Stable Local Volatility Function Calibration Using Spline Kernel

Thomas F. Coleman * Yuying Li ‡ Cheng Wang§

January 25, 2013

Abstract

We propose an optimization formulation using the l_1 norm to ensure accuracy and stability in calibrating a local volatility function for option pricing. Using a regularization parameter, the proposed objective function balances calibration accuracy with model complexity. Motivated by the support vector machine learning, the unknown local volatility function is represented by a spline kernel function and the model complexity is controlled by minimizing the 1-norm of the kernel coefficient vector. In the context of support vector regression for function estimation based on a finite set of observations, this corresponds to minimizing the number of support vectors for predictability. We illustrate the ability of the proposed approach to reconstruct the local volatility function in a synthetic market. In addition, based on S&P 500 market index option data, we demonstrate that the calibrated local volatility surface is simple and resembles the observed implied volatility surface in shape. Stability is illustrated by calibrating local volatility functions using market option data from different dates.

1 Introduction

One of the most important problems in finance is accurate and stable model calibration from the market option data. Calibration accuracy refers to the agreement between the model option values and the observed market option prices. Stability refers to the property that the calibrated model should be similar from a slightly changed data. Stability is crucial in practical applicability of the calibrated model and is in general a more elusive property to achieve. Calibration accuracy requires a model to be sufficiently complex but stability demands a model to be sufficiently simple, in accordance with the Ockham's razor. Proper control of these conflicting objectives is the key to ensure model calibration accuracy and stability. In this paper, we propose a new approach to calibrate, stably and accurately, a local volatility function in a diffusion process from market option prices.

The conflicting objectives in calibration of an option pricing model can be easily understood. The well-known Black-Scholes model (Black and Scholes, 1973) is simple and its calibration is

*Combinatorics and Optimization, University of Waterloo, 200 University Avenue West, Waterloo, Ontario, Canada N2L 3G1, email: tfcoleman@uwaterloo.ca. The author acknowledges funding from the Ophelia Lazaridis University Research Chair (which he holds) and the National Sciences and Engineering Research Council of Canada. The views expressed herein are solely from the authors.

†David R. Cheriton School of Computer Science, University of Waterloo, 200 University Avenue West, Waterloo, Ontario, Canada N2L 3G1, email: yuying@uwaterloo.ca. The author acknowledges funding from the National Sciences and Engineering Research Council of Canada.

‡Both authors would like to thank the anonymous referees whose comments have improved the presentation of the paper.

§David R. Cheriton School of Computer Science, University of Waterloo, 200 University Avenue West, Waterloo, Ontario, Canada N2L 3G1, email:c17wang@cs.uwaterloo.ca

typically stable. Under the pricing measure, the Black-Scholes model can be described as

$$\frac{dS_t}{S_t} = (r - q)dt + \sigma dW_t$$

where S_t is the stock price at time t, $r > 0$ is the risk free interest, q is a constant dividend yield $(0 < q < r)$, and $\sigma > 0$ is a constant volatility. The process W_t is a standard Brownian motion. In practice, the volatility σ cannot be *directly* observed from the market. The implied volatility value which is inverted from a market option price is widely used; traders frequently quote the implied volatility in place of the actual option price. Unfortunately the model lacks sufficient accuracy in pricing options; the well documented implied volatility smile attests to this fact (Rubinstein, 1994; Shimko, 1993). Many more complex models such as jump diffusion models, see, e.g., Merton (1976); Bates (1991); Naik and Lee (1990), stochastic volatility models (Hull and White, 1987; Heston, 1993), as well as jump coupled with stochastic volatility models (Broadie et al., 2007) have been proposed in mathematical finance literature. One potential problem with a complex model is additional computational and implementation cost, loss of intuition, and potential decrease in calibration stability. Practitioners may, and in fact often do, favor a simpler model.

The simplest extension to the Black-Scholes model is the local volatility function generalized BS model proposed in (Dupire, 1994; Derman and Kani, 1994),

$$\frac{dS_t}{S_t} = (r - q)dt + \sigma(S_t, t)dW_t \tag{1}$$

where the local volatility function $\sigma(S, t)$ deterministically depends on the underlying price S_t and time t. This model is attractive because there remains one single source of randomness and pricing can be based entirely on complete risk elimination. Many computational methods have been proposed to calibrate local volatility functions, see, e.g., (Dupire, 1994; Andersen and Brotherton-Ratcliffe, 1997; Coleman et al., 1999; Orosi, 2010; Glover and Ali, 2011). In He et al. (2006), we extend the calibration method in (Coleman et al., 1999) to a jump diffusion model coupled with a local volatility function. The local volatility function diffusion model remains popular in practice because of its simplicity.

The local volatility function plays an important role in option pricing. Local variance in a diffusion model (1) is shown to be a conditional expectation of the instantaneous variance in a stochastic volatility model, see, e.g., (Gatheral, 2006). Even in the simplest extension (1) to the Black-Scholes model, it is difficult to balance the conflicting goals of calibration accuracy, which requires a model to be sufficiently complex to match all given data, and stability, which demands the model to be sufficiently simple so that a slight change of data does not cause a large change in the calibrated model. In Coleman et al. (1999), the local volatility function is represented by a cubic spline with a fixed number of spline knots and end conditions. The conflicting objectives of sufficiently complex model to achieve calibration accuracy. Model simplicity for stability is balanced by choosing a small number of spline knots (making a model simple) to match market option prices sufficiently accurately. Unfortunately this process is difficult to automate and the calibrated local volatility function from this ad hoc procedure may lack stability and have unrealistic oscillations.

In this paper, using spline kernels, we propose a regularized optimization formulation to ensure both accuracy and stability in the local volatility function calibration for (1). The objective function in the proposed formulation balances the calibration accuracy with the model complexity based on a regularization parameter. The unknown local volatility function is represented by a kernel spline. The complexity of the model is controlled by minimizing the 1-norm of the coefficient vector for a kernel spline. In the context of the support vector regression for function estimation based on a finite observations, this corresponds to minimizing the number of support vectors, which in general leads to good generalization property.

The presentation of the paper is as follows. We motivate the proposed optimization formulation in §2. The accuracy and stability of the calibrated local volatility surface is illustrated computa-

tionally based on both synthetic data and market data in §4. Concluding remarks are given in §5.

2 An L_1 Optimization Formulation for Stability

In a local volatility function model (1), $\sigma(S,t)$ is assumed to be a deterministic function of the asset price S and time t. Under a model (1), the option risk can be completely eliminated by trading the underlying asset under some conditions including no arbitrage, continuous trading, and no market friction. In addition, the option value function $V(S,t)$ of time t and the underlying price S satisfies the Black-Scholes partial differential equation (PDE),

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2(S,t)\frac{\partial^2 V}{\partial^2 S} + (r-q)S\frac{\partial V}{\partial S} - rV = 0. \tag{2}$$

Initially the underlying price S_0 at time $t = 0$ is given. Let $V^0(K,T)$ denote the initial European option value for all strikes K and maturity T. It can be shown that, under (1), the initial value function $V^0(K,T)$ satisfies the adjoint partial differential equation below,

$$\frac{\partial V^0}{\partial T} - \frac{1}{2}\sigma(K,T)^2K^2\frac{\partial^2 V^0}{\partial K^2} + (r-q)K\frac{\partial V^0}{\partial K} + qV^0 = 0, \tag{3}$$

see, e.g., Dupire (1994) and Andersen and Brotherton-Ratcliffe (1997).

It can be readily seen from (3), that, under some assumptions, the local volatility function can be determined if the initial option price $V^0(K,T)$ is known for any strikes $K > 0$ and maturity $T > 0$, i.e.,

$$\left(\sigma(K,T)\right)^2 = 2\frac{\frac{\partial V^0}{\partial T} + qV^0 + K(r-q)\frac{\partial V^0}{\partial K}}{K^2\frac{\partial^2 V^0}{\partial K^2}}, \tag{4}$$

see, e.g., Dupire (1994) and Andersen and Brotherton-Ratcliffe (1997) for more discussions. Therefore, assuming the ratio on the right hand side of (4) is always nonnegative, the local volatility function can be uniquely determined if the initial European option price $V^0(K,T)$ is known for all $K > 0$ and $T > 0$.

Unfortunately market typically provides option prices only for a limited finite set of strikes and maturities. Assume that m initial market option price $\{\bar{V}_j^0\}_{j=1}^m$, corresponding to strike and maturity pairs (K_j, T_j), $j = 1, \cdots, m$, are provided. The objective of calibrating a local volatility function becomes determining $\sigma(S,t)$ such that the model option prices match given market data and the calibration of the local volatility function is stable.

Accuracy and stability requirement clearly makes local volatility function estimation a challenging problem. We consider first the following simpler problem. Assume that the local volatility function only depends on the underlying asset price S, i.e., $\sigma(S,t) \equiv \sigma(S)$. In addition, we assume for now that we actually have direct observations of the local volatility $\bar{\sigma}_j = \sigma(S_j)$, $j = 1, 2, \cdots, m$. The problem of determining a function $\sigma(S)$ from the observations $(S_j, \bar{\sigma}_j)$, $j = 1, \cdots, m$, is a well known statistical learning problem. In particular, the support vector regression (SVR) offers a potential solution approach, see, e.g., (Vapnik, 1998).

Support vector learning has a theory of uniform convergence in probability and has shown to provide good generalization performance on a wide variety of learning problems. It generalizes a number of well-known learning models such as neural networks and radial basis functions networks, see, e.g., (Vapnik, 1998). For regression problems, Girosi (1998) has shown that SVR is equivalent, under certain conditions, to some sparse approximation schemes.

