Volatility Calibration Using Spline and High Dimensional Model Representation Models

Balazs Feil, Sergei Kucherenko, and Nilay Shah
Imperial College London,
e-mail: s.kucherenko@ic.ac.uk

Abstract
The local volatility function is approximated using two different models: a bicubic spline and High Dimensional Model Representation (HDMR) model. For a bicubic spline the local volatility values σ_i at a chosen discrete set of knots are determined by minimizing the least square error between market and model option prices. Model option prices are found using the one-factor diffusion process for the underlying asset. For the HDMR model, the parameters to be optimized are the parameters of the HDMR model. Smoothness of the approximation functions is crucial to overcome the ill-posedness of the volatility calibration problem. Two models are compared using two test cases. It is shown that the HDMR model can produce more accurate results than the cubic spline model, and it is also cheaper to run. It is demonstrated that the considered approach can accurately reproduce not only option prices but Greeks as well.

Keywords
volatility calibration, local volatility, Dupire equation, global optimization, High Dimensional Model Representation (HDMR)

1 Introduction

Volatility is one of the most important market variables in financial practice and theory. It is not directly observable in the market. The Black–Scholes model can be used to estimate volatility from observable option prices. By inverting the Black–Scholes formula with option market data such estimates, known as implied volatility, can be obtained. They show strong dependence of volatility values on strike price and time to maturity. This dependence, known as the volatility smile, cannot be captured in the Black–Scholes model.

The constant implied volatility approach, which uses different volatilities for options with different strikes and maturities, works well for pricing simple European options but it fails to provide adequate solutions for pricing exotic or American options. This approach also produces incorrect hedge factors (Gamma, Vega, Delta, etc.) even for simple options. One of the practically used approaches is to use a one-factor diffusion process with a volatility function depending both on the asset price and time: $\sigma(S_t, t)$. This is a deterministic approach and $\sigma(S_t, t)$ is known as the local volatility or the forward volatility (Wilmott, 2006). A different class of methods is based on the stochastic volatility models (see Wilmott, 2006, Gatheral, 2006 for details).

It was shown in Dupire (1994) and Andersen and Brotherton-Ratcliffe (1998) that the local volatility function can be uniquely determined from the European call options of all strikes and maturities. In practice, the market European option prices are limited to a finite discretely spaced set of call prices. A possible solution to this problem is to interpolate and extrapolate missing call price data. It is easy to show that this approach potentially introduces non-market information into the data. A volatility reconstruction problem (also known as a local volatility calibration) belongs to a class of inverse problems. It is ill-posed (a small change in the input can result in a large change in the output) and, as a result, there are an infinite number of solutions $\sigma(S_t, t)$ that match the market option price data. The Greeks representing the sensitivities of options to a change in underlying parameters are very sensitive to the reconstructed volatility surface profiles and they cannot be correctly calculated without regularization of the ill-posed problem.

Many early papers on local volatility calibration such as Derman and Kani (1994), Jackwerth and Rubenstein (1996), and Bouchouev and Isakov (1997) used an assumption of a continuum complete set of European call option prices, which has severe shortcomings, namely a necessity of interpolation and extrapolation of missing call price data. Tikhonov regularization was applied to stabilize the inverse problem in Jackson et al. (1998), Crépey (2003), Egger and Engl (2005), and Hanke and Rossler (2005).

Published online in Wiley InterScience | DOI: 10.1002/wilj.18
(www.interscience.wiley.com)

A different approach is to use smooth functions for approximation of the local volatility surface. Coleman et al. (1999) used a bicubic spline with the local volatility values σ_i at chosen discrete set of knots. Values of $\{\sigma_i\}$ are determined by minimizing the least square error between market and model option prices. Model option prices are found by solving the Black–Scholes equations. It was shown that smoothness of the approximation functions is crucial in order to overcome the ill-posedness of the volatility calibration problem. Other authors (Kim et al., 2006) used radial basis functions instead of splines for volatility approximation. Ben Hamida and Cont (2005) used the evolutionary algorithm for optimization instead of more commonly used gradient-based methods. They argued that since the evolutionary algorithm does not require differentiability of the objective function, instead of the quadratic pricing error the absolute pricing error which yields more stable numerical values can be used. Significant efficiency increase was achieved by using the Dupire equation (Dupire, 1994) for evaluation of the model option prices. It enabled a speed-up by a factor equal to the number of options being calibrated in comparison with the approach based on solving a set of the Black–Scholes equations.

Coleman et al. (2003) compared the performance of dynamic hedging using the constant volatility method, the implied volatility method, and the volatility function method based on the bicubic spline approximation. For a synthetic European option example and the S&P 500 futures option market data they demonstrated that the volatility function method yields significantly more accurate hedge factors and smaller hedge errors. It also performs significantly better in dynamic hedging when compared with the constant and implied volatility methods.

In this work we use the High Dimensional Model Representation (HDMR) for local volatility function approximation. HDMR uses an ANOVA type decomposition of the original functions and polynomial approximations of the component functions. Generally the coefficients of the polynomial decomposition are found by Monte Carlo integration, hence this method is known as RS (Random Sampling)–HDMR. The improved RS-HDMR method based on QMC sampling (QRS–HDMR) was developed in Feil et al. (2009). In the framework of the volatility calibration problem the coefficients of the polynomial decomposition are unknowns which are found by solving the least squares minimization problem.

This paper is organized as follows: The next section gives a general problem formulation and introduces spline and HDMR approximations of local volatility functions. It also gives a description of how model option values are obtained using the Black–Scholes and Dupire PDEs. Section 3 presents the details of numerical solution techniques. A brief description of the developed software is given in Section 4. Section 5 presents a comparison of the spline and HDMR models for two case problems: an absolute diffusion problem which has a closed form solution (for which a reconstructed volatility surface can be compared with the actual volatility surface), and the real market European call option data on the S&P 500 stock index. Finally, conclusions are presented in Section 6.

2 Problem Formulation

2.1 Local Volatility Model

It is assumed that the underlying asset follows a continuous one-factor diffusion process:

$$dS_t = \mu(S_t, t)S_t dt + \sigma(S_t, t)S_t dW_t \tag{1}$$

for some fixed time horizon $t \in [0, T]$. Here W_t is a Brownian motion, $\mu(S_t, t)$ is the drift and $\sigma(S_t, t)$ is a local volatility function which is assumed to be continuous and sufficiently smooth so that equation (1) with corresponding initial and boundary conditions has a unique solution. Model (1) is calibrated with a given set of option prices to approximate the local volatility function $\sigma(S_t, t)$ using a spline or some other polynomial approximation as explained in the next section.

2.2 Spline Approximation

Following Coleman et al. (1999), the local volatility function is represented by a bicubic spline (see Appendix A.1 for details) which is computed by solving an inverse box-constrained nonlinear optimization problem as follows. Let $\{\overline{V}_j\}$, $j = 1, \ldots, m$ denote the m given market option prices. Given $\{S_i, t_i\}, i = 1, p$, $p \le m$ spline knots with corresponding local volatility values $\sigma_i = \sigma(S_i, t_i)$ an interpolating cubic spline $c(S, t, \sigma)$ with a fixed end condition (e.g., the natural spline end condition) is uniquely defined by setting $c(S_i, t_i) = \sigma_i, i = 1, \ldots, p$. The local volatility values σ_i at a set of fixed knots are determined by solving the following minimization problem:

$$\min_{\overline{\sigma} \in \Re^p} f(\overline{\sigma}) = \frac{1}{2} \sum_{j=1}^{m} w_j (V_j(\overline{\sigma}) - \overline{V}_j)^2 \tag{2}$$

$$\text{subject to } l_i \le \sigma_i \le u_i \qquad \forall i = 1, \ldots, p.$$

Here $V_j(\overline{\sigma}) = V_j(c(S, t), K_j, T_j)$, $\overline{V}_j$'s are given option prices at given strike price/expiration time (K_j, T_j), $j = 1, \ldots, m$ pairs, w_j's are weights, σ_i, $i = 1, \ldots, p$ are the model parameters, l's and u's are lower and upper bounds, respectively. $\{V_j(\overline{\sigma})\}$ are model-based option values. It is explained in Section 3 how they are evaluated.

To guarantee an accurate and appropriate reconstruction of the local volatility, the number of spline knots p should be no greater than the number of available market option prices m.

2.3 High Dimensional Model Representation

One of the very promising developments of model analysis is the replacement of complex models and models which need to be run repeatedly on-line with equivalent operational meta models. For many practical problems only low order correlations of the input variables are important. By exploiting this feature, an

Published online in Wiley InterScience | DOI: 10.1002/wilj
(www.interscience.wiley.com)

efficient set of techniques called HDMR was developed by Rabitz and co-authors (Li et al., 2002).

The HDMR model can be used to approximate the local volatility surface (see Appendix A.2 for definition of HDMR). In this case the structure of the problem remains the same; the only difference is that the parameters to be optimized are the parameters of the HDMR model. For the local volatility function the HDMR model has the following form:

$$\sigma(S,t) = \sigma_0 + \sum_{r=1}^{k_S} \alpha_r^S \phi_r(S) + \sum_{r=1}^{k_t} \alpha_r^t \phi_r(t) + \sum_{p=1}^{l_S}\sum_{q=1}^{l_t} \beta_{pq}^{St}\phi_{pq}(S,t).$$

Here, $\phi_r(x_i)$,$\phi_{pq}(x_i,x_j)$ are sets of one and two-dimensional basis functions, and α_r^i and β_{pq}^{ij} are the polynomial coefficients to be found from solving optimization problems. To make the model more robust, the constant σ_0 is set to be the average implied volatility, and it is not considered as a parameter to be optimized.

