0.9972
Multi quadratic RBF (Nelder-Mead)	0.0031	0.0312
Thin plate spline RBF (Nelder-Mead)	0.0012	0.0175
Gaussian RBF (Trust-Region)	0.3024	1
Multi quadratic RBF (Trust Region)	0.0012	0.0228
Thin plate spline RBF (Trust-Region)	0.0012	0.0165
Coleman <i>et al.</i> [1999] Spline Interpolation	0.0021	0.0078

Table 7.1: Comparison of radial basis function and spline interpolation approaches for the analytical problem

The research shows that careful consideration needs to be given to the number and placement of the centre points in order to avoid overfitting. With careful choices of these function sets and regularisation parameters smooth and accurate surfaces can be generated.

Interestingly we find significant variations on the boundary of the surfaces, as can be seen in Figure 7.1. This can be explained by two observations. Firstly, the greek vega discussed in earlier chapters, shows that options which are struck near the current underlying price are more greatly affected by volatility than those struck further away. Thus, changes in the volatility surface in regions far from the underlying price have little effect on the value of the cost function, and given the error tolerances, are negligible. This instability at the boundary is an effect also documented by Lagnado and Osher [1997a]. This research found that relative error was low in significant areas, while in areas which are insignificant (low vega regions are those which are far away from the current underlying price, and close to maturity) the relative error spikes. The fact that high relative error occurs only in insignificant areas again reinforces the approach as a valid one. Figure 7.1 shows the relative error profile of the thin plate spline approach on the numerical example - the error spikes where the strike price is far away from the current underlying price and close to maturity.

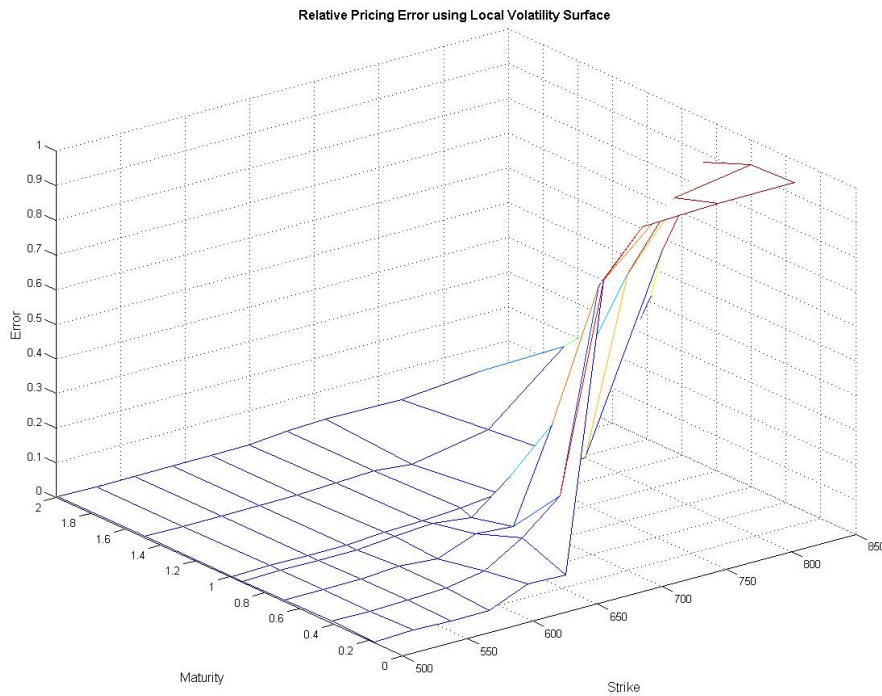


Figure 7.1: Relative error profile of the thin plate spline approach on the numerical example

The second effect is due to the lack of knowledge of the surface outside of the region described by the observable points, while the finite difference approach requires values in regions outside this area of knowledge be used - this leads to variability at the boundaries which eventually stabilises.