We motivate our proposed formulation for the volatility function calibration by examining the properties of the solution to the support vector regression. Here we follow a derivation of the solution of SVR using the classical regularization theory described in (Girosi, 1998). Assume

that the training data $\{(S_i, \bar{\sigma}_i)\}_1^m$ is obtained from sampling some unknown underlining function $\sigma(S) \equiv \sigma(S; \alpha)$:

$$\sigma(S; \alpha) = \sum_{n=1}^{\infty} \alpha_n \phi_n(S) + \alpha_0$$

where $\{\phi_n(S)\}_{n=0}^{+\infty}$ are specified basis functions. Determining the function $\sigma(S)$ is equivalent to specifying the coefficients $\{\alpha_j\}$. This problem is clearly underdetermined because the training observation set $\{\bar{\sigma}_j\}_{j=1}^m$ is finite. Regularization by imposing an additional smoothness constraint on the solution has been used to overcome this ill-posedness. For example, we can solve a variational problem

$$\min_{\alpha} \quad C \sum_{i=1}^{m} \mathcal{E}(\bar{\sigma}_i - \sigma(S_i; \alpha)) + \sum_{n=1}^{\infty} \frac{\alpha_n^2}{\lambda_n} \tag{5}$$

where $C \geq 0$ is a constant, $\mathcal{E}(\cdot)$ is some error cost function and $\{\lambda_n\}_{n=1}^{\infty}$ is a decreasing positive sequence. A quadratic function, i.e., $\mathcal{E}(x) = x^2$, is an example of the error function. Vapnik (1998) proposes to use an ϵ-insensitive cost function

$$\mathcal{E}(z) = |z|_\epsilon \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } |z| \leq \epsilon \\ |z| - \epsilon & \text{otherwise} \end{cases}$$

It has been shown in Girosi (1998) that, with the assumed smoothness functional, independent of the cost function $\mathcal{E}(\cdot)$, the solution of (5) has the form

$$\sigma(S; \beta, \bar{\beta}) = \sum_{i=1}^{m} (\bar{\beta}_i - \beta_i) \mathcal{K}(S, S_i) + \beta_0 \tag{6}$$

where $\mathcal{K}(x, y)$ is a *symmetric kernel function*,

$$\mathcal{K}(x, y) = \sum_{n=1}^{\infty} \lambda_n \phi_n(x) \phi_n(y),$$

which is the inner product of the basis $\{\phi_n(x)\}_{n=0}^{+\infty}$ and $\{\phi_n(y)\}_{n=0}^{+\infty}$. In addition, $\bar{\beta}$ and β solve the convex quadratic programming problem below

$$\begin{aligned} \min_{\beta, \bar{\beta}} \quad & \epsilon \sum_{i=1}^{m} (\bar{\beta}_i + \beta_i) - \sum_{i=1}^{m} \bar{\sigma}_i (\bar{\beta}_i - \beta_i) + \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} (\bar{\beta}_i - \beta_i)(\bar{\beta}_j - \beta_j) \mathcal{K}(S_i, S_j) \\ \text{s.t.} \quad & \sum_{i=1}^{m} (\bar{\beta}_i - \beta_i) = 0 \\ & 0 \leq \bar{\beta}, \beta \leq C. \end{aligned} \tag{7}$$

Note that the solution $\sigma(S)$ in (6) is computed from the kernel function $\mathcal{K}(x, y)$ rather than the basis functions $\{\phi_n(y)\}_{n=0}^{+\infty}$. This use of kernel functions has led to computational success of support vector machining learning. In practice, one often specifies the kernel function directly.

Let β^* and $\bar{\beta}^*$ be the solution to (7). Let SV denote the set of support vectors identified as follows:

$$\begin{aligned} \text{SV} \quad & \stackrel{\text{def}}{=} \quad \{i \ : \ \text{either } \beta_i^* \neq 0 \text{ or } \bar{\beta}_i^* \neq 0\} \\ & = \quad \{i \ : \ \bar{\beta}_i^* - \beta_i^* \neq 0\} \end{aligned}$$

where the last equality is due to the fact that β_i and $\bar{\beta}_i$ cannot both be nonzero when $\epsilon > 0$.

The SVR solution (6) can now be expressed as

$$\sigma(S;\beta^*,\bar{\beta}^*) = \sum_{i\in \mathrm{SV}} (\bar{\beta}_i^* - \beta_i^*)\mathcal{K}(S,S_i) + \beta_0^* \tag{8}$$

The above expression implies that data points which are not support vectors have no influence in the solution. In the non-degenerate cases, slight perturbations of such data points will not affect the solution (8). Furthermore, theoretical analysis suggests that a small number of support vectors leads to a small generalization error.

Unfortunately, the local volatility function calibration problem is much more complex than the above function estimation problem based on direct function observations. Instead, only option prices, which depend nonlinearly on the local volatility function, are provided. The LVF model calibration is a function estimation problem based on *indirect* measurements, which are the given market option prices.

Motivated by the fact that the predictability of the SVR solution is directly related to a small number of nonzero coefficients (support vectors) in (8), we propose the following approach to estimate the local volatility function from a finite set of option price (indirect) observations. We use kernel splines to represent a local volatility function $\sigma(K,T)$. As direct observation on the unknown local volatility is impossible, we choose a set of training points $\{(K_i,T_i),\ i=1,\cdots,l\}$, which may or may not coincide with strikes and maturities of observed option prices. We represent a local variance function using a spline kernel as follows:

$$\left(\sigma((K,T);x)\right)^2 = \left(\sum_{i=1}^{l} x_i \mathcal{K}\left((K,T),(K_i,T_i)\right) + x_0\right)^2, \tag{9}$$

where $\mathcal{K}(\cdot,\cdot)$ is the tensor product of the two 1-dimensional spline generating kernels with an infinite number of knots in K and T respectively. We note that option values depend explicitly on $\sigma^2((K,T);x)$ rather than $\sigma((K,T);x)$. Following the standard optimization notation, here we use x to denote the unknown coefficient vector $\bar{\beta}-\beta$ for the spline kernel. A description of the one dimensional symmetric kernels generating splines is provided in Appendix A. In our implementation, we use the kernel generating spline with order 1, which is given below:

$$\begin{aligned}\mathcal{K}\left((K,T),(K_i,T_i)\right) &= \left(1 + KK_i + \frac{1}{2} \mid K-K_i \mid (K\wedge K_i + K_b)^2 + \frac{(K\wedge K_i + K_b)^3}{3}\right) \times \\ &\quad \left(1 + TT_i + \frac{1}{2} \mid T-T_i \mid (T\wedge T_i + T_b)^2 + \frac{(T\wedge T_i + T_b)^3}{3}\right)\end{aligned}$$

where (K,T) denotes a variable in the two-dimensional (strike, maturity) space. Here K_b and T_b denote lower bounds for (K,T), i.e., $(K,T)\in[-K_b,+\infty)\times[-T_b,+\infty)$. In addition, $T\wedge T_i = \min(T,T_i)$.

Given a set of option prices $\{\overline{V}_j^0\}_{j=1}^l$ with $\overline{V}_j^0$ corresponding to the initial option price with strike $\overline{K}_j$ and maturity $\overline{T}_j$, the goals of the option model calibration are twofold: accuracy and stability. Accuracy ensures that, when the calibrated model is used in pricing, option pricing is consistent with the current market information. Stability becomes especially important when the calibrated model is used for hedging and other risk management purposes. Therefore we first want to minimize the calibration error, which can be measured as

$$\sum_{j=1}^{m} w_j \left(V^0(\bar{K}_j,\bar{T}_j;x) - \bar{V}_j^0\right)^2$$

where each model initial option value $V^0(\bar{K}_j, \bar{T}_j; x)$ is uniquely determined by the local volatility function (9) specified by the unknown coefficient vector x. Weights $\{w_j \geq 0\}$ are included here to facilitate achieving desired accuracy when option values are of significantly different magnitudes.

To achieve stability, we keep the local volatility function simple to minimize the generalization error of the local volatility function based on the given finite price observations. We attempt to achieve this by explicitly pushing the coefficients in the kernel function representation (9) to zero as much as possible but still ensuring calibration accuracy. This corresponds to forcing the cardinality of the set SV of the support vector in (8) to be small. We attempt to achieve this by using $\|x\|_1$ as a regularization function. To see why this should work, we can regard the term $\rho\|x\|_1$, where $\rho \geq 0$ is a constant, as the exact penalty function for the constraints $x_i = 0$, $i = 1, \cdots, n$. Indeed, with a finite $\rho > 0$, typically some subset of the equality constraints will be satisfied. In addition, there exists a finite lower bound for ρ such that when ρ is greater than this bound, all the constraints $x_i = 0$, $i = 1, \cdots, n$, will be satisfied.

Combining these two objectives together, we propose to solve the following optimization problem:

$$\min_{x \in \Re^{l+1}} \ \frac{1}{2} \sum_{j=1}^{m} w_j \left(V^0(\bar{K}_j, \bar{T}_j; x) - \bar{V}_j^0 \right)^2 + \rho \sum_{i=0}^{l} |x_i| \tag{10}$$

where the constant $\rho \geq 0$ is a regularization parameter balancing the tradeoff between the objectives for accuracy and stability. For a larger parameter ρ, the calibration error is larger but the calibrated local volatility function tends to be simpler. Note that, if we use the quadratic penalty function $\|x\|_2^2$ in the objective function, even though the objective function becomes smooth, the coefficients $\{x_i\}$ are typically nonzero at the solution. For further discuss on the exact penalty function and the quadratic penalty function, see, e.g., (Fletcher, 1980).

The optimization problem (10) has the following equivalent constrained optimization formulation:

$$\begin{aligned} & \min_{x \in \Re^{l+1}, z \in \Re^{l+1}} \frac{1}{2} \sum_{j=1}^{m} \left(V^0(\bar{K}_j, \bar{T}_j; x) - \bar{V}_j^0 \right)^2 + \rho \sum_{i=0}^{l} z_i \\ \text{subject to} \quad & z_i - x_i \geq 0, \ i = 0, \cdots, l \\ & z_i + x_i \geq 0, \ i = 0, \cdots, l \\ & z_i \geq 0, \ i = 0, \cdots, l \end{aligned}$$

3 A Trust Region Method for the Proposed Calibration Problem

We propose to use an affine scaling trust region method to solve the piecewise smooth minimization problem (10) directly. The proposed method is an extension of the affine scaling trust region method for the bound constrained minimization proposed in (Coleman and Li, 1996a).

Without loss of generality, we describe the computational method for the problem below,

$$\min_{x \in \Re^n} f(x) + \|x\|_1 \tag{11}$$

where $f : \Re^n \to \Re^1$ is a twice continuously differentiable function. The option calibration problem (10) can clearly be written as (11) with $n = l + 1$. The objective function of (11) has a smooth component $f(x)$ and a piecewise linear component $\|x\|_1$.