The volatility calibration problem in this case has a form

$$\min_{\{\overline{\alpha},\overline{\beta}\}\in\Re^p} f(\overline{\alpha},\overline{\beta}) = \frac{1}{2}\sum_{j=1}^{m} w_j (V_j(\overline{\alpha},\overline{\beta}) - \overline{V}_j)^2,$$

$$\text{subject to} \quad \begin{array}{ll} \alpha_i^{S,L} \le \alpha_i^S \le \alpha_i^{S,U} & \forall i = 1,\ldots,k_s, \\ \alpha_i^{t,L} \le \alpha_i^t \le \alpha_i^{t,U} & \forall i = 1,\ldots,k_t, \\ \beta_{pq}^{St,L} \le \beta_{pq}^{St} \le \beta_{pq}^{St,U} & \forall p = 1,\ldots,l_s, \forall q = 1,\ldots,l_t \end{array}$$

Here $\alpha_i^{S,L}$, $\alpha_i^{t,L}$, $\beta_{pq}^{St,L}$ and $\alpha_i^{S,U}$, $\alpha_i^{t,U}$, $\beta_{pq}^{St,U}$ are given lower and upper bounds for the polynomial coefficients.

2.4 Calculation of Option Values

Estimation of the objective fuction $f(\overline{\sigma})$ in (2) requires the calculation of $V_j(\overline{\sigma})$ for each pair (K_j, T_j), $j = 1,\ldots,m$. It can be done by solving the Black–Scholes equation

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma(S,t)^2 S^2 \frac{\partial^2 V}{\partial S^2} + (r-q)S\frac{\partial V}{\partial S} - rV = 0 \tag{3}$$

with the following boundary and initial conditions (for the European call):

$$\begin{aligned} &\lim_{S\to\infty} \frac{\partial V(S,t)}{\partial S} = \exp(-q(T-t)), t \in [0,T], \\ &V(S=0,t) = 0, t\in[0,T], \\ &V(S,T) = \max(S-K,0). \end{aligned} \tag{4}$$

Here $V(S,t)$ denotes the option value of an underlying asset with strike price K and expiry date T at $(S,t), t \in [0,T]$. It is assumed that S_{init} (spot price), r (risk free interest rate), q (dividends), and $\sigma(S,t)$ are given.

This approach requires solution of the problem (3)–(4) m times at each iteration of the optimization method. There is a more efficient approach which requires only one solution of the Dupire equation (Dupire,1994):

$$\frac{\partial V}{\partial T} - \frac{1}{2}\sigma(K,T)^2 K^2 \frac{\partial^2 V}{\partial K^2} + (r-q)K\frac{\partial V}{\partial K} = 0. \tag{5}$$

Unlike the Black–Scholes equation this is a forward equation. The boundary and initial conditions for the European call have the following form:

$$\begin{aligned} &V(K,T) = 0, K >> S_{\text{init}}, T\in[0,T_{\max}], \\ &V(K,T) = S_{\text{init}} - K, K\approx 0, T\in[0,T_{\max}], \\ &V(K,0) = \max(S_{\text{init}} - K, 0). \end{aligned} \tag{6}$$

All option prices $V_j(\overline{\sigma})$ for (K_j,T_j), $i = 1,\ldots,m$ pairs can be obtained with one solution of the problem (5),(6).

3 Numerical Solution Techniques

Bicubic and HDMR approximations for the volatility surface are discussed in Appendix A. Both the Black–Scholes and the Dupire equations are solved by the finite difference method using the implicit time integration scheme (Wilmott [2006]). The details of the finite difference approximation scheme are given in Appendix B.

A uniform grid with $N \times M$ points in the region $[0;\ 2S_{\text{init}}] \times [0;\ \tau]$, where τ is the maximum maturity in the market option data, is used for solving the discretized Black–Scholes or the Dupire equations. The spline knots are also chosen to be on a uniform rectangular mesh covering the region Ω in which the volatility values are significant in pricing the market options (the region Ω covers not far out of the money and in the money region of S). This region is not known explicitly, therefore a grid $[\delta_1 S_{\text{init}}, \delta_2 S_{\text{init}}] \times [0,\tau]$ is used. Parameters δ_1 and δ_2 are chosen to be in the regions $\delta_1 \in [0.6, 0.8]$ and $\delta_2 \in [1.4, 1.6]$.

For the HDMR model, similarly to the spline based model the variable volatility model is used only in the ''important region'' Ω. Outside this region, a constant volatility is used, namely $\sigma = \sigma(\delta_1 S_{\text{init}}, t)$ at $S < \delta_1 S_{\text{init}}$, and $\sigma = \sigma(\delta_2 S_{\text{init}}, t)$ at $S > \delta_2 S_{\text{init}}$ for all t.

The general scheme of the volatility calibration algorithm is as follows (we consider the case of the Dupire equation. The algorithm remains the same for the case of the Black–Scholes equation):

Step 1. Set the uniform grid for solving the discretized Dupire equation.
Step 2. Set the spline knots.
Step 3. Set the initial values for the local volatility at chosen spline knots $\{\sigma_i^{(0)}\}$, $i = 1,\ldots,p$.

Published online in Wiley InterScience | DOI: 10.1002/wilj
(www.interscience.wiley.com)

Step 4. Solve the discretized Dupire equation with the local volatility function approximated by spline with the local volatility values at chosen spline knots $\{\sigma_i^{(k)}\}$, $i = 1, \ldots, p$. Obtain the set of model based option values $\{V_j(\overline{\sigma})\}$ and the value of the objective function $f(\overline{\sigma})$ for the minimization problem (2).
Step 5. Using the objective function $f(\overline{\sigma})$ and the chosen minimization algorithm find the undated values of the local volatility at spline knots $\{\sigma_i^{(k+1)}\}$.
Step 6. Repeat steps 4–5 until a local minimum is found. The local minimizer $\{\sigma_i^*\}$, $i = 1, \ldots, p$ defines the local volatility function.

For the case of the HDMR model the unknown variables with respect to which the minimization problem is solved $\{\sigma_i^{(k)}\}$, $i = 1, \ldots, p$ in the presented algorithm are replaced with the sets of $\{\alpha_r^i\}$ and $\{\beta_{pq}^{ij}\}$ variables as explained in Section 2.3.

4 Description of the Developed Software

The proposed methods were implemented in a C++ application program *SobolVol*. Four types of models are supported:

1. the 2D spline model similar to the one proposed by Coleman et al. (1999) (referred to in the following text as ''spline'');
2. the spline based model using a set of 1D splines similar to the model proposed by Jackson et al. (1999) (referred to as ''1D spline'');
3. full 2D HDMR model (referred as ''HDMR'');
4. the HDMR-based model similar to Jackson's, but with a set of 1D HDMR models in K direction and linear interpolation between a fixed set of T_i points (referred to as ''1D HDMR'').

There is a choice of two optimization methods:

1. the local Levenberg-Marquardt (LM) method (Flannery et al., 2005);
2. the global SobolOpt method (Kucherenko and Sytsko, 2005).

The Black–Scholes and Dupire equations contain only the square of the volatility value, therefore it is natural to approximate variance $\sigma^2(K,T)$ rather than $\sigma(K,T)$ using spline or HDMR approximations. Volatility $\sigma(K,T)$ then can be found simply by taking a square root of $\sigma^2(K,T)$. As far as we know, only $\sigma(K,T)$ has been used in the volatility reconstruction setting so far. *SobolVol* has an option to choose either $\sigma(K,T)$ or $\sigma^2(K,T)$ surfaces for reconstruction.

SobolVol uses market (K_i,T_i,V_i) data to reconstruct local volatility. It can also use a predefined volatility function to test the methods when the original real local volatility surface is known (*f.e.* given by an analytical formula).

Besides the volatility reconstruction, *SobolVol* is able to compute the Greeks along a predefined (K,T) mesh using the real volatility model (if it exists), the reconstructed volatility model, the implied volatility or a constant volatility model. Examples of the input files for the *SobolVol* program are given in Appendix C.

5 Numerical Results

We consider two test cases. The first one is an absolute diffusion problem

$$\frac{dS_t}{S_t} = \mu(S_t,t)dt + \frac{C}{S_t}dW_t,$$

where $C > 0$ is a constant. Here volatility $\sigma(S_t,t) = C/S_t$ is only a function of underlying and it does not depend on time. There is a closed form solution for this problem; therefore a reconstructed volatility surface can be compared with the actual volatility surface. The second problem uses the real market European call option data on the S&P 500 stock index with 70 call options (70 pairs of strikes and maturities (K_j,T_j)). We use the same data as Coleman et al. (1999) to be able to compare and benchmark the results.

5.1 An Absolute Diffusion Problem

Following Coleman et al. (1999) the following parameters are used: the local volatility function $\sigma(S,t) = 15/S$, the initial stock index $S_{\text{init}} = 100$, the risk free interest rate $r = 0.05$, and the dividend rate $q = 0.02$. 22 European call options are considered as market option data. Half of them have half year maturity with strike prices [90:2:110] (middle figure is a step increment), and another half have one year maturity with the same strikes. The option strike and maturity values are as follows:

$$K = [90;92;\ldots;110;90;92;\ldots;110],$$
$$T = [0.5;0.5;\ldots;0.5;1;1;\ldots;1].$$

To show how the reconstructed local volatility depends on the choice of using $\sigma(K,T)$ or $\sigma^2(K,T)$ for approximation and the number of 1D HDMR models we considered a few different cases. The summary of the results is given in Tables 1 and 2.