Experimentation has shown clearly that the ability to solve the numerical pricing problem quickly and accurately is at the crux of this problem. Results presented here are limited by the error in the numerical finite difference scheme. Decreasing the size in the discretisation, while making the solution more accurate, significantly decreases the speed and makes the whole procedure of recovering the local volatility surface prohibitively expensive. It is therefore proposed that future research be conducted to investigate approaches to either speeding up the finite difference scheme or, perhaps more importantly, finding a different numerical pricing scheme which has better convergence properties.

One of the main shortcomings of this research was control over the regularisation parameter, λ , which is employed to reduce overfitting. The approach taken was one of trial and error which certainly did not produce optimal results. While the regularisation is easy to find in a linear setting (see Orr [1996]) it is a much more challenging problem in this non-linear case. Further research should be conducted to investigate the interaction of the function set and the regularisation parameter and endeavours made to find a solution which recovers an optimal regularisation factor.

With an accurate and fast numerical pricing scheme and an optimal recovery of the regularisation parameter we believe that the radial basis function approach to recovering the local volatility surface will be a robust one.

Chapter 8

Conclusion

This research has shown the significant computational challenge of reconstructing the local volatility surface. Despite this, the radial basis function approach has been shown to be viable. The research shows that

- Radial basis functions, and in particular, thin plate splines are capable of producing accurate, smooth local volatility surfaces.
- Error estimates are of comparable scale to the spline interpolation method presented by Coleman *et al.* [1999]
- Given a sufficiently robust, tractable and efficient optimisation algorithm such as the trust-region method used in this research, solutions can be generated in a reasonable and practical amount of time

Despite the success of this research, it is perhaps the problems and pitfalls which have been identified for further investigation which may prove most useful

- Reconstruction is entirely dependent on an approach to quickly and accurately solve the discretised general Black-Scholes equation. Crank-Nicholson has been shown to have limitations in this regard.
- Accurate solutions are very sensitive to the function set and the placement of this set in the solution space. This research used both heuristic and trial and error for this purpose. An automated solution would be highly beneficial.
- One significant problem encountered in this research was the inability to accurately determine the smoothing co-efficient, λ . Combined with the function set, a robust solution to reconstruction of local volatility surfaces would require optimal recovery of this parameter.

We believe this research has provided grounding and a framework for the exploration of using radial basis functions to recovery local volatility surfaces. While by no means complete and robust, we believe it has shown significant potential exists.

Bibliography

- [Baxter 1992] B. J. C. Baxter. *The interpolation theory of radial basis functions*. PhD thesis, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, 1992.
- [Black and Scholes 1973] F Black and M Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–54, May-June 1973.
- [Bossu *et al.* 2005] S Bossu, E Strasser, and R Guichard. *Just what you need to know about variance swaps*. Technical report, JP Morgan, Equity Derivatives, 2005.
- [Byrd *et al.* 1988] R.H Byrd, R.B Schnabel, and G.A Shultz. Approximate solution of the trust region problem by minimization over two-dimensional subspaces. *Mathematical Programming*, 40:247–263, 1988.
- [Chistensen and Munk 2004] M. M. Chistensen and C. Munk. *A Note on the Numerical Solution of the Black-Scholes-Merton PDE*. Technical report, Department of Finance and Accounting, University of Southern Denmark, 2004.
- [Coleman and Verma 1998] T.F Coleman and A. Verma. *A Preconditioned Conjugate Gradient Approach to Linear Equality Constrained Minimization*. Technical report, Computer Science Department and Cornell Theory Centre, Cornell University, 1998.
- [Coleman *et al.* 1999] T.F Coleman, Y Li, and A Verma. Reconstructing the unknown local volatility function. *The Journal of Computational Finance*, 2(3):77–102, 1999.
- [Cox and Ross 1976] John C. Cox and Stephen A. Ross. The valuation of options for alternative stochastic processes. *Journal of Financial Economics*, 3(1-2):145–166, 1976.
- [Dupire 1994] B Dupire. Pricing with a smile. *Risk Magazine*, 7(1):18–20, 1994.
- [Ederington and Guan 2002] L. Ederington and W. Guan. Why are those options smiling? *Journal of Derivatives*, 10(2), 2002.
- [Franke 1982] R. Franke. Scattered data interpolation: tests of some methods. *Mathematics of computation*, 38(157):181–200, Jan 1982.
- [Hagan *et al.* 2002] P.S. Hagan, D. Kumar, A.S. Lesniewski, and D.E Woodward. Managing smile risk. *Wilmott Magazine*, 2002.
- [Haug and Taleb 2008] Espen G. Haug and Nassim N. Taleb. Why we have never used the black-scholes-merton option pricing formula (four version). *Social Science Research Network Working Paper Series*, 2008. Retrieved 24 October 2009, from <http://ssrn.com/abstract=1012075>
- [Heston 1993] S.L Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The Review of Financial Studies*, 6(2):327–343, 1993.