In (Coleman and Li, 1996a), an interior point trust region method is proposed to solve the bound-constrained minimization problem,

$$\min_{x\in\Re^n} f(x), \qquad \text{subject to} \quad l \le x \le u \tag{12}$$

where $l, u \in \Re^n$, $l < u$, and $f : \Re^n \to \Re^1$ is a smooth function. At each iteration, the main computation is approximately solving a trust region subproblem based on an appropriate scaling, which depends on the first order Kuhn-Tucker optimality condition as well as the distance of the current iterate to the constraints. In addition, a reflective technique is used to accelerate convergence, see (Coleman and Li, 1994, 1996b) for more details.

Optimization problem (11) and the bound constrained minimization (12) share some important properties; solving these problems is to identify which variables are active, i.e., at the bounds for (12) or equal to zero for (11). Similar to the method in (Coleman and Li, 1994, 1996b) for the bound constrained minimization, we now describe subsequently a trust region method for solving (11). Analogous to the motivation in (Coleman and Li, 1994, 1996b), the trust region subproblem is based on proper affine scaling, which depends on the first order Kuhn-Tucker condition for (11) as well as the distance of the current iterate to $x = 0$.

It can be easily shown that the first-order Kuhn-Tucker conditions for (11) can be stated as follows: if x is a local minimizer, then, for $1 \le i \le n$,

$$\begin{array}{l} x_i\left((\nabla f(x))_i + \text{sign}(x_i)\right) = 0 \\ |(\nabla f(x))_i| \le 1 \end{array} \tag{13}$$

where

$$\text{sign}(x_i) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } x_i \ge 0 \\ -1 & \text{otherwise.} \end{cases}$$

and $(\nabla f)_i$ denotes the ith component of the gradient of $f(x)$.

Let the vector $v(x) \in \Re^n$ be defined below,

$$v_i(x) \stackrel{\text{def}}{=} \begin{cases} |\ x_i\ | & \text{if } |(\nabla f(x))_i| \le 1, \\ 1 & \text{otherwise .} \end{cases} \tag{14}$$

and

$$D(x) \stackrel{\text{def}}{=} \text{diag}\left(v(x)\right),$$

Thus the Kuhn-Tucker condition (13) can be stated as

$$D(x)(\nabla f(x) + \text{sign}(x)) = 0 \tag{15}$$

In other words, a vector x satisfies equations (15) if and only if the first-order Kuhn-Tucker conditions of (11) hold at x.

Assume that the current iterate $x_k \in \Re^n$ satisfies the condition $(x_k)_i \ne 0$ for $1 \le i \le n$. Let $g_k = \nabla f(x_k) + \text{sign}(x_k)$. A Newton step for (15) at the kth iteration satisfies

$$(J_k^v \cdot \text{diag}(g_k) + \text{diag}(v_k) \cdot \nabla^2 f(x_k))d_k = -\text{diag}(v_k)g_k \tag{16}$$

where $J^v(x) \in \Re^{n\times n}$ is a diagonal matrix which corresponds to the Jacobian of $|v(x)|$. Each diagonal element equals either zero or ± 1.

We can write equation (16) in an equivalent form with a symmetric coefficient matrix corresponding to a Hessian matrix of a quadratic function. Let $D_k = \text{diag}(|v_k|^{\frac{1}{2}})$ and

$$\begin{aligned} s_k &= D_k^{-\frac{1}{2}} d_k \\ \hat{g}_k &= D_k^{\frac{1}{2}} g_k \\ \hat{M}_k &= D_k^{\frac{1}{2}} \cdot \nabla^2 f(x_k) \cdot D_k^{\frac{1}{2}} + \text{diag}(g_k) \cdot J_k^v. \end{aligned}$$

Then the Newton equation (16) is equivalently expressed as

$$\hat{M}_k s_k = -\hat{g}_k. \tag{17}$$

The above equation (17) suggests an affine scaling transformation: $s_k = D_k^{-\frac{1}{2}} d_k$ and consideration of the following trust region subproblem,

$$\min_{s \in \Re^n} \{\hat{\psi}_k(s) : \|s\|_2 \leq \Delta_k\}, \tag{18}$$

where

$$\hat{\psi}_k(s) = \hat{g}_k^T s + \frac{1}{2} s^T \hat{M}_k s.$$

The subproblem (18) has its counterpart in the original variable space:

$$\min_{d \in \Re^n} \{\psi_k(d) : \|D_k^{-\frac{1}{2}} d\|_2 \leq \Delta_k\} \tag{19}$$

where

$$\begin{aligned}
&\psi_k(d) = g_k^T d + \frac{1}{2} d^T M_k d \\
&C_k = D_k^{-\frac{1}{2}} \cdot \mathrm{diag}(g_k) \cdot J_k^v \cdot D_k^{-\frac{1}{2}} \\
&M_k = \nabla^2 f_k + C_k \\
&d_k = D_k^{\frac{1}{2}} s_k
\end{aligned}$$

The affine scaling matrix D_k controls the shape of the ellipsoid created by $\|D_k^{-\frac{1}{2}} d\| \leq \Delta_k$. With the choice of $D(x) = \mathrm{diag}(|v(x)|)$, the ellipsoid is short in directions corresponding to components of $(x_k)_i$ close to zero and $|(\nabla f(x_k))_i| \leq 1$, and long in other directions. In this way the solution to the quadratic model (19) leads to small steps along the direction with components of $(x_k)_i$ close to zero and $|(\nabla f(x_k))_i| \leq 1$.

For any given direction d, we consider the following piecewise quadratic approximation of the objective function (11):

$$\phi_k(d) = \nabla f_k^T d + \frac{1}{2} d^T \nabla^2 f_k d + \|x_k + d\|_1 - \|x_k\|_1 + \frac{1}{2} d^T C_k d \tag{20}$$

It can be easily verified that $\phi_k(d)$ approximates the change of the objective function with at least an accuracy of a linear order $O(\|d\|)$.

The nonlinear system (15), derived from the KKT condition, is not differentiable when $x_i = 0$. We define the differentiable region $\mathcal{F} \stackrel{\text{def}}{=} \{x \ : \ x \in \Re^n, (x)_i \neq 0, 1 \leq i \leq n\}$. If the line segment from x_k to $x_k + d$ is a subset of $\mathcal{F}$, the quadratic approximation $\phi_k(d)$ equals the objective function $\psi_k(d)$ of the trust region subproblem.

Thus, in the proposed algorithm, we maintain differentiability for all iterates $\{x_k\}$. A simple backtracking technique used in interior point methods in (Coleman and Li, 1994, 1996b) can similarly be used to avoid landing exactly on the points of non-differentiability.

Assume that d_k is the solution to the trust region subproblem (19). It is possible that non-differentiability occurs from x_k to $x_k + d_k$, i.e., some variables may become zero during the step. For any descent direction d, let $\phi_k^*[d]$ denote the minimum value of $\phi_k(d)$ within the trust region, i.e.,

$$\phi_k^*[d] \stackrel{\text{def}}{=} \min_{\|\alpha D_k^{-\frac{1}{2}} d\|_2 \leq \Delta_k} \phi_k(\alpha d)$$

Similar to the algorithm in (Coleman and Li, 1996a) for bound constrained minimization, to ensure global convergence, at each iteration, a sufficient decrease condition needs to be satisfied. In our implementation, we use the condition below

$$\phi_k(d_k) \leq \beta_g \phi_k^*[-D_k^{-2} g_k]$$

where $0 < \beta_g < 1$ is a given constant. Let p_k be the solution to the trust region subproblem (19), to ensure local quadratic convergence, the asymptotic sufficient decrease condition for local quadratic convergence is

$$\phi_k(d_k) \leq \beta_p \phi_k^*[p_k]$$

where $0 < \beta_p < 1$ is a given constant.

The proposed trust region algorithm is summarized in Figure 1; this algorithm is an extension of the algorithm proposed in (Coleman and Li, 1996a) for bound constrained minimization. As in any typical trust region algorithm, the trust region size needs to be adjusted in each iteration to ensure a sufficient agreement between the approximating function $\phi_k(d)$ and the original objective function in (11). Convergence analysis for the proposed algorithm can be established similar to the analysis in (Coleman and Li, 1996a).

4 Computational Examples for Calibrating Local Volatility Functions

To illustrate the accuracy and stability of our proposed approach for the local volatility function calibration, we now consider a few examples, including both synthetic market and real market LVF calibration examples. In addition, we discuss various computational implementation issues such as data scaling for the kernel function, initial guess of the local volatility function, and influence of the regularization parameter ρ.

4.1 Computing Initial Option Values

In order to obtain a solution to the proposed optimization problem (10), we need to calculate function value and the gradient of the smooth part $f(x)$ of the objective function in (10), which is given below

$$f(x) \stackrel{\text{def}}{=} \frac{1}{2} \sum_{j=1}^{m} \left(V^0(\bar{K}_j, \bar{T}_j; x) - \overline{V}_j^0 \right)^2. \tag{21}$$

Thus an optimization algorithm for solving (10) requires computation of the initial option values $V^0(\bar{K}_j, \bar{T}_j; x),\ j = 1, \cdots, m$, for any given kernel coefficient vector x.

Initial European option values can be computed from either the Black-Scholes PDE (2) or its dual PDE (3). Since each evaluation of $f(x)$ requires all initial values $V^0((\bar{K}_j, \bar{T}_j); x),\ j = 1, \cdots, m$, solving the dual PDE yields all m option values by one PDE solve and thus leads to more efficient computation, compared to solving the Black-Scholes PDE (2). Therefore we solve the following dual equation (3) to compute initial option values $\{V_j^0\}_{j=1}^m$:

$$\frac{\partial V^0(K,T)}{\partial T} - \frac{1}{2}\sigma^2(K,T)K^2 \frac{\partial^2 V^0(K,T)}{\partial K^2} + (r-q)K \frac{\partial V^0(K,T)}{\partial K} + qV^0(K,T) = 0, \quad \forall K > 0,\ \forall T > 0.$$

We use the Crank-Nicolson finite difference method in a finite domain $\mathrm{D} = [0, K_{\max}] \times [0, T_{\max}]$, where $K_{\max} > 0$ is a large value, e.g., three times the initial underlying price, and $T_{\max} = \max(\{\bar{T}_j\})$. The computation also requires specification of appropriate initial and boundary conditions. For European calls, we have the initial condition $V^0(K, 0) = \max(S_0 - K, 0)$. We implement

Proposed Trust Region Algorithm. Let $0 < \mu < 1$.