5.1.1 Spline-based Volatility Reconstruction

For the absolute diffusion problem, an approach with two 1D models gives good agreement with the analytical formula solution. However, for three 1D spline models along the time direction T (at $T = 0$, $T = 0.5$ and $T = 1$), the reconstructed surface is not flat in the time direction T as it should be (compare Figures 1 and 2), although the cost function values are very low in both cases (Table 2). The surface becomes more distorted if more 1D models are used.

The results were obtained using a local optimization method. The local optimization approach requires an initial starting point to start the minimization procedure. The volatility reconstruction problem is an inverse problem with potentially multiple solutions and as such can be very sensitive to the initial starting point.

Numerical experiments made with the usage of the global optimization routine *SobolOpt* suggest that there exist many local minima but it is not guaranteed that the global minimum gives

Published online in Wiley InterScience | DOI: 10.1002/wilj
(www.interscience.wiley.com)

Table 1: Summary of the different test cases.

Name	Data set	σ^2	Model	Mesh for K	Mesh for T	Init. value
Case 1a	S&P	no	spline	Kmesh 1	Tmesh	0.15
Case 1a (σ^2)	S&P	yes	spline	Kmesh 1	Tmesh	0.15^2
Case 1b	S&P	no	spline	Kmesh 2	Tmesh	0.15
Case 1b (σ^2)	S&P	yes	spline	Kmesh 2	Tmesh	0.15^2
Case 2	S&P	no	spline	given K_i	given T_i	given σ_i
Case 2 (σ^2)	S&P	yes	spline	given K_i	given T_i	given σ_i^2
Case 3a	S&P	no	spline	Kmesh 1	Tmesh	$\overline{\sigma}_{imp}$
Case 3a (σ^2)	S&P	yes	spline	Kmesh 1	Tmesh	$\overline{\sigma}^2_{imp}$
Case 3b	S&P	no	spline	Kmesh 2	Tmesh	$\overline{\sigma}_{imp}$
Case 3b (σ^2)	S&P	yes	spline	Kmesh 2	Tmesh	$\overline{\sigma}^2_{imp}$
Case 3c	S&P	no	spline	given K_i	Tmesh	$\overline{\sigma}_{imp}$
Case 3c (σ^2)	S&P	yes	spline	given K_i	Tmesh	$\overline{\sigma}^2_{imp}$
Case 4	abs.diff.	no	1D spline	Kmesh 3	[0 0.5 1]	0.15
Case 4 (σ^2)	abs.diff.	yes	1D spline	Kmesh 3	[0 0.5 1]	0.15^2
Case 5	abs.diff.	no	1D hdmr	—	[0 1]	0
Case 5 (σ^2)	abs.diff.	yes	1D hdmr	—	[0 1]	0
Case 6	abs.diff.	no	1D hdmr	—	[0 0.5 1]	0
Case 6 (σ^2)	abs.diff.	yes	1D hdmr	—	[0 0.5 1]	0
Case 7	S&P	no	1D hdmr	10 polynomials	Tmesh	0
Case 7 (σ^2)	S&P	yes	1D hdmr	10 polynomials	Tmesh	0
Case 8	S&P	no	hdmr	$k_K = k_T = 8$	$l = l' = 4$	0
Case 8 (σ^2)	S&P	yes	hdmr	$k_K = k_T = 8$	$l = l' = 4$	0
Case 9	S&P	no	hdmr	$k_K = k_T = 10$	$l = l' = 5$	0
Case 9 (σ^2)	S&P	yes	hdmr	$k_K = k_T = 10$	$l = l' = 5$	0

Tmesh: 0 : 0.33 : 2
Kmesh 1: [0.8 : 0.066: 1.4] S_{init}(used by Coleman et al. [1999])
Kmesh 2: [0.1; 0.83; 0.93:0.066:1.26; 1.36; 1.9]S_{init}
Kmesh 3: [0 : 0.33 : 2] S_{init}

a better approximation to the real volatility values than one of the local minima. Ben Hamida and Cont (2005) commented that there is no justification for looking for a solution that is more accurate than the uncertainty in the option values (e.g., given by the bid-ask spread).

5.1.2 HDMR-based Reconstruction

The 1D HDMR approach (model type 4) gave good results compared to the spline or 1D spline approaches. The mean value of the models σ_0 was fixed being equal to the average implied volatility value. The initial conditions (values of α_r^i and β_{pq}^{ij} coefficients) were taken to be equal to 0, i.e. the initial model was constant. The cost function values turned out to be very sensitive to the model structure (polynomial degrees, location of the 1D models, values used for normalization). The model structure seems to be more important for obtaining good results than that for the spline based approach, especially for the S&P data set (the results are presented in the following sections).

When the 1D HDMR model was used with just two 1D models, the reconstructed surface is very similar to the actual one (compare Figures 1 and 3). Similar to the 1D spline model case, if more 1D models are used the volatility surface gets more distorted (see Figure 4 with three 1D models).

It was not possible to reconstruct the local volatility surface reliably and accurately enough using the full 2D HDMR based model, therefore the 1D HDMR approach was used for all calculations.

5.1.3 Greeks and Hedging

The Greeks are very sensitive to the reconstructed volatility surface profiles. Following Coleman et al. (1999) we used the absolute diffusion volatility model given by the formula $\sigma(S,t) = 75/S$. The spot asset price $S_{\text{init}} = 100$ and the risk free interest rate $r = 0.05$ were the same as in the test case above. The dividend rate q was taken to be equal to 0. European call options with strikes and maturities at the grid $[80 : 4 : 120] \times [0.25, 0.5, 1]$ were considered. The same spline knots as in Coleman et al. (1999), namely $[0 : 20 : 2S_{\text{init}}] \times [0, 0.5, 1]$ were used. Figure 5 shows the option price and Greeks with maturity $T = 0.25$ year using the true local volatility, reconstructed volatility, implied volatilities and constant volatility (equal to the average implied volatility $\overline{\sigma} = 0.7515$). Our results show very good agreement between true and reconstructed values for the Greeks. However, the usage of

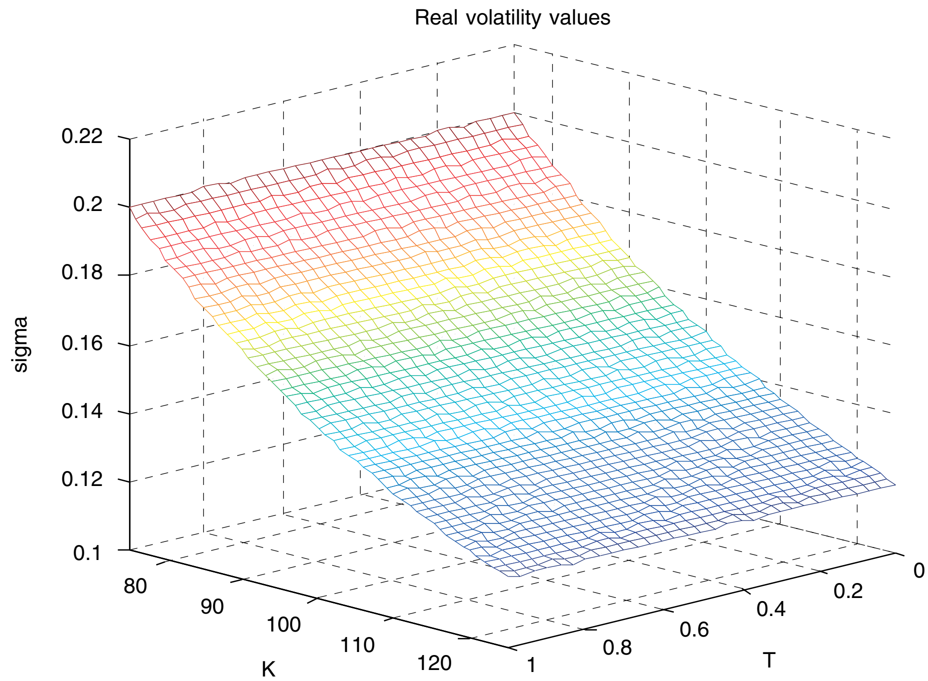


Figure 1: Real volatility values given by an analytical formula for the absolute diffusion problem.

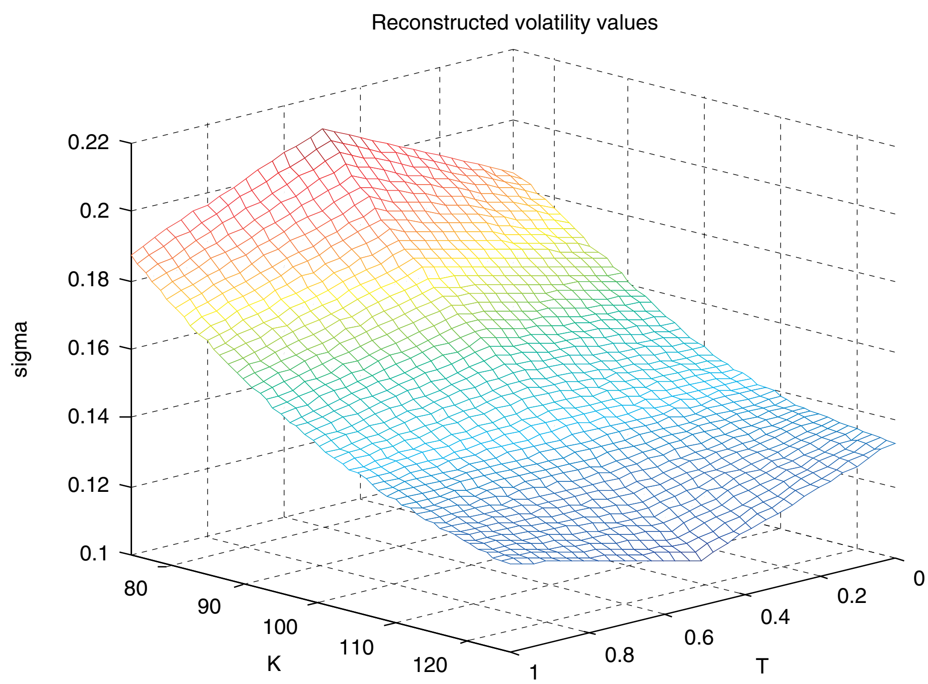


Figure 2: Reconstructed local volatility surface with the 1D spline approach using 3 1D models. Average implied volatility equal to 0.149 is used as an initial condition. (Case 4, Table 1).

implied volatility and constant volatility models can lead to large errors in Greeks calculations.