- [Hoerl 1962] A. E. Hoerl. Application of ridge analysis to regression problems. *Chemical Engineering Progress*, 58:54–59, 1962.
- [Hull and White 1987] J Hull and A White. The pricing of options on assets with stochastic volatilities. *The journal of finance*, 42(2):281–300, 1987.
- [Hull 2006] J.C. Hull. *Options, Futures and Other Derivatives*. Prentice Hall, sixth edition, 2006.
- [Jackson *et al.* 1998] N Jackson, E Suli, and S Howison. *Computation of Deterministic Volatility Surfaces*. Technical report, Oxford University Computing Laboratory, Numerical Analysis Group, 1998.
- [Kim *et al.* 2006] Bo-Hyun Kim, Daewon Lee, and Jaewook Lee. Local volatility function approximation using reconstructed radial basis function networks. In *ISNN (2)*, pages 524–530, 2006.
- [Lagnado and Osher 1997a] R Lagnado and S Osher. Reconciling differences. *Risk magazine*, 10:79–83, 1997.
- [Lagnado and Osher 1997b] Ronald Lagnado and Stanley Osher. A technique for calibrating derivative security pricing models: Numerical solution of an inverse problem. *Computing in Economics and Finance*, 101, 1997.
- [Mathworks] Mathworks. *Matlab Optimization Toolbox User's Guide*. Retrieved 18 March 2009, from http://www.mathworks.co.uk/access/helpdesk/help/pdf/_doc/optim/optim_tb.pdf
- [Merton 1976] R Merton. Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics*, 3:125–144, 1976.
- [Orr 1996] M. J. L. Orr. *Introduction to Radial Basis Function Networks*. Technical report, Centre for Cognitive Science, University of Edinburgh, 1996. Retrieved 3 August 2007, from <http://www.anc.ed.ac.uk/rbf/papers/intro.ps>
- [Risk Glossary 2006] Risk Glossary. *Volatility Skew*, 2006. Retrieved 5th May 2006, from http://www.riskglossary.com/articles/volatility_skew.htm
- [Rubinstein 1994] M Rubinstein. Implied binomial trees. *Journal of Finance*, 69:771–818, 1994.
- [Tikhonov 1963] A. Tikhonov. Solution of incorrectly formulated problems and the regularization method. In *Soviet Math. Doklady*, volume 4, pages 1035–1038, 1963.