For $k = 0, 1, ...$

Step 1. Compute f_k, g_k, D_k, M_k and C_k; define the quadratic model

$$\psi_k(d) = g_k^T d + \frac{1}{2} d^T M_k d.$$

Step 2. Compute a step $d_k \in \mathcal{F}$, based on the subproblem:

$$\min_d \{\psi_k(d) : \|D_k^{-\frac{1}{2}} d\|_2 \leq \Delta_k\}.$$

Step 3. Compute

$$\rho_k^f = \frac{f(x_k + d_k) - f(x_k) + \|x_k + d_k\|_1 - \|x_k\|_1 + \frac{1}{2} d_k^T C_k d_k}{\phi_k(d_k)}$$

Step 4. If $\rho_k^f > \mu$, then set $x_{k+1} = x_k + s_k$. Otherwise set $x_{k+1} = x_k$.

Step 5. Update Δ_k as specified below.

Updating Trust Region Size Δ_k
$0 < \mu < \eta < 1$, $\Lambda_U > \Lambda_L > 0$, $0 < \gamma_3 < 1$ and $0 < \gamma_0 < \gamma_1 < 1 < \gamma_2$

1. If $\rho_k^f < 0$ then set $\Delta_{k+1} = \min(\Delta_k \gamma_0, \Lambda_U)$.

2. If $0 < \rho_k^f < \mu$ then set $\Delta_{k+1} = \min(\Delta_k \gamma_k, \Lambda_U)$ where $\gamma_k = \max(\gamma_0, \gamma_1 \|D_k^{-\frac{1}{2}} d_k\|_2 / \Delta_k)$.

3. Assume $\rho_k^f \geq \mu$. Set $\Delta_{k+1} = \min(\Delta_k \gamma_k, \Lambda_U)$ where $\gamma_k = \gamma_0$, if $\|D_k^{-\frac{1}{2}} d_k\|_2 / \|p_k\|_2 < \gamma_3$ and $\Delta_k > \Lambda_L$, and $\gamma_k = \max(1, \gamma_2 \|D_k^{-\frac{1}{2}} d_k\|_2 / \Delta_k)$. Otherwise $\gamma_k = 1$ if $\rho_k^f \geq \eta$.

Figure 1: A trust region algorithm for minimizing a nonlinear function plus a 1-norm

the following boundary conditions,

$$\lim_{K\to+\infty} V^0(K,T) = 0$$
$$\frac{\partial V^0(K,T)}{\partial T} + qV^0(K,T) = 0, \ \ at \ \ K = 0.$$

In addition to the function value $f(x)$, a typical optimization algorithm requires the gradient and even Hessian matrix of $f(x)$.

Let $F : \Re^n \to \Re^m$ denote the vector of the weighted calibration error below:

$$F_j(x) = \sqrt{2w_j}\left(V^0(\bar{K}_j, \bar{T}_j; x) - \bar{V}_j^0\right), \quad j = 1, \cdots, m.$$

Then

$$f(x) = \frac{1}{2}F(x)^T F(x)$$

Let $J(x)$ be the Jacobian matrix of the first order derivatives of F with respect to x, i.e., $J(x) = \nabla F(x)$. In our implementation, we use automatic differentiation, see, e.g., (Coleman and Verma, 1996), to compute the $m \times n$ Jacobian matrix $J(x)$. Furthermore the Hessian matrix of $f(x)$ is

$$\nabla^2 f(x) = J(x)^T J(x) + \sum_{j=1}^{m} F_j(x) \cdot \nabla^2 F_j(x)$$

Since it is expected that $F(x) \approx 0$ when x approaches a solution x^*, we approximate Hessian matrix of $f(x)$ simply as below:

$$\nabla^2 f(x) \approx J(x)^T J(x) \tag{22}$$

In the spline kernel representation (9) for the local volatility function, l training vectors $\{K_i, T_i\}_{i=1}^l$ need to be specified in the region D. These training vectors do not necessarily correspond to the strikes and maturities of the option price observations. In addition, the total number of training vectors does not have to correspond to the total number of observations. In the proposed optimization, we can in fact choose the number of training vectors to be larger than the total number of observations. The training points that actually play a role in the spline representation in the calibrated volatility will be selected through the l_1 regularization. However a large number of training vectors l increase computational cost of the optimization. In our computation, we often choose the number of training vectors l to be approximately the number of market option price observations m. In addition, since the value of the local volatility far from the strike prices of the given options does not affect the initial model option values significantly, we typically place the training points uniformly in the region $[0.7S_0, 1.3S_0] \times [0, T_{max}]$.

4.2 Computing a good starting point

The calibration problem (10) is nonconvex. A judicial choice of the starting point x^0 can increase the probability of finding the global minimizer. In addition, the computational cost of the optimization algorithm can be greatly reduced with a good starting point. In the context of local volatility calibration, it is reasonable to choose the initial x^0 such that the corresponding volatility surface given by x^0 is as close to the initial implied volatility surface as much as possible. Suppose that m implied volatilities $\{\bar{\sigma}_j\}_{j=1}^m$ are given for options with strike and maturity pairs $\{(\bar{K}_j, \bar{T}_j)\}_{j=1}^m$. We choose the initial point x^0 such that the following linear systems are satisfied in the least squares sense:

$$\sigma((\bar{K}_j, \bar{T}_j); x^0) = \bar{\sigma}_j, \ \ 1 \le j \le m \tag{23}$$

Since we also want to bound the magnitudes of the coefficients, we determine the starting point by solving

$$\min_{x^0 \in \Re^{l+1}} \sum_{j=1}^{m} \left(\left(\sum_{i=1}^{l} x_i^0 \mathcal{K}((\bar{K}_j, \bar{T}_j), (K_i, T_i)) + x_0^0 \right) - \bar{\sigma}_j \right)^2 \tag{24}$$
$$l_b \leq x^0 \leq u_b$$

where l_b and u_b are lower and upper bounds respectively and $\{(K_i, T_i)\}_{i=1}^{l}$ are specified training vectors.

4.3 Reconstructing the Local Volatility Surface from a Synthetic Market

We first illustrate the accuracy and stability of the proposed local volatility function calibration method using a synthetic example described in (Coleman et al., 1999). We consider a synthetic market in which the underlying price follows an absolute diffusion process where the local volatility function σ^* is given below:

$$\sigma^*(S, t) = \frac{\sigma_c}{S} \tag{25}$$

with $\sigma_c = 15$. Thus the local volatility function depends only on S. Assume that the initial asset price $S_0 = 100$, the risk free interest rate $r = 0.05$, and the dividend rate $q = 0.02$.

Assume that 22 European call prices from this synthetic market are given; these prices are computed by the assumed local volatility function diffusion model (25). Eleven of these options have 0.5 year maturity with strikes $[90 : 2 : 110]$. The other half of options have 1 year maturity with the same set of strikes.

We determine the initial guess x^0 from the implied volatilities by solving (24) with the lower bound $l_b = -1$ and the upper bound $u_b = 1$. We set the number of training vectors $l = 18$ and place these training points evenly in the significant region; specifically the training vectors are given by $[80 : 5 : 120] \times [0.25, 0.75]$.

Figure 2 compares the calibrated local volatility function with the true synthetic market local volatility function $\sigma^*(S, t) = \frac{\sigma_c}{S}$ at time $t = 0$, $t = 0.5$, and $t = 1$. The left plots correspond to the calibration with the regularization parameter $\rho = 1$. For the right plots, the regularization parameter $\rho = 10^{-2}$. These plots demonstrate that the calibrated local volatility function is very close to the true (synthetic) market local volatility in the depicted region of $[70, 130] \times [0, 1]$. In addition, the calibrated local volatility is more accurate when t is close to 1. For regularization parameter $\rho = 1$, the calibration error $\sum_{j=1}^{m} (V_j - \bar{V}_j)^2$ equals 7×10^{-4}. When the regularization parameter $\rho = 10^{-2}$, a smaller calibration error of 1.2×10^{-5} is achieved.

As discussed before, for a support vector machine, a smaller number of nonzero coefficients (support vectors) in the kernel solution representation typically leads to a more stable prediction. Here we regard $\text{SV} = \{i \ : |x_i^*| \geq 10^{-6}\}$ as the " support vectors". For the calibration example with $\rho = 1$, the number of support vector is 8. For $\rho = 10^{-2}$, the number of support vector is 9. Note that here the dimension of the coefficient vector x is 19.

To investigate stability of calibration, we add noise in the available market option prices. Specifically we add 1% of random price errors. We let the training points be $[80 : 5 : 120] \times [0.25, 0.75]$. Other parameters setting is the same as the noise-free setting. The optimization algorithm takes similar number of iterations as before. The calibration error is, not surprisingly, larger. Figure 3 demonstrates the calibrated local volatility function for $\rho = 1$ and $\rho = 0.01$ respectively. We observe that a larger ρ leads to a slightly more stable model calibration, i.e. the local volatility function seems to experience less change.