5.2 S&P Example

In this section we consider a problem of approximating the local volatility function from the S&P 500 European index call options data. The same option data of October 1995 with no more than two years maturity as in Coleman et al. [1999] was used. The initial index is $S_{\text{init}} = 590$; values of other parameters are: $r = 0.06$, $q = 0.0262$.

To show how the reconstructed local volatility depends on the choice of spline knots for the spline model and the

Published online in Wiley InterScience | DOI: 10.1002/wilj
(www.interscience.wiley.com)

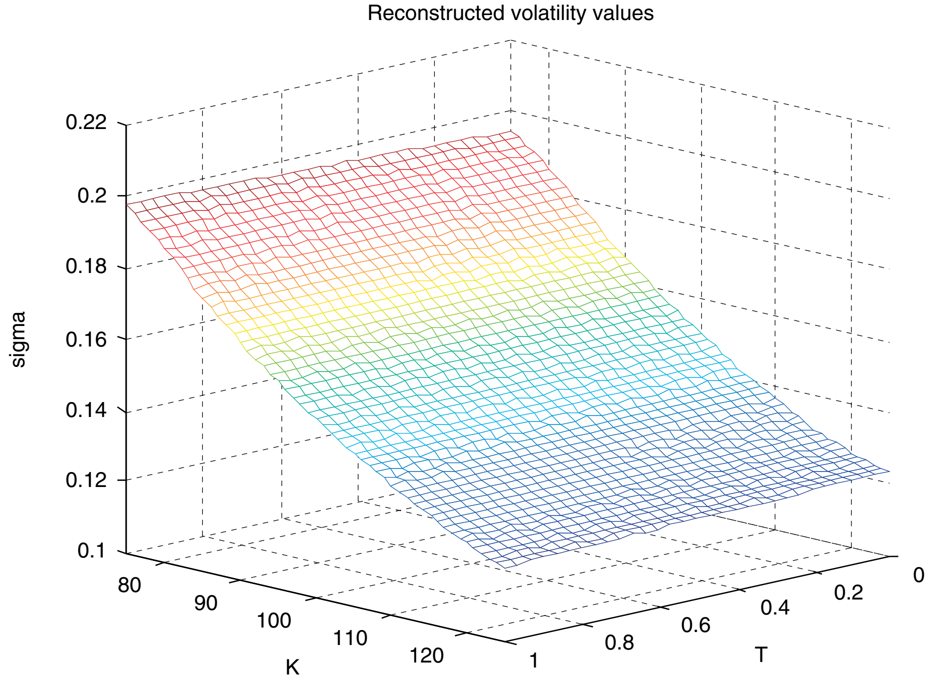


Figure 3: Reconstructed local volatility surface with the 1D HDMR approach using two 1D models with 10 polynomials (for each of 1D models). The initial condition is taken to be 0.0 for all coefficients (Case 5, Table 1).

number of 1D HDMR models and polynomial orders for the HDMR models, we solved a few cases with different sets of parameters. The summary of the results is given in Tables 1 and 2. Figure 6 displays reconstructed local volatility surfaces. A comparison of cases 1–3 (the spline model) shows that for chosen sets of spline knots and initial conditions the reconstructed local volatility surfaces look similar for cases in which $\sigma(K,T)$ was chosen for approximation. For cases of the $\sigma^2(K,T)$ approximation model some volatility surfaces look unrealistically flat (cases 2, 3a, 3c). However, for these cases the values of objective functions are quite high which flags bad solutions (Table 2).

Figure 6 shows that the reconstructed volatility surfaces for cases of the σ approximation model (apart from the case 8) are similar to each other and to the results presented in Coleman et al. [1999]. The biggest difference can be found in the region near to the maximum maturity $T = 2$. The cost function values are low for all cases. The situation is different for the $\sigma^2(K,T)$ approximation model: only cases 1b and 3b show good agreement with other good profiles for $\sigma(K,T)$. Only for these cases, values of the cost (error) function are low.

Comparison of the CPU times shows that using $\sigma^2(K,T)$ for approximation results in much shorter CPU times. This effect is easy to explain for the HDMR model: assuming the same number of polynomial terms in decompositions of $\sigma(K,T)$ and $\sigma^2(K,T)$, for the $\sigma(K,T)$ approximation model the number of terms for the case of $\sigma(K,T)$ approximation grows quadratically as only variance $\sigma^2(K,T)$ is used in PDEs (3) or (5). It results in higher number of degrees of freedom in the optimization problem than for

Table 2: Comparison for the approximation of σ and σ^2.

Name of case	Value of objective function		CPU time (sec)	
	σ	σ^2	σ	σ^2
Case 1a	0.0016	1.3320	4515	256
Case 1b	0.0032	0.0379	2104	754
Case 2	$1.40{\cdot}10^{-10}$	$4.05{\cdot}10^{+32}$	525	4
Case 3a	0.0083	6.2570	81	56
Case 3b	0.0036	0.1355	4447	588
Case 3c	0.0016	6.8300	545	48
Case 4	$5.71{\cdot}10^{-11}$	$3.98{\cdot}10^{-11}$	<1	<1
Case 5	$4.54{\cdot}10^{-10}$	$2.19{\cdot}10^{-10}$	<1	<1
Case 6	$3.61{\cdot}10^{-10}$	$1.59{\cdot}10^{-11}$	≈ 1	≈ 1
Case 7	0.0109	5.4321	24	14
Case 8	0.0118	5.7236	280	126
Case 9	0.0063	4.6718	5470	262

the case of the straightforward HDMR approximation of $\sigma^2(K,T)$. However, solutions are more sensitive to the initial conditions and the model parameters for the $\sigma^2(K,T)$ approximation model. For the absolute diffusion problem both approximation models give good results. However, for the S&P data set the $\sigma^2(K,T)$ approximation model gives acceptable results in only one case out of six cases for which the spline model was used.

5.2.1 Greeks

The Greeks for Case 3a (spline model) and Case 7 (1D HDMR model) are presented in Figures 7 and Fig. 8, respectively. The

Published online in Wiley InterScience | DOI: 10.1002/wilj
(www.interscience.wiley.com)

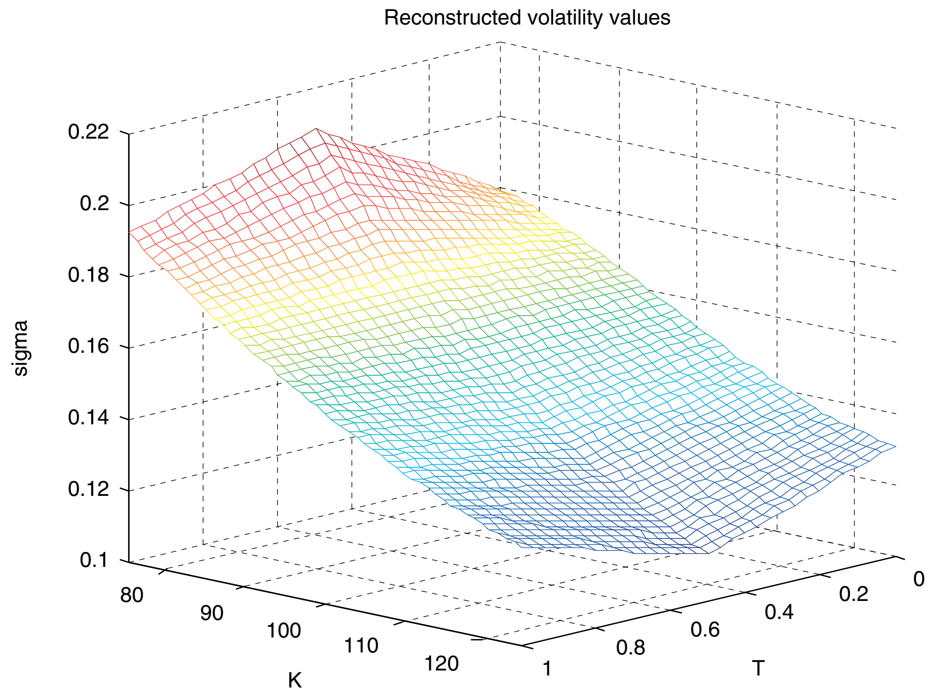


Figure 4: Reconstructed local volatility surface using the 1D HDMR approach using three 1D models with 7 polynomials (for each of them). The initial condition is taken to be 0.0 for all coefficients (Case 6, Table 1).

constant volatility here means the average implied volatility value, which is 0.1319 in this case. As the figures show, although the option price curves are almost identical for the constant and reconstructed volatility models, the differences in the hedge factors are considerable.