Appendix A

Source Code

A.1 Train a volatility surface

```
function vs = trainSurface( StrikesTrain, MaturitiesTrain, VolsTrain, NNType, h, k,
    SMax, T, ActualValsTrain, optOptions, StrikesTest, MaturitiesTest,
    ActualValsTest)

%Train a volatility surface

InitialNetwork = TrainRBFNetwork( StrikesTrain, MaturitiesTrain, VolsTrain, NNType,
    'GCV' );

switch (NNType)
    case 'G'
        distMatrix = DistanceSurfaceG( h, k, T, SMax*3, InitialNetwork );
    case 'MQ'
        distMatrix = DistanceSurfaceMQ( h, k, T, SMax*3, InitialNetwork );
    case 'TPS'
        distMatrix = DistanceSurfaceTPS( h, k, T, SMax*3, InitialNetwork );
end

%CF = @(weights) (CostFunctionAnalytical(weights, distMatrix, SMax, T, h, k,
    StrikesTest, MaturitiesTest, ActualValsTest));
CF = @(weights) (CostFunctionNumerical(weights, distMatrix, SMax, T, h, k,
    StrikesTest, MaturitiesTest, ActualValsTest));
%CF = @(weights) (CostFunctionNumericalNM(weights, distMatrix, SMax, T, h, k,
    StrikesTest, MaturitiesTest, ActualValsTest));
%CF = @(weights) (CostFunctionAnalyticalNM(weights, distMatrix, SMax, T, h, k,
    StrikesTest, MaturitiesTest, ActualValsTest));

optimalWeights = lsqnonlin( CF, InitialNetwork.Weights, [], [], optOptions );
%optimalWeights = fminsearch( CF, InitialNetwork.Weights, optOptions );
vs = VolSurface( distMatrix, optimalWeights );
```

A.2 Numerical example cost function

```
function f = CostFunction(input, DistMatrix, SMax, T, h, k, Strikes, Maturities,
    ActualVals)

%Cost function for the numerical problem

S = 1;
r = 0.05;
lambda = 0;
weights = input;
vols = VolSurface(DistMatrix, weights);
totalWeight = 0;
count = 1;
    for j = 1:length(Strikes)
        p = CrankN(max(Maturities), Strikes(j), r, 'Call', vols(:, 1:SMax*2/k - 1));

        for i = 1:length(Maturities)
            price = p(round(S/k) , round(Maturities(i)/h) + 1);
            f(count) = price - ActualVals(j,i);
            count = count + 1;
        end
    end
f(count) = lambda*sum(abs(weights));
end
```

A.3 Analytical example cost function

```
function f = CostFunction(input, DistMatrix, SMax, T, h, k, Strikes, Maturities,
    ActualVals)

%Cost function for the analytical problem

S = 100; r = 0.05;
lambda = 0.01;
weights = input;
vols = VolSurface(DistMatrix, weights);
count = 1;
    for j = 1:length(Strikes)
        p = CrankNA(max(Maturities), Strikes(j), r, 'Call', vols(:, 1:SMax*3/k - 1));

        for i = 1:length(Maturities)
            price = p(round(S/k) , round(Maturities(i)/h) + 1);
            f(count) = price - ActualVals(j,i);
            count = count + 1;
        end
    end
f(count) = lambda*sum(abs(weights));
end
```

A.4 Generate the distance surfaces, used in calculation of the volatility surface

```
function f = DistanceSurface(h, k, T, SMax,RBFNetwork)

%Generate a distance surface for the gaussian function

M = round(T/h);
N = round(SMax/k);
nc = RBFNetwork.Centres;
w = RBFNetwork.Width;

dist = zeros(M,N,length(nc));
for i = 1:M
    for j = 1:N
        for ii = 1:length(nc)
            dist(i,j,ii) = sqrt(((j-1)*k - nc(ii,1)).^2 + ((i-1)*h - nc(ii,2)).^2);
            dist(i,j,ii) = exp(-0.5*(dist(i,j,ii)./w).^2);
        end
    end
end
f = dist;
```

```
function f = DistanceSurface(h, k, T, SMax,RBFNetwork)

%Generate a distance surface for the Multiquadratic function

format long;
M = round(T/h);
N = round(SMax/k);
nc = RBFNetwork.Centres;
w = RBFNetwork.Width;
dist = zeros(M,N,length(nc));

for i = 1:M
    for j = 1:N
        for ii = 1:length(nc)
            dist(i,j,ii) = sqrt(((j-1)*k - nc(ii,1)).^2 + ((i-1)*h - nc(ii,2)).^2);
            dist(i,j,ii) = sqrt(dist(i,j,ii).^2 + w);
        end
    end
end
f = dist;
```

```

function f = DistanceSurface(h, k, T, SMax, RBFNetwork)

%Generate a distance surface for the Thin Plate Spline function

format long;
M = round(T/h);
N = round(SMax/k);
nc = RBFNetwork.Centres;
w = RBFNetwork.Width;
dist = zeros(M,N,length(nc));

for i = 1:M
    for j = 1:N
        for ii = 1:length(nc)
            dist(i,j,ii) = sqrt(((j-1)*k - nc(ii,1)).^2 + ((i-1)*h - nc(ii,2)).^2);
            if dist(i,j,ii) ≠ 0
                dist(i,j,ii) = dist(i,j,ii).^2*log(dist(i,j,ii));
            end
        end
    end
end
f = dist;