To further illustrate the stability of the proposed calibration approach, we increase the number of training points from 18 to 52; these training points are $[70 : 5 : 130] \times [0.2 : 0.2 : 0.8]$. We choose

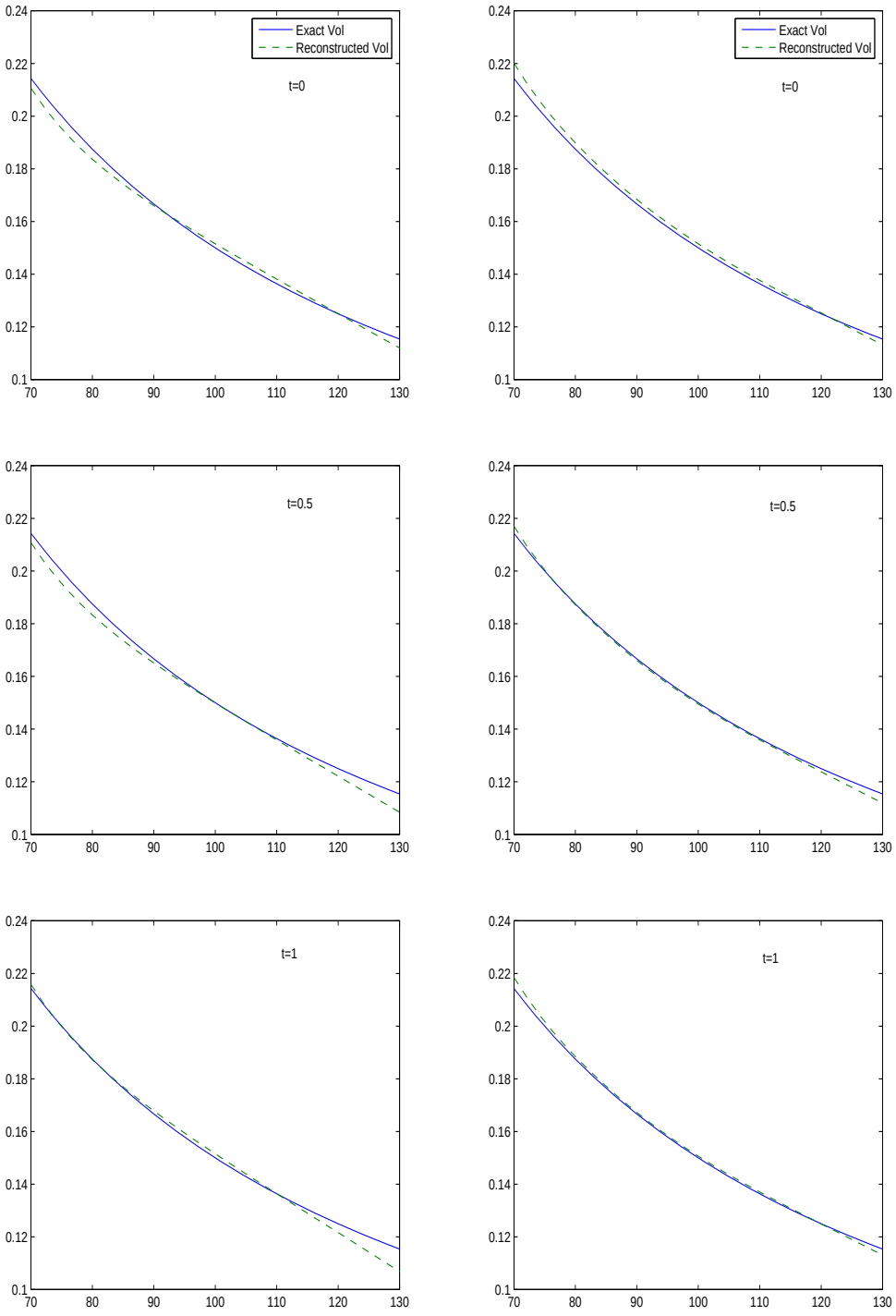

Figure 2: Two-dimensional local volatility function calibration from 22 option prices. Training points are $[80 : 5 : 120] \times [0.25, 0.75]$. The calibrated local volatility function is graphed at $t = 0,\ 0.5,\ 1$. For left plots, the regularization parameter $\rho = 1$. For right plots, the regularization parameter $\rho = 0.01$

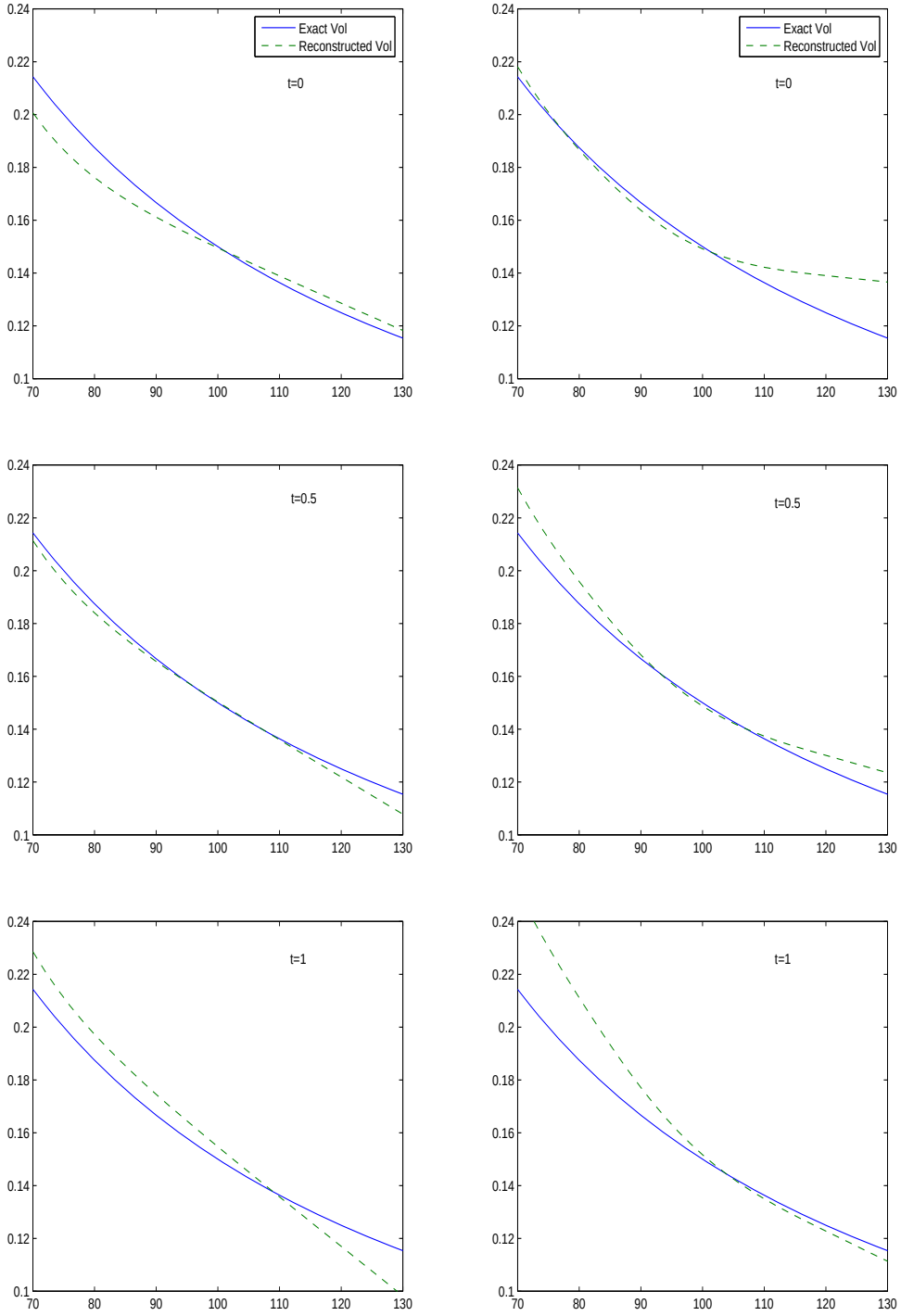

Figure 3: Local volatility calibrated from 22 noisy option prices. Training vectors: $[80 : 5 : 120] \times [0.25, 0.75]$. For the plots on the left, the regularization parameter $\rho = 1$. For plots on the right, the regularization parameter $\rho = 0.01$

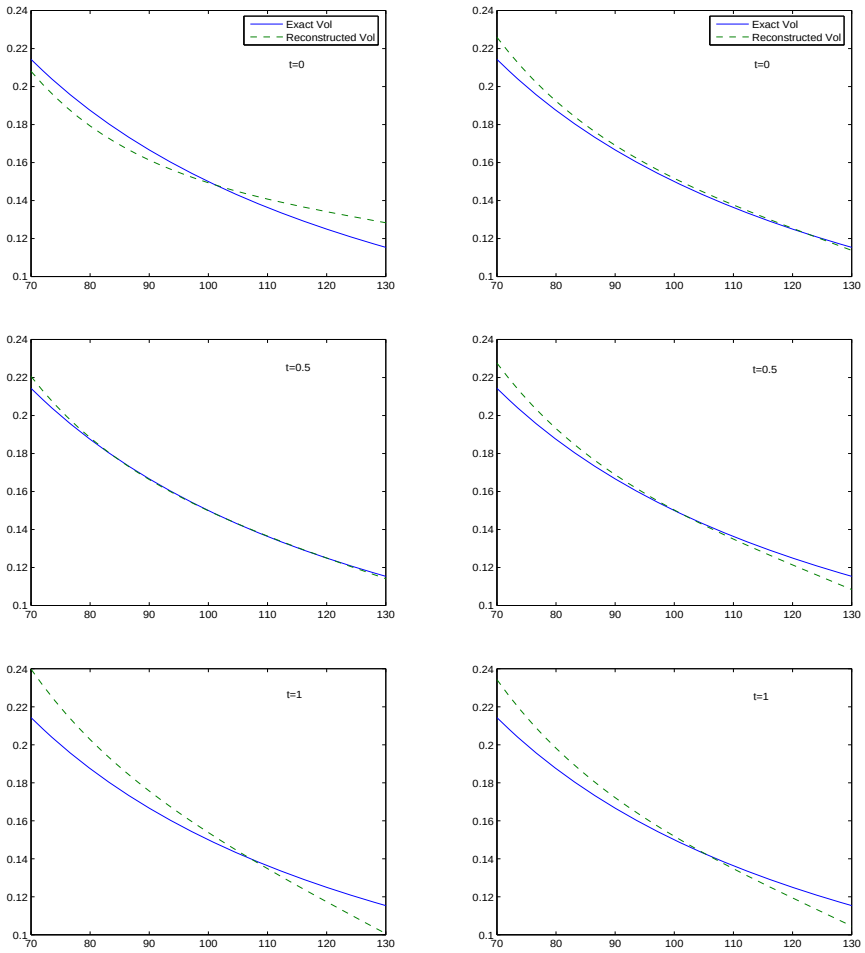

Figure 4: Calibrated local volatility function with 52 training points $[70:5:130]\times[0.2:0.2:0.8]$. The regularization parameter $\rho = 1$. Left plots are calibrated from prices with no noise. Right plots are calibrated from noisy prices.

the parameter $\rho = 1$ and calibrate the local volatility function from both error-free prices and noisy prices as before. The optimization takes a similar number of iterations and calibration errors are similar to previous examples. Figure 4 demonstrates the accuracy and stability of local volatility function calibration. We note that the local volatility surface appears to be fairly close to Figure 3 for the calibration with 18 training points. This example shows that the proposed calibration approach is relatively insensitive to the number and placement of the training points.