The Greeks are similar for the two reconstructed volatility models, but the solution obtained with the 1D HDMR model seems to be slightly better especially for the Vega hedge factor as it does not show a minor hump present in the spline based model. The HDMR model is also almost four times faster in terms of the CPU time.

6 Conclusion

A novel method based on the HDMR model for approximation of local volatility functions is developed. Volatility calibration is performed using minimization of the least square error between market and model option prices. Model option prices are obtained using the Dupire equations. A similar approach is used for volatility calibration based on the cubic spline model. Two models are compared using two test cases. It is shown that a 1D HDMR model can produce more accurate results than the cubic spline model. This model is also faster in terms of the CPU time. It is demonstrated that considered approach can accurately reproduce not only option prices but Greeks as well.

Appendix A Approximation of Volatility Surfaces

A. 1 Bicubic spline

The bicubic spline is defined over a rectangle R in the (K, T) plane, the sides of R being parallel to the K- and T-axes. R is divided into rectangular panels by lines parallel to the axes. Over each panel the bicubic spline is a bicubic polynomial which can be presented in the following form

$$\sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij}K^iT^j.$$

Each of these polynomials joins the polynomials in adjacent panels with continuity up to the second derivative. The constant K-values of the dividing lines parallel to the T-axis form the set of interior knots for the variable K, corresponding to the set of interior knots of a cubic spline. Similarly, the constant T-values of dividing lines parallel to the K-axis form the set of interior knots for the variable T. Instead of representing the bicubic spline in terms of the above set of bicubic polynomials, in practice it is represented for the sake of computational speed and accuracy in the form

$$spline(K, T) = \sum_{i=1}^{p}\sum_{j=1}^{q} c_{ij}M_i(K)N_j(T),$$

Published online in Wiley InterScience | DOI: 10.1002/wilj (www.interscience.wiley.com)

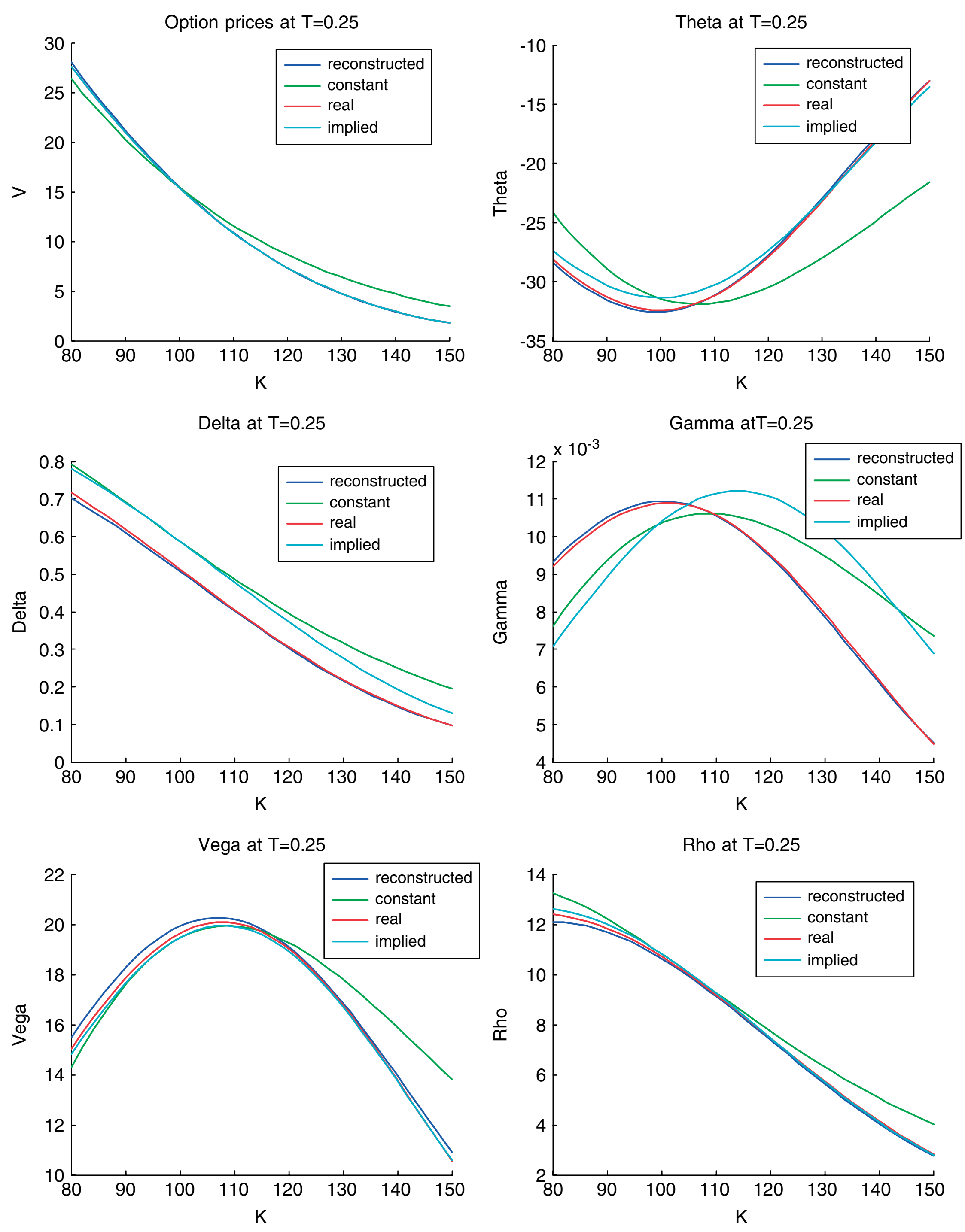


Figure 5: The Greeks with the true, reconstructed, implied, and constant volatility functions for the absolute diffusion problem with T = 0.25.

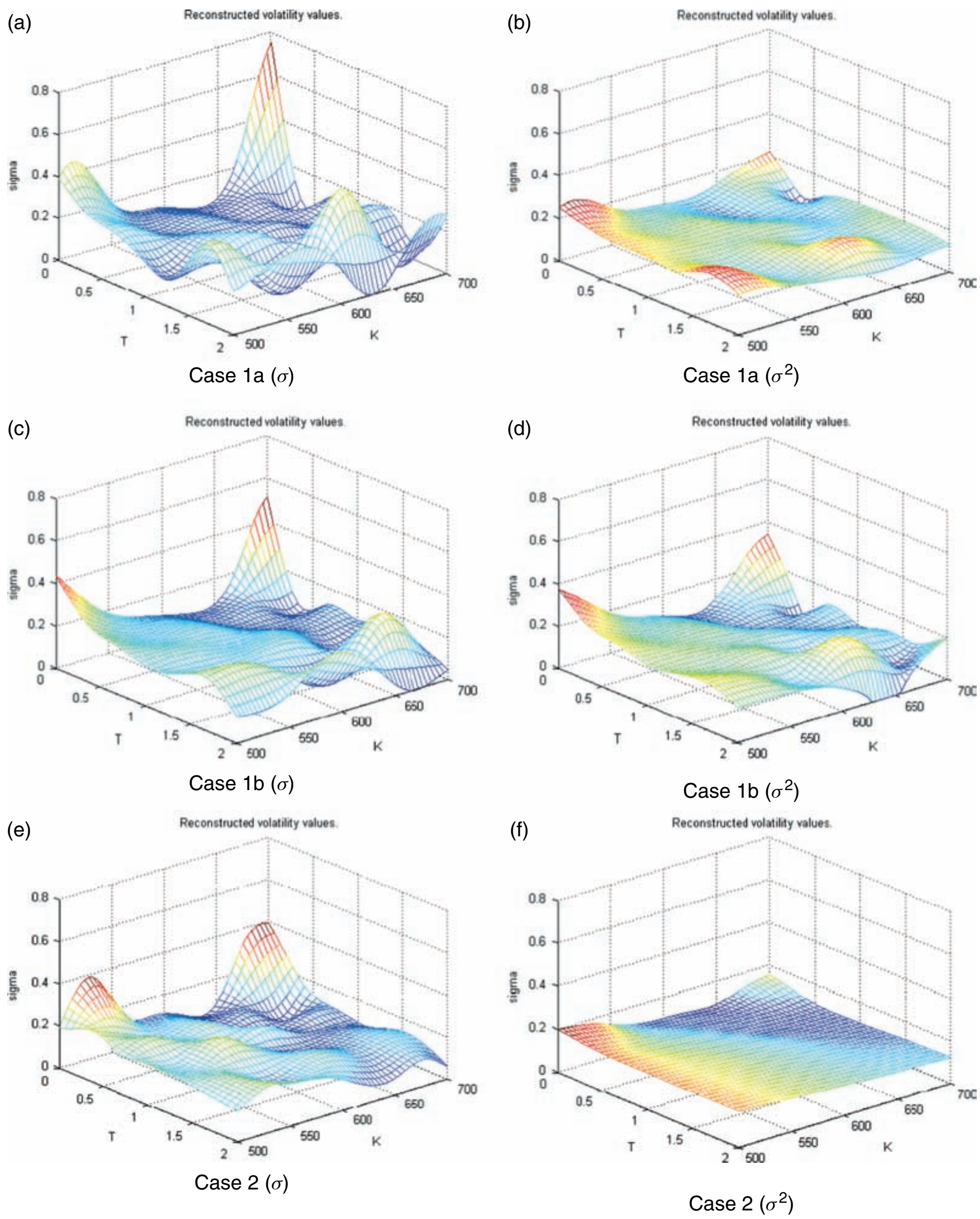


Figure 6: The reconstructed volatility models for the S&P test cases.

where $M_i(K), i = 1, 2, \ldots, p$, and $N_j(T), j = 1, 2, \ldots, q$, are normalized B-splines (see Hayes and Halliday (1974) for further details of bicubic splines and Hayes (1974) for normalised *B*-splines). The *SobolVol* uses spline routines from the NAG C Library (NAG C Library).