```

A.5 Generate a volatility surface from a distance surface and weights

```

function f = VolSurface(DistanceSurface, w)

%Take the generated distance surface and weights and generate a volatility
%surface

dsSize = size(DistanceSurface);
vol = zeros(dsSize(1), dsSize(2));
vol = reshape(reshape(DistanceSurface, [], length(w)) * w, [dsSize(1), dsSize(2)]);
f = vol;

```

A.6 Crank Nicholson for the generalised Black-Scholes PDE

```
function f = CrankN(T,E,r,optionType,vols)
%Crank-Nicholson solution of the generalised Black-Scholes equation

SMax = 1.4*2;
h = 0.01; k = 0.02;

M = round(T/h);
N = round(SMax/k);

f = zeros(N-1,M+1);
bu = zeros(N-1,M+1);
bl = zeros(N-1,M+1);

    for i = 1:N-1
        f(i,1) = max([i*k - E, 0]);
    end;

    for j = 1:M
        bl(N-1,j+1) = SMax - E*exp(-r*j*h);
    end;

newSol = zeros(N-1,1);

for i = 0:M-1

    j = 1:N-1;

    a = 0.5*r*h*j + -1*(h/4).*(vols(i+1,:).^2).*(j.^2);
    b = 1 - 0.5*r*h.*j + 0.5*h.*(vols(i+1,:).^2).*(j.^2) + 0.5*r*h;
    c = -1*(h/4).*(vols(i+1,:).^2).*(j.^2);
    aBar = (h/4).*vols(i+1,:).^2.*j.^2;
    bBar = 1 - 0.5*r*h.*j - 0.5*h.*(vols(i+1,:).^2).*(j.^2) - 0.5*r*h;
    cBar = 0.5*r*h.*j + (h/4).*(vols(i+1,:).^2).*(j.^2);
    G(1) = bBar(1)*f(1,i+1) + cBar(1)*f(2,i+1);

    for ii = 2:N-2
        G(ii) = aBar(ii)*f(ii-1,i+1) + bBar(ii)*f(ii,i+1) + cBar(ii)*f(ii+1,i+1);
    end
    G(N-1) = aBar(N-1)*f(N-2,i+1) + bBar(N-1)*f(N-1,i+1);

    sol = G' + aBar(1)*bu(:,i+2) + cBar(N-1)*bl(:,i+2) - a(1)*bu(:,i+2)
        - c(N-1)*bl(:,i+2);

    cprime(1) = c(1)/b(1);
    dprime(1) = sol(1)/b(1);
    for ii = 2:N-1
        cprime(ii) = c(ii)/(b(ii) - cprime(ii-1)*a(ii));
        dprime(ii) = (sol(ii) - dprime(ii-1)*a(ii))/(b(ii) - cprime(ii-1)*a(ii));
    end;

    newSol(N-1) = dprime(N-1);

    for ii = N-1:-1:2
        newSol(ii-1) = dprime(ii-1) - cprime(ii-1)*newSol(ii);
    end
    f(:,i+2) = newSol;
end
```