4.4 Calibration from S&P 500 Index Option Data

In spite of its attractive properties (such as market completeness), the calibrated local volatility function diffusion model often encounters criticisms in practical applications. One criticism is that the calibrated local volatility function from the market data often have unreasonable oscillations. In addition, the local volatility functions calibrated within a small time window seem to have unreasonably large changes.

Maturity\Strike	85%	90%	95%	100%	105%	110%	115%	120%
0.695	.172	.157	.144	.133	.118	.104	.100	.101
1	.171	.159	.150	.138	.128	.115	.107	.103
1.5	.169	.160	.151	.142	.133	.124	.119	.113

Table 1: Implied volatilities for Oct 95 S&P 500 index options with strikes in % of the spot price.

Maturity\Strike	85%	90%	95%	100%	105%	110%	115%	120%
0.695	101.9	76.26	52.76	32.75	16.47	6.02	1.93	.62
1	108	83.6	61.55	41.57	25.41	12.75	5.5	2.13
1.5	117.2	94.37	73.14	53.97	37.33	23.68	14.3	7.65

Table 2: European call prices on October 95 S&P 500 index options with strikes in % of the spot price.

To investigate how our proposed calibration approach responds to these criticisms, we now consider the local volatility functions calibrated from our proposed method using the S&P 500 market option data. Specifically, we are interested in the characteristics and stability of the calibrated local volatility surface. Table 1 presents implied volatilities for S&P 500 index options on October 1995; this data is also used in (Andersen and Brotherton-Ratcliffe, 1997) and (Coleman et al., 1999). On this day, the S&P 500 index value $S_0 = \$590$, interest rate $r = 6\%$, and dividend rate $q = 2.62\%$. The corresponding European call prices are listed in Table 2.

We first calibrate the option prices by setting the weights uniformly, i.e., $w = [1, 1, \cdots, 1]$. We choose 18 training points given by $[0.8S_0 : 0.05S_0 : 1.2S_0] \times [0.5, 1]$. Note that the calibration error is expected to be much larger than in the previous synthetic market example because the option prices are almost 6 time larger. Therefore we choose a larger regularization $\rho = 10$ to balance the calibration error and model complexity consideration.

The left plot in Figure 6 graphs the calibrated local volatility surface with uniform weights. It can be observed that the local volatility surface is quite smooth in the graphed region $[.7S_0, 1.3S_0] \times [0, 1.5]$. We also note that the calibrated local volatility bears similar shape to that of the implied volatility.

The relative price errors $\frac{\bar{V}(\bar{K}_i,\bar{T}_i)-V(\bar{K}_i,\bar{T}_i)}{V(\bar{K}_i,\bar{T}_i)}$ (in %) is presented in Table 3. From Table 3, the calibration error is within $\pm 1\%$ for most options. However, for out-of-the money options with short maturities, the relative errors are larger because these prices are relatively smaller; thus the contribution of the corresponding squared errors to the objective function $f(x)$ become relatively negligible. The calibration errors for out-of-the-money call options with short maturities can be decreased if we allocate a larger weight w_j for these terms. For example, we increase the weights for the four out-of-money options on the top right corner in Table 1 by setting individual weights as below,

$$w_j = \begin{cases} 45 & j = 8 \\ 4 & j = 7, 16 \\ 2 & j = 15 \\ 1 & \text{otherwise.} \end{cases} \tag{26}$$

Table 4 presents the relative calibration errors with individual weights in (26). It can be observed that errors for out-of-the money call options are now smaller than the errors when the weights are set uniformly.

Next, we investigate stability of the local volatility calibration for the proposed method by considering multiple calibrations from market option data within a short time period. We consider

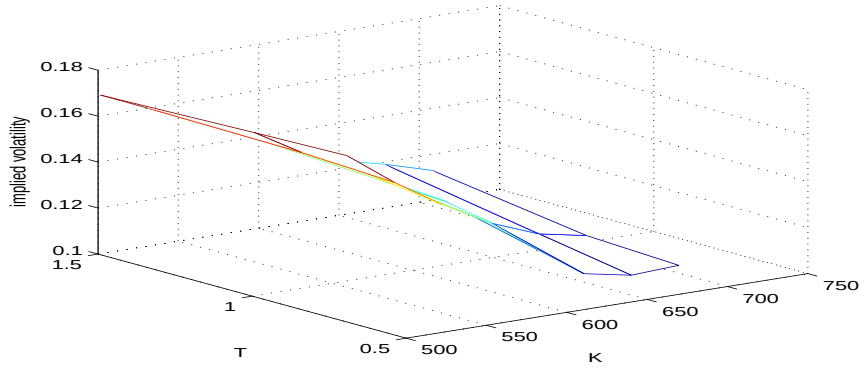

Figure 5: Implied volatility for S&P 500 index option market data on Oct 1995.

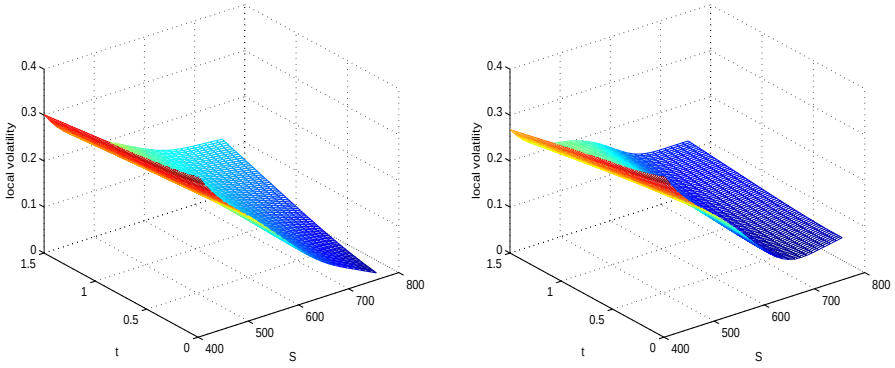

Figure 6: Calibrated local volatility surface from S&P 500 index option market data on Oct 1995. Left plot: uniform weights. Right plot: individual weights (26)

Maturity\Strike	85%	90%	95%	100%	105%	110%	115%	120%
0.695	-.10	0.19	.66	.06	3.55	11.4	-2.37	-51.79
1	-.10	-.06	-.72	-1.08	-2.53	-.76	-2.35	-19.37
1.5	.17	.11	.07	-.10	.19	1.37	-.26	.12

Table 3: Relative calibration error in % for S&P 500 index options on Oct 95; strikes are in % of the spot and uniform weights are used.

Maturity\Strike	85%	90%	95%	100%	105%	110%	115%	120%
0.695	-.09	0.20	.61	-.23	2.78	11.95	11.11	-13.39
1	-.11	-.02	-.64	-1.07	-2.79	-.71	2.94	2.74
1.5	.09	.15	.25	.1	.04	.45	-1.29	1.5

Table 4: Relative calibration errors in % for the S&P 500 index options data on October 95 with individual weights (26)

02 March 2004

Maturity\Strike	1025	1050	1100	1125	**1150**	1200	1250	1300
.58	.197	.1872	.1645	.1588	.1538	.1398	.1323	.1257
.84	.194	.1801	.1709	.1595	.1576	.1448	.1344	.1324
1.34	.1976	.1908	.1782	.1725	.1649	.1577	.1503	.1402

05 April 2004

Maturity\Strike	1025	1050	1100	1125	**1150**	1200	1250	1300
.5	.1952	.1852	.1597	.1582	.1471	.1305	.1228	.1164
1	.201	.1936	.1797	.1689	.1628	.152	.1473	.1394
1.25	.2097	.196	.1894	.1785	.1776	.1673	.1584	.1511

Table 5: Implied volatilities for S&P 500 index options on 02 Mar 2004 and 05 April 2004

02 March 2004

Maturity\Strike	1025	1050	1100	1125	**1150**	1200	1250	1300
.58	141	120.4	81	65	50.9	26.9	12.7	5
.84	148.7	127.1	93	75	62.2	37.4	20	10.9
1.34	164.2	145.8	111.8	96.4	81	57.6	38.6	23

05 April 2004

Maturity\Strike	1025	1050	1100	1125	**1150**	1200	1250	1300
.5	138.9	117.9	77.2	62.2	46	21.7	8.8	2.8
1	157.1	138.1	103	85.2	70.6	45.9	29.2	16.3
1.25	167.7	146.4	115.2	97.3	85.3	60.2	40.3	25.6

Table 6: European call prices on S&P 500 Index. On March 2, 2004, the index value $S_0 = \$1149.1$. The index value is $S_0 = \$1150.57$ on April 5, 2004

02 March 2004

Maturity\Strike	1025	1050	1100	1125	**1150**	1200	1250	1300
.58	-.9	-.88	.92	.12	-1.34	.48	-2.35	-5.51
.84	-.16	1.41	-.38	1.86	-.86	.57	3.7	-6.59
1.34	.02	.0	-2.45	-.46	.76	-.92	-1.68	3.57

05 April 2004

Maturity\Strike	1025	1050	1100	1125	**1150**	1200	1250	1300
.5	-.42	-.48	1.66	-1.57	-.29	2.92	-3.55	-16.19
1	.09	-.06	-.68	1.13	1.13	1.7	-2.79	-2.09
1.25	-.54	1.29	-1.11	1.07	-1.52	-1.16	.37	3.06

Table 7: Relative calibration error in % for S&P 500 index options on March 2, 2004 and April 5, 2004

02 March 2004

Maturity\Strike	1025	1050	1100	1125	**1150**	1200	1250	1300
.58	-.92	-.93	.78	-.05	-1.51	.63	-.87	-.66
.84	-.14	1.42	-.38	1.86	-.82	.84	4.64	-4.45
1.34	.02	.02	-.19	-.38	.83	-.99	-2.2	1.88

05 April 2004

Maturity\Strike	1025	1050	1100	1125	**1150**	1200	1250	1300
.5	-.45	-.54	1.5	-1.76	-.44	3.72	.95	-3.56
1	.12	-.02	-.62	1.21	1.27	2.13	-1.92	-.84
1.25	-.54	1.29	-1.11	1.06	-1.56	-1.31	-.2	1.24

Table 8: Relative calibration error in % for S&P 500 Index Options on March 2, 2004 and April 5, 2004 calibrated with individual weights

here two sets of S&P 500 options data, chosen from two close by dates, March 2, 2004 and April 5, 2004. On March 2, 2004, the index value $S_0 = \$1149.1$. The index value is $S_0 = \$1150.57$ on April 5, 2004. The other pricing parameters for the two examples are the same: interest rate $r = 1\%$ and dividend yield $q = 1.6\%$. Table 5 and 6 present the implied volatilities and the corresponding European call prices. Similarly 18 training points $[0.8S_0 : 0.05S_0 : 1.2S_0] \times [0.5, 1]$ are chosen. Considering the magnitudes of the option prices, we set the regularization parameter $\rho = 10$.