A. 2 High Dimensional Model Representation (HDMR)

The High Dimensional Model Representation (HDMR) also known as ANOVA in a general n dimensional case is a finite hierarchical correlated function expansion in terms of the input variables $\mathbf{x} = (x_1, \ldots, x_n)$:

$$f(\mathbf{x}) = f_0 + \sum_{i=1}^{n} f_i(x_i) + \sum_{1 \le i < j \le n} f_{ij}(x_i, x_j) + \cdots + f_{12\ldots n}(x_1, \ldots, x_n),$$

where f_0 is the mean value of $f(\mathbf{x})$, f_i's are the first order component functions, f_{ij}'s are the second order ones etc (Li et al., 2002). This decomposition is unique if the following condition is imposed on the component functions:

$$\int_0^1 f_{i_1,\ldots,i_s}(x_{i_1}, \ldots, x_{i_s})\, dx_{i_p} = 0.$$

Published online in Wiley InterScience | DOI: 10.1002/wilj
(www.interscience.wiley.com)

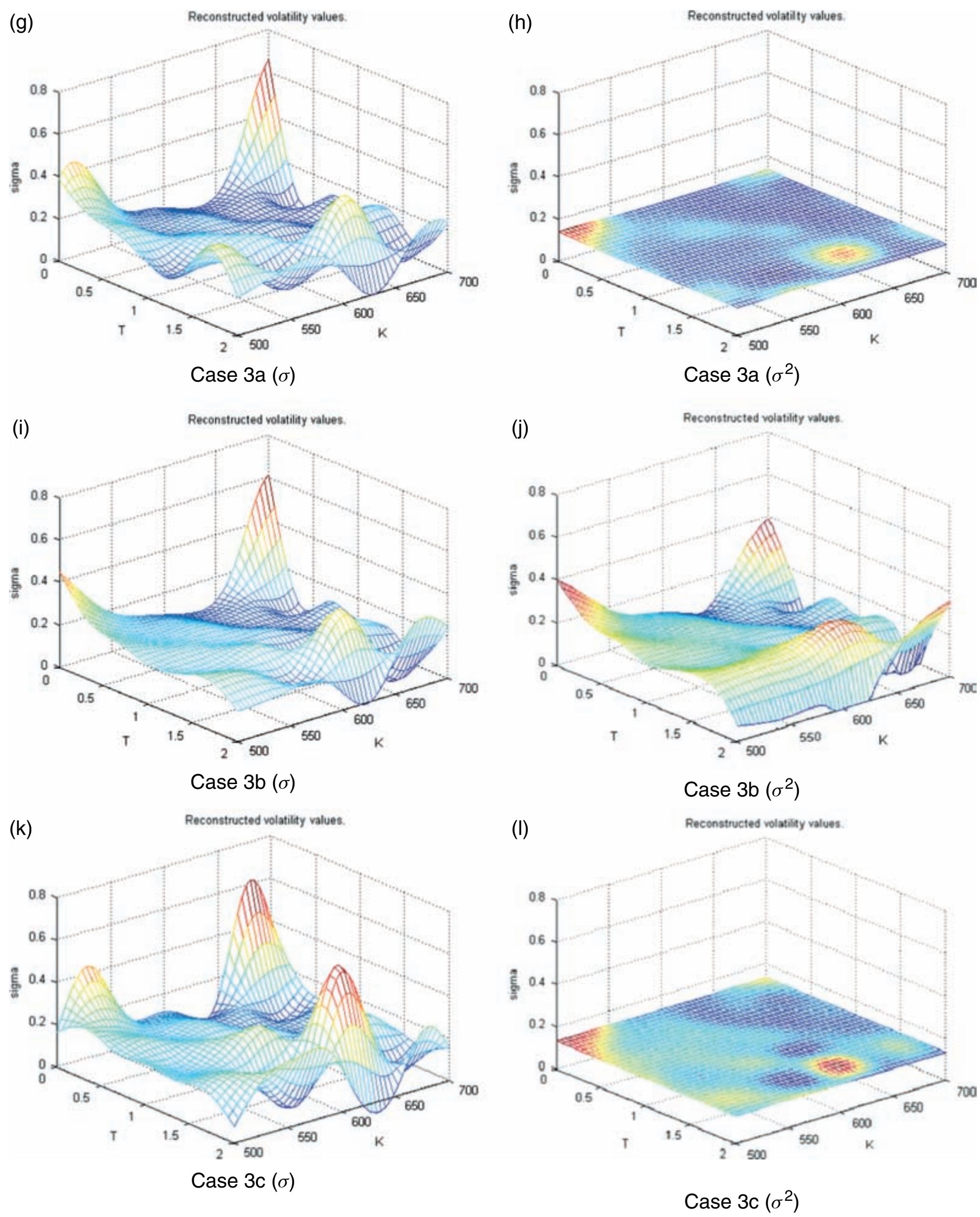


Figure 6: Continued

It is easy to show that in this case all component terms in ANOVA–HDMR decomposition are orthogonal.

Using a complete basis set of orthonormal polynomials, and assuming that the component functions are piecewise smooth and continuous, they can be expressed via the expansion (in this work we consider only the first and the second order component functions):

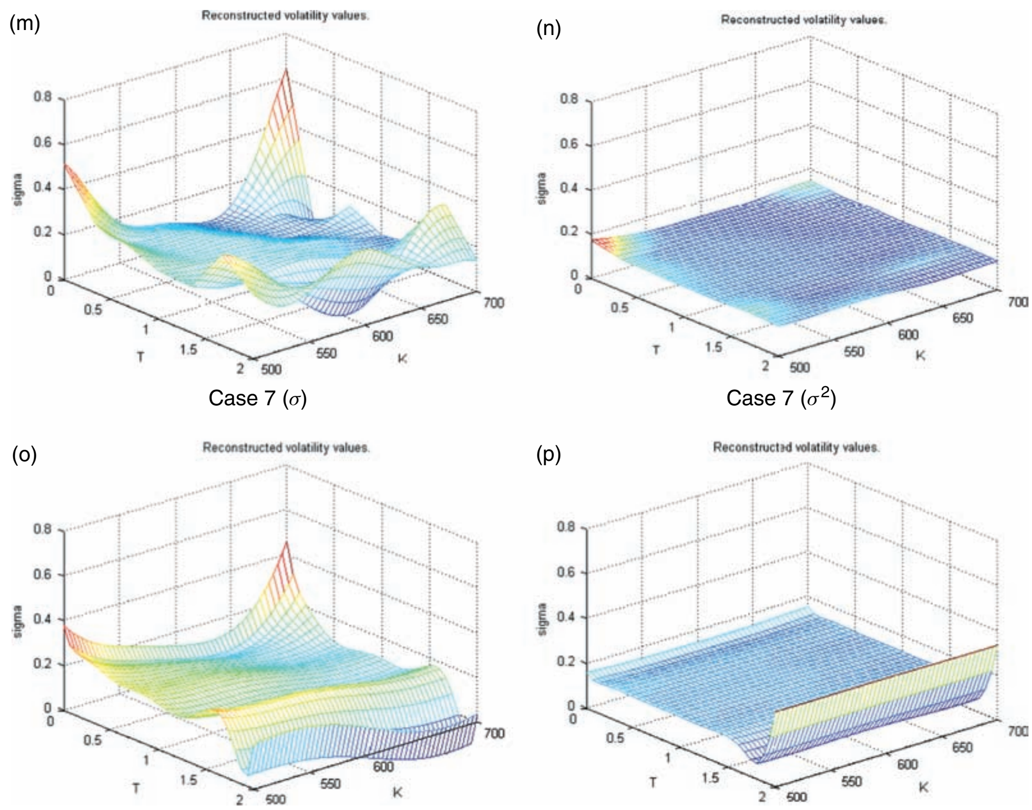
$$f_i(x_i) = \sum_{r=1}^{\infty} \alpha_r^i \phi_r(x_i), \tag{A.2.1}$$

$$f_{ij}(x_i, x_j) = \sum_{p=1}^{\infty} \sum_{q=1}^{\infty} \beta_{pq}^{ij} \phi_{pq}(x_i, x_j). \tag{A.2.2}$$

Here, $\phi_r(x_i)$, $\phi_{pq}(x_i, x_j)$ are sets of one and two-dimensional basis functions (orthonormal polynomials), α_r^i and β_{pq}^{ij} are coefficients of the expansion. In practice the summation in (A.2.1) and (A.2.2) is limited to some maximum polynomial orders k, l, l':

$$f_i(x_i) \approx \sum_{r=1}^{k} \alpha_r^i \phi_r(x_i),$$

$$f_{ij}(x_i, x_j) \approx \sum_{p=1}^{l} \sum_{q=1}^{l'} \beta_{pq}^{ij} \phi_{pq}(x_i, x_j).$$



Case 7 (σ) Case 7 (σ^2)
Case 8 (σ) Case 8 (σ^2)

Figure 6: Continued

Exploiting the orthogonality of the basis functions, the coefficients can be determined by integration of the product of $f(\mathbf{x})$ and the basis functions. In the framework of the volatility reconstruction problem the coefficients α_r^i and β_{pq}^{ij} are defined by solving the optimization problem as explained in Section 2.3.