A.7 Train a linear RBF Network, used to generate initial solutions

```
function RBFNetwork = TrainRBFNetwork( Strikes, Maturities, Volatilities,
    RadialFunction, ErrorFunction)

%Train a LINEAR RBF Network

count = 1;

for i = 1:length( Strikes )
    for j = 1:length( Maturities )
        X( count, : ) = [ Strikes( i ) Maturities( j ) ];
        y( count ) = Volatilities( j, i );
        count = count + 1;
    end
end

[ w r err ] = OptimalParameters( X, y, X, RadialFunction, ErrorFunction );

RBFNetwork.TrainingX = Strikes;
RBFNetwork.TrainingY = Maturities;
RBFNetwork.TrainingInput = Volatilities;
RBFNetwork.RadialFunction = RadialFunction;
RBFNetwork.ErrorFunction = ErrorFunction;
RBFNetwork.Weights = w;
RBFNetwork.Centres = X;
RBFNetwork.Width = r;
RBFNetwork.Err = err;
```


A.8 Find the optimal parameters of a linear radial basis function approximation

```
function [optW optR err] = OptimalParameters(X, y, C, RadialFunction, ErrorFunction)

%Generate the set of optimal parameters for a linear radial basis function
%eDistance is the euclidean distance between two points
%Gaussian, multiQuadratic and thinPlateSpline are the respective radial
%basis functions

X=X';
C=C';
y=y';
m = length(C);
p = length(y);
err = 20000000;
optR = 0;

for r = 0:0.05:1
for i = 1:m
    for j = 1:p
        switch RadialFunction
            case 'G'
                H(j,i) = Gaussian(eDistance(X(:,j),C(:,i)), r);
            case 'MQ'
                H(j,i) = multiQuadratic(eDistance(X(:,j),C(:,i)), r);
            case 'IMQ'
                H(j,i) = 1/multiQuadratic(eDistance(X(:,j),C(:,i)), r);
            case 'TPS'
                H(j,i) = thinPlateSpline(eDistance(X(:,j),C(:,i)));
        end
    end
end

l = 1;
lold = 0;

while (abs(l - lold) > 0.00001)
    lold = l;
    lambda = l*ones(1,m);
    Lambda = diag(lambda);
    invA = inv(transpose(H)*H + Lambda);
    P = eye(p) - H*invA*transpose(H);
    w = invA*transpose(H)*y;
    l = (transpose(y)*P*P*y*trace(invA - l*(invA)^2))/(transpose(w)*invA*w*trace(P));
end

switch ErrorFunction
    case 'GCV'
        mse = p*y'*P*P*y / Trace(P)^2;
end

if mse < err
    err = mse;
    optR = r;
    optW = w;
end
end
```

A.9 CEV model - used in the analytical example

```
function res = CallValue(r,a,Beta,spot,tau,sigma,strike)

%CEV model for analytical solution

ks = kstar(r,a,sigma,Beta,tau);
xs = xstar(ks,spot,r,a,Beta,tau);
format long;
Sum1=0;
Sum2=0;
for i = 0:100
    Sum1 = Sum1 + (exp(-xs)*(xs^i)*gammainc(ks*strike^(2-Beta),i+1+(1/(2-Beta)),
        'upper'))/gamma(i+1);
    Sum2 = Sum2 + (exp(-xs)*(xs^(i+(1/(2-Beta))))*gammainc(ks*strike^(2-Beta),
        i + 1, 'upper'))/gamma(i+1 + 1/(2-Beta));
end

res = spot*exp(-a*tau)*Sum1 - strike*exp(-r*tau)*Sum2

function res = kstar(r, a, sigma, Beta,tau)
res = 2*(r-a)/((sigma^2)*(2-Beta)*(exp((r-a)*(2-Beta)*tau) - 1));

function res = xstar(kstar, Spot, r, a, Beta, tau)
res = kstar*(Spot^(2-Beta))*exp((r-a)*(2-Beta)*tau);
```