Table 7 reports the relative calibration error in % with uniform weights. Figure 7 and Figure 8 graph the (1D) local volatilities at the specified time and the (2D) local volatility surface respectively. Comparing the left plots with the right plots, it can be observed that the calibrated local volatility on March 2, 2004 closely resembles the calibrated local volatility on April 5, 2004. Indeed, the shape of the calibrated local volatility also resembles closely to that of the observed implied volatility. The local volatility is higher for the out-of-the- money option.

To improve the calibration accuracy for out-of-the money options, we assign larger weights for the out-of-the money options on the top right corner as below,

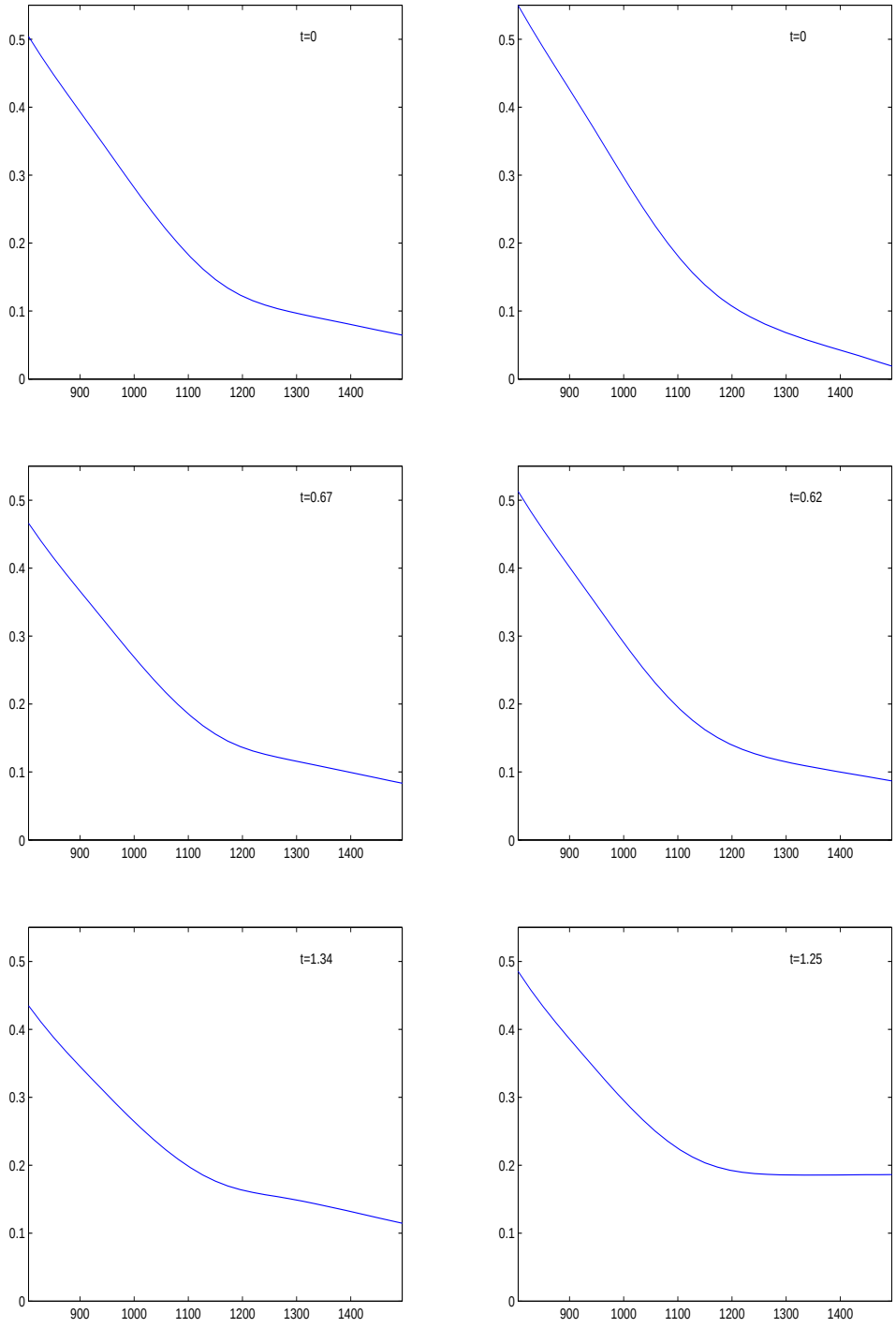

Figure 7: Calibrated local volatility for the S&P 500 index. Left plots are for the market option data on March 2, 2004. Right plots are for the option market data on April 5, 2004. Uniform weights are used.

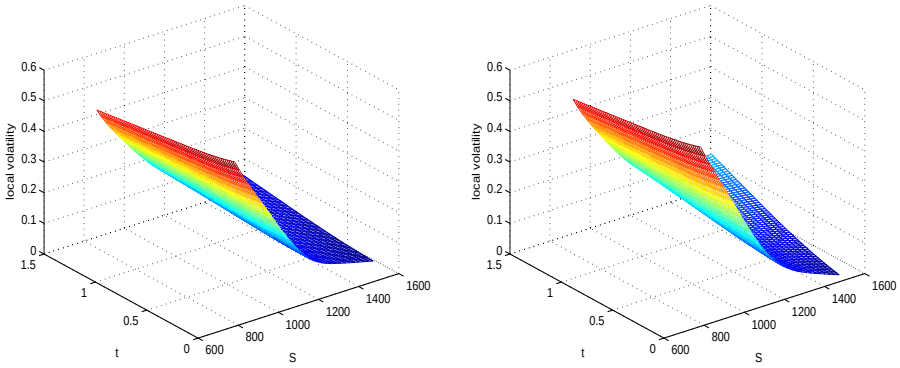

Figure 8: Calibrated local volatility surface for the S& P 500 index. Left plot is for market option data on March 2, 2004. Right plot is from market option data on April 5, 2004. Uniform weights are used

$$w_j = \begin{cases} 9 & j = 8 \\ 4 & j = 16, 24 \\ 2 & j = 7, 15, 23 \\ 1 & \text{otherwise.} \end{cases} \tag{27}$$

Table 8 presents the relative calibration error when the weights w are set as in (27). Figures 9 and 10 graph the corresponding calibrated local volatility at the specified time and the 2D local volatility surface respectively. From Table 8, it can be observed that the calibration error for out-of-the-money options are significantly improved. In addition, the calibrated local volatility surfaces remain simple and smooth, closely resemble in shape the calibrated local volatility with uniform weights. The difference is that now the local volatility around the strikes of the out-of-the-money options becomes slightly higher.

5 Concluding Remarks

The local volatility function diffusion model for option pricing extends the classical Black-Scholes constant volatility model. This extension is attractive in practice because it allows for the possibility of calibrating the observed implied volatility smile and maintains market completeness at the same time. The ability to calibrate the volatility smile and volatility term structure comes from the increase in model complexity, i.e., volatility is changed from a constant to a deterministic function of the asset price and time. Unfortunately, when calibrating a local volatility function in practice, one often encounters a challenge: the market offers only a limited set of implied volatilities (thus option prices) which are insufficient in determining an unique local volatility function. In addition, instead of direct observations of local volatility values, the market offers only option prices which depend nonlinearly on the local volatility function.

These calibration challenges have hampered practical application of the local volatility function diffusion model for option pricing and risk management. The calibrated local volatility function based solely on minimizing calibration errors often lacks credibility because it typically contains unnatural oscillations. In addition, the calibrated local volatility functions from the market data can change greatly on different days.