Appendix B Implicit Finite Difference Scheme for Solving Dupire PDE

The Dupire partial differential equation has the following form:

$$\frac{\partial f}{\partial T} + (r-q)K\frac{\partial f}{\partial K} - \frac{1}{2}\sigma^2 K^2 \frac{\partial^2 f}{\partial K^2} + qf = 0, \qquad \text{(B.1)}$$

where f is the option price, K is the strike price, and T is the maturity. For a detailed discussion of the Dupire equation see e.g. Coleman et al. (1999), Ben Hamida and Cont (2005); Dupire (1994); Hanke and Rosler (2005).

PDE (B.1) is solved by using a finite-difference approximation on a regular mesh between K_{min} and K_{max} for the strike price and between 0 and T_{max} for the maturity. These meshes are divided into $M$and N parts in K and T directions, respectively:

$$M\Delta K = K_{\max} - K_{\min}$$
$$N\Delta T = T_{\max}.$$

The partial derivatives are approximated as follows:

$$\frac{\partial f}{\partial K} \approx \frac{f_{i,j+1} - f_{i,j-1}}{2\Delta K},$$
$$\frac{\partial f}{\partial T} \approx \frac{f_{i,j} - f_{i-1,j}}{\Delta T},$$
$$\frac{\partial^2 f}{\partial K^2} \approx \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{\Delta K^2}.$$

The finite-difference approximation of the Dupire PDE has the form:

$$a_j f_{i,j-1} + b_j f_{i,j} + c_j f_{i,j+1} = f_{i-1,j} \quad \forall i = 1\ldots N, \forall j = 1\ldots M,$$

where

$$a_{i,j} = -\frac{1}{2}(r-q)\left(\frac{K_{\min}}{\Delta K} + j\right)\Delta t - \frac{1}{2}\sigma^2\left(\frac{K_{\min}}{\Delta K} + j\right)^2 \Delta t,$$
$$b_{i,j} = 1 + \left[\sigma^2\left(\frac{K_{\min}}{\Delta K} + j\right)^2 + q\right]\Delta t,$$
$$c_{i,j} = \frac{1}{2}(r-q)\left(\frac{K_{\min}}{\Delta K} + j\right)\Delta t - \frac{1}{2}\sigma^2\left(\frac{K_{\min}}{\Delta K} + j\right)^2 \Delta t.$$

Published online in Wiley InterScience | DOI: 10.1002/wilj
(www.interscience.wiley.com)

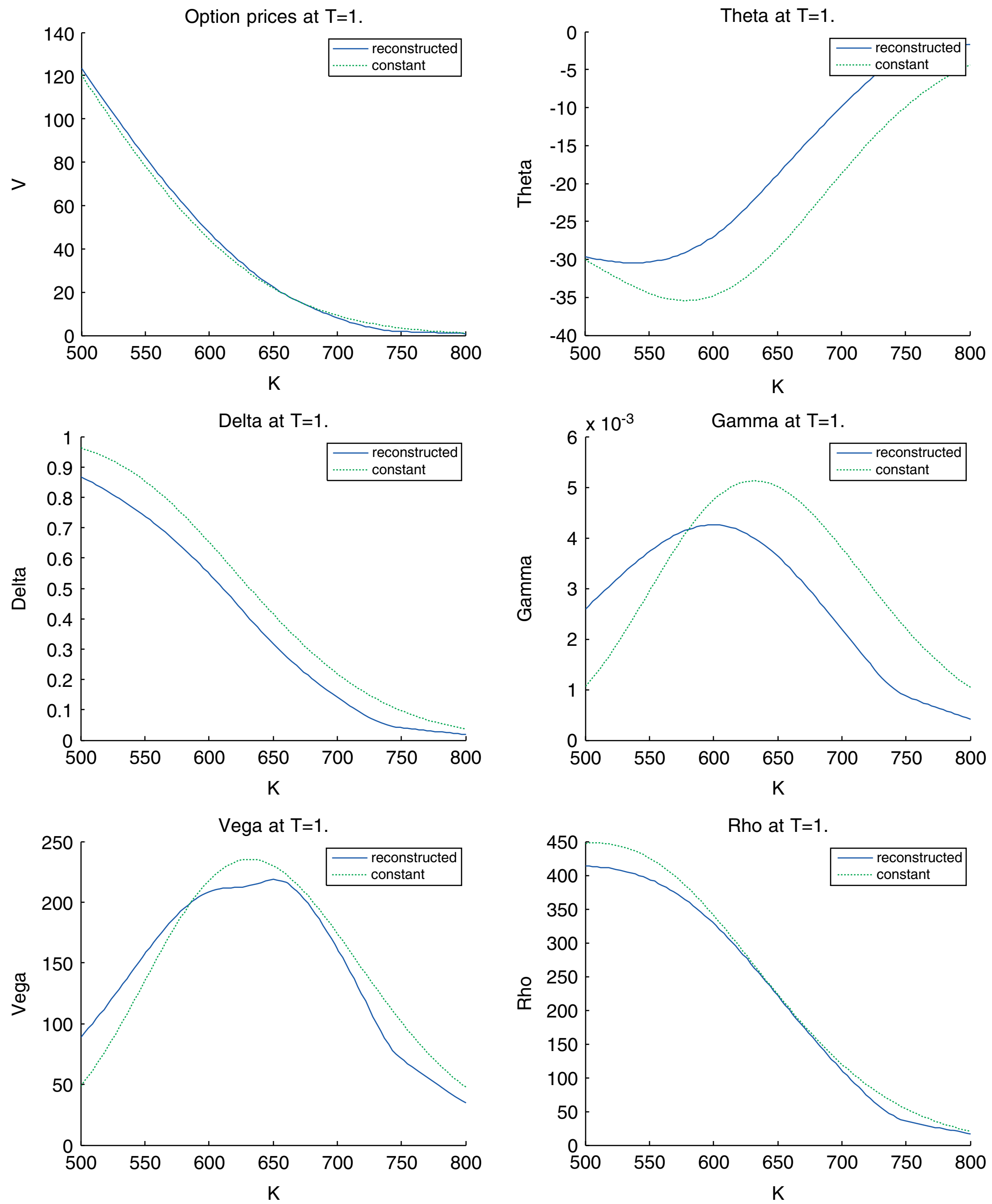


Figure 7: The Greeks with the reconstructed and constant volatility functions for the S&P data set with 1 year maturity (Case 3a, Table 1).

Here $\sigma = \sigma(K_j, T_i)$. It is a forward equation with the initial condition (for the European call):

$$f(K,0) = \max(S_{\text{init}} - K, 0)$$

and boundary conditions:

$$\begin{aligned} f(K,T) &= 0, & K &\gg S_{\text{init}}, \forall T(\text{at } K_{max}) \\ f(K,T) &= S_{\text{init}} - K, & K &\approx 0, \forall T(\text{at } K_{min}) \end{aligned}$$

Appendix C Input files for the *SobolVol* Program

The program *SobolVol* requires three input files. They are listed below.

1 InputData.txt file

This file contains the data for setting

1) parameters for solution of the Dupire equation;
2) parameters to compute the Greeks;