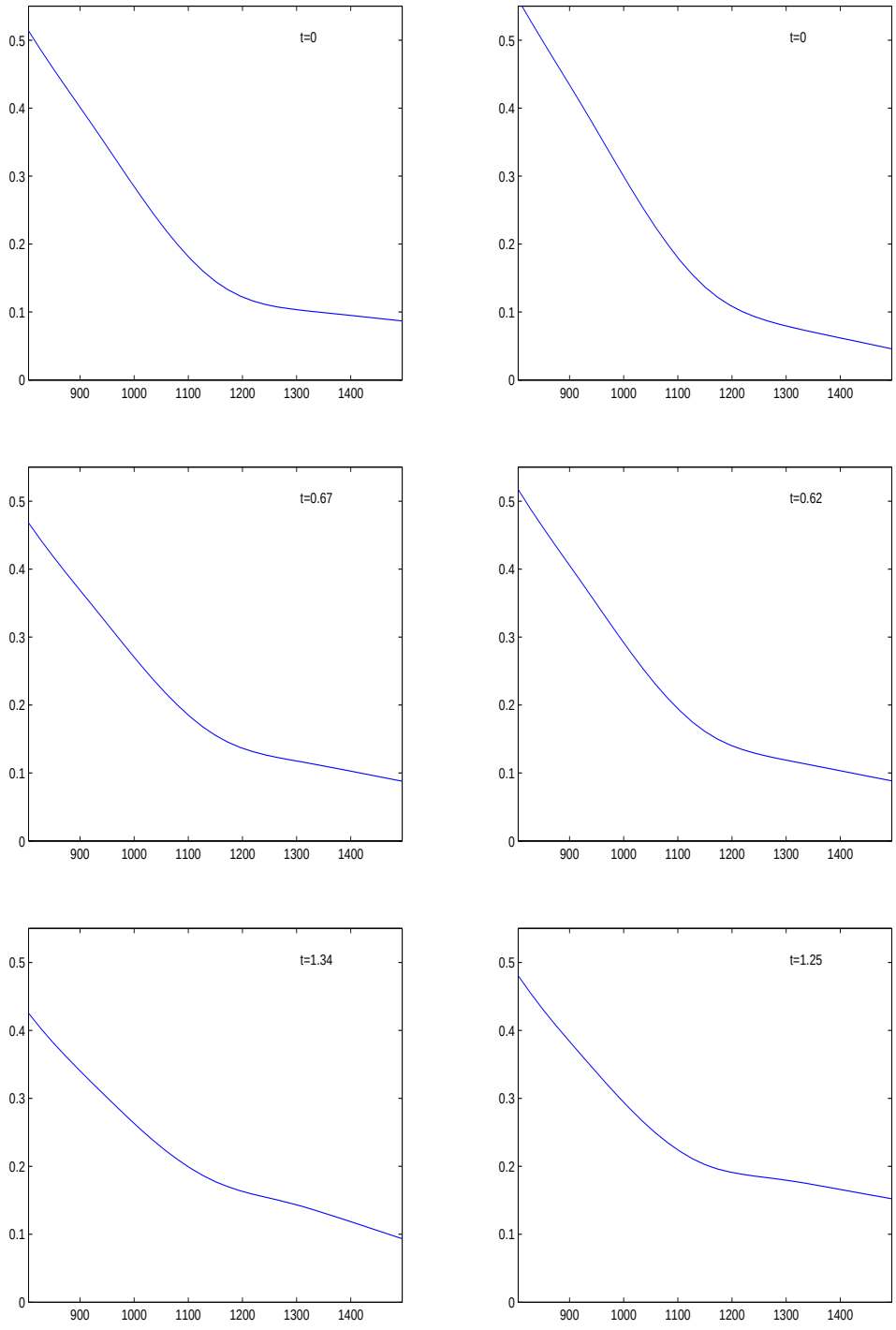

Figure 9: 2-D calibrated local volatility of S&P 500 index option with individual weights. Left plots are from option data on March 2, 2004. Right plots are from option data on April 5, 2004

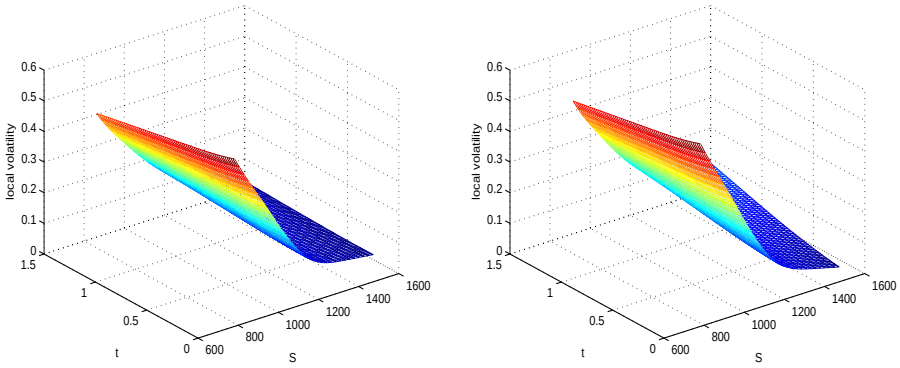

Figure 10: Calibrated local volatility surface for the S& P 500 index. Left plot is for market option data on March 2, 2004. Right plot is from market option data on April 5, 2004. Individual weights are used.

From statistical learning theory, it is well known that a successful function estimation method based on a finite set of observations requires the balancing act of ensuring accuracy of estimation on the observed data as well as maintaining simplicity of the model. Unfortunately, the local volatility calibration problem departs from the classical function estimation problem in that only the option price observations, rather than local volatility function observations, are provided by the market.

We propose an optimization formulation to calibrate the local volatility function accurately and stably. We represent a local volatility function based on kernel functions generating splines. Using a regularization parameter, the proposed objective function balances the calibration accuracy with the model simplicity. The complexity of the model is controlled by minimizing the 1-norm of the kernel coefficient vector. In the context of the support vector regression for function estimation based on a finite observations, this corresponds to minimizing the number of support vectors.

In this paper, we first illustrate the ability of the proposed approach to reconstruct the local volatility accurately and stably in a synthetic local volatility market. In addition, based on market S&P 500 option data, we further demonstrate that the calibrated local volatility surface is simple and resembles in shape the observed implied volatility surface. The stability of the calibration is illustrated by the fact that the calibrated local volatility functions from option data on closeby dates show no significant change.

In this paper, we have examined the performance of the proposed method for the liquid S&P 500 standard index options which have relatively short term maturities. Although the proposed method can definitely be applied to other option markets, including options with long term maturities, it may be interest to empirically investigate accuracy and stability of the method in those markets. Moreover, it will be interesting to further investigate and improve computational efficiency of the proposed optimization method to make it applicable in real time applications.

References

Andersen, L. B. G. and R. Brotherton-Ratcliffe (1997). The equity option volatility smile: An implicit finite difference approach. *Journal of Computational Finance* 1(2), 5–37.

Bates, D. S. (1991). The crash of '87: Was it expected? The evidence from options markets. *Journal of Finance* 46, 1009–1044.

Black, F. and M. Scholes (1973). The pricing of options and corporate liabilities. *Journal of Political Economy* 81(1), 637–654.

Broadie, M., M. Chernov, and M. Johannes (2007). Model specification and risk premia: Evidence from futures options.

Coleman, T. F. and Y. Li (1994). On the convergence of reflective Newton methods for large-scale nonlinear minimization on bounds. *Mathematical Programming* 67, 189–224.

Coleman, T. F. and Y. Li (1996a). An interior, trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization* 6(2), 418–445.

Coleman, T. F. and Y. Li (1996b). A reflective Newton method for minimizing a quadratic function subject to bounds on the variables. *SIAM Journal on Optimization* 6(4), 1040–1058.

Coleman, T. F., Y. Li, and A. Verma (1999). Reconstructing the unknown local volatility function. *Journal of Computational Finance* 2(3), 77–102.

Coleman, T. F. and A. Verma (1996). Structure and efficient Jacobian calculation. In M. Berz, C. Bischof, G. Corliss, and A. Griewank, eds., *Computational Differentiation: Techniques, Applications, and Tools*, pp. 149–159. SIAM, Philadelphia, Penn.
URL `citeseer.ist.psu.edu/coleman96structure.html`

Derman, E. and I. Kani (1994). Riding on a smile. *Risk* 7, 32–39.

Dupire, B. (1994). Pricing with a smile. *Risk Magazine* 7(1), 18–20.

Fletcher, R. (1980). *Practical Methods of Optimization: Unconstrained Optimization*. John Wiley and Sons.

Gatheral, J. (2006). *The Volatility Surface: A Practitioner's guide*. Wiley.

Girosi, F. (1998). An equivalence between sparse approximation and support vector machines. *Neural Computation* 10.

Glover, J. and M. M. Ali (2011). Using radial basis functions to construct local volatility surfaces. *Applied Mathematics and Computation* pp. 4834–4839.

He, C., J. S. Kennedy, T. F. Coleman, P. A. Forsyth, Y. Li, and K. Vetzal (2006). Calibration and hedging under jump diffusion. *Review of Derivative Research* 9, 1–35.

Heston, S. L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Review of Financial Studies* 6, 327–343.

Hull, J. and A. White (1987). The pricing of options on assets with stochastic volatilities. *Journal of Finance* 42, 281–300.

Merton, R. C. (1976). Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics* 3, 125–144.

Naik, V. and M. Lee (1990). General equilibrium pricing of options on the market portfolio with discontinuous returns. *Review of Financial Studies* 3, 493–521.

Orosi, G. (2010). Improved implementation of local volatility and its application to s&p 500 index options. *Journal of Derivatives* 17(3), 53–64.

Rubinstein, M. (1994). Implied binomial trees. *Journal of Finance* 49, 771–818.

Shimko, D. (1993). Bounds of probability. *Risk* pp. 33–37.

Vapnik, V. N. (1998). *Statistical Learning Theory.* John Wiley & Sons, A Wiley Inter-Science Publication.

A Kernels Generating Splines with an Infinite Number of Knots

Here we briefly describe kernels generating splines with an infinite number of knots. The presentation follows from discussion in §11.6.2 in Vapnik (1998). Suppose that we want to approximate a one-dimensional function of one variable s defined on the interval $[-b, +\infty)$, $0 < b < \infty$, by splines of order $d \geq 0$ with infinite number of knots: $\{t_i\}$, $1 \leq i < \infty$. First the one-dimensional variable s is mapped into a vector u in the feature space of an infinite-dimension:

$$s \to u = (1, s, \cdots, s^d, (s - t_1)_+^d, \cdots, (s - t_i)_+^d, \cdots)$$

where

$$(s - t_k)_+^d = \begin{cases} 0 & \text{if } s \leq t_k, \\ (s - t_k)^d & \text{if } s > t_k. \end{cases}$$

Then the spline has the form:

$$g(s) = \sum_{i=0}^{d} a_i s^i + \int_{-b}^{+\infty} a(t)(s - t)_+^d dt,$$

where a_i, $i = 0, ..., d$ and $a(t)$ are coefficients of expansion. The kernel generating the spline can be obtained by determining the inner product as follows

$$\mathcal{K}(s_j, s_i) = \int_{-b}^{+\infty} (s_j - t)_+^d (s_i - t)_+^d dt + \sum_{k=0}^{d} s_j^k s_i^k$$

For the linear spline with $d = 1$ in particular, we have the following function representation for the kernel generating spline:

$$\mathcal{K}(s_j, s_i) = 1 + s_j s_i + \frac{1}{2} \mid s_j - s_i \mid (s_j \wedge s_i + b)^2 + \frac{(s_j \wedge s_i + b)^3}{3}$$

where s_j, s_i are training data points in the interval $[-b, +\infty)$, and $s_j \wedge s_i$ denotes $\min(s_j, s_i)$. It can shown that the above kernel function is twice differentiable.