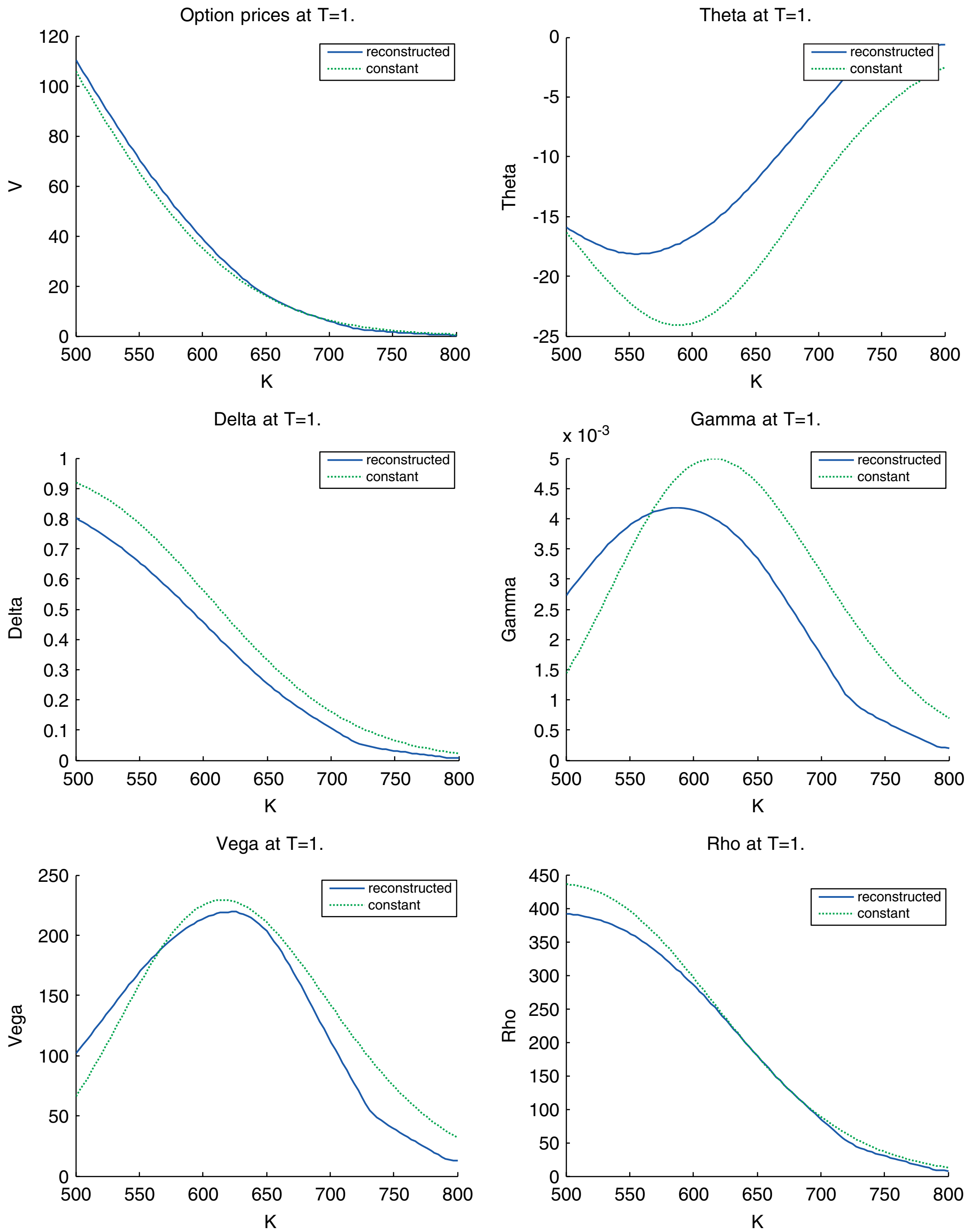


Figure 8: The Greeks with the reconstructed and constant volatility functions for the S&P data set with 1-year maturity (Case 7, Table 1).

3) parameters for plotting the volatility surface;
4) data point (pairs (K_i, T_i)) and implied volatilities if market data is used.

The Dupire PDE is solved using the implicit finite difference method that requires a mesh along K and T directions. Usual parameters such as the spot price, the interest and dividend rates are also required. These parameters are provided in the section entitled "Parameters for Dupire PDE".

In section "Parameters to compute the Greeks" user can specify the boundaries of the $K \times T$ domain in which Greeks to be calculated. Similarly, in section "Parameters for plotting the

Published online in Wiley InterScience | DOI: 10.1002/wilj (www.interscience.wiley.com)

volatility surface'' user can specify the $K \times T$ domain for plotting the volatility surface.

In the next section of the *InputData.txt* file the number of data points m ($(K_j, T_j), j = 1, \ldots, m$ pairs), strike prices K_j and maturities T_j are set. There are two options: a) to use a built-in volatility model ($\sigma = 15/S$); b) to use market data, in which case the user should specify the implied volatility values for all (K_j, T_j) pairs.

To get familiar with the program it is recommended to use a built-in volatility model as a learning exercise.

Important note: The data are separated from the comments with colons (:).

```
Parameters for solving the Dupire PDE:

      Minimal K on the mesh: 0.0 (*Sinit)
      Maximal K on the mesh: 2.0 (*Sinit)
      Number of steps in K direction: 100
      Number of steps in T direction: 60
      Initial stock price : 590
      Interest rate    : 0.06
      Dividend rate  : 0.0262

Parameters to compute the Greeks:

      Kmin: 501.5 //500
      Kmax: 678.5 //800
      Number of steps in K direction: 4 //21

      Tmin: 0.85 //1
      Tmax: 1.15
      Number of steps in T direction: 4 //2

Parameters for plotting the volatility surface:

      Kmin: 500
      Kmax: 700
      Number of steps by K: 51

Number of steps in T direction (from 0 to the
 maximal maturity): 31


Number of data points: 70

Strike prices:

501.50 531.0 560.50 590.0 619.50 649.0
 678.50 708.0 767.0 826.0
501.50 531.0 560.50 590.0 619.50 649.0
 678.50 708.0 767.0 826.0
501.50 531.0 560.50 590.0 619.50 649.0
 678.50 708.0 767.0 826.0
501.50 531.0 560.50 590.0 619.50 649.0
 678.50 708.0 767.0 826.0
501.50 531.0 560.50 590.0 619.50 649.0
 678.50 708.0 767.0 826.0
501.50 531.0 560.50 590.0 619.50 649.0
 678.50 708.0 767.0 826.0
501.50 531.0 560.50 590.0 619.50 649.0
 678.50 708.0 767.0 826.0

Maturities:

 0.1750 0.1750 0.1750 0.1750 0.1750 0.1750
   0.1750 0.1750 0.1750 0.1750
 0.4250 0.4250 0.4250 0.4250 0.4250 0.4250
   0.4250 0.4250 0.4250 0.4250
 0.6950 0.6950 0.6950 0.6950 0.6950 0.6950
   0.6950 0.6950 0.6950 0.6950
 0.9400 0.9400 0.9400 0.9400 0.9400 0.9400
   0.9400 0.9400 0.9400 0.9400
 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
   1.0000 1.0000 1.0000 1.0000
 1.5000 1.5000 1.5000 1.5000 1.5000 1.5000
   1.5000 1.5000 1.5000 1.5000
 2.0000 2.0000 2.0000 2.0000 2.0000 2.0000
   2.0000 2.0000 2.0000 2.0000

use a built-in volatility model (y/n): n
(if ``no'', then provide the implied volatility
values below)

Implied volatilities:

 0.1900 0.1680 0.1330 0.1130 0.1020 0.0970
   0.1200 0.1420 0.1690 0.2000
 0.1770 0.1550 0.1380 0.1250 0.1090 0.1030
   0.1000 0.1140 0.1300 0.1500
 0.1720 0.1570 0.1440 0.1330 0.1180 0.1040
   0.1000 0.1010 0.1080 0.1240
 0.1710 0.1590 0.1490 0.1370 0.1270 0.1130
   0.1060 0.1030 0.1000 0.1100
 0.1710 0.1590 0.1500 0.1380 0.1280 0.1150
   0.1070 0.1030 0.0990 0.1080
 0.1690 0.1600 0.1510 0.1420 0.1330 0.1240
   0.1190 0.1130 0.1070 0.1020
 0.1690 0.1610 0.1530 0.1450 0.1370 0.1300
   0.1260 0.1190 0.1150 0.1110
```

2 OptData.txt file

This file contains the data for a volatility approximation model and optimization. The type of the model is specified at the beginning of the file. There are four different model types to choose: 1) *spline* 2) *hdmr* 3) *1D spline* 4) *1D hdmr*. Model types are discussed in the text above.

The next part of the file specifies the number of coefficients (the number of function values for the spline model, the number of polynomial coefficients for the HDMR model etc.). The user needs to provide lower and upper bounds for variables (coefficients of the volatility approximation model) in the optimization problem.

Two optimization algorithms can be used: the first is the Levenberg-Marquardt (local) method and the second is the global optimization approach implemented in the *SobolOpt* program. *SobolOpt* is the part of the *SobolVol* program (the details can be found in Kucherenko and Sytsko, 2005). In the case of choosing *SobolOpt*, two other parameters are required. They should be specified in the end of *OptData.txt* file.

```
Model type (spline / hdmr / spline1d /hdmr1d):
         spline

Number of coefficients (max 1000):
         22

Lower bounds for the parameters:
         0 0 0 0 0 0 0 0 0 0
         0 0 0 0 0 0 0 0 0 0
         0 0

Upper bounds for the parameters:
         1 1 1 1 1 1 1 1 1 1
         1 1 1 1 1 1 1 1 1 1
         1 1

Optimization algorithm (lm / sobolopt /
         simulation): sobolopt

Initial conditions
(if all are = -1, then the mean
value of implied volatility will be used in
optimization):
        0.15 0.15 0.15 0.15 0.15 0.15 0.15
          0.15 0.15 0.15
        0.15 0.15 0.15 0.15 0.15 0.15 0.15
          0.15 0.15 0.15
        0.15 0.15

Size of the reduced sample set for SobolOpt:
         32

Size of the full sample set for SobolOpt:
         64
```

3 ModelData.txt file

In this file model parameters are specified. The model itself is chosen in the *OptData.txt* file.

There is an option to approximate σ or σ^2. Next part of the file deals with setting the spline knots (number of data, mesh points) or parameters of the HDMR model.

For the *spline* model, the user specifies at which (K_j, T_j) points the values of the volatility model function are used.

Here is an example of the *ModelData.txt* file for the *spline* model:

```
Sigma^2 is approximated (y/n): n

Parameters to the spline mesh
(if Kmesh and Tmesh number of points = -1, then
the implied
volatility values given in InputData.txt will
be used as initial values)

      Number of points in the Kmesh: 10
      Kmesh points (*Sinit):

            0.8000 0.8667 0.9333 1.0000 1.0667
            1.1333 1.2000 1.2667 1.3333 1.4000

      Number of points in the Tmesh: 7
      Tmesh points:

            0 0.3333 0.6667 1.0000 1.3333
              1.6667 2.0000

Spline end condition
(if 0, the derivative will be zero; if >=1e30,
the natural spline end condition is used):

      1e30
```

For the full HDMR model (*hdmr*), specified parameters include the number of polynomial coefficients for the first and second order component functions (the number of alpha coefficients for K and T, and the number of beta coefficients for (K_j,T_j) terms). Here is an example of the *ModelData.txt* file for the *hdmr* model:

```
Sigma^2 is approximated (y/n): y

Parameters for normalization in HDMR (if -1,
the default value will be used):

      Kmin: 0.3 (*Sinit) // default 0.3
      Kmax: 2.0 (*Sinit) // default 2.0
      Tmin: 0.0     // default 0.0

      Tmax: -1 // default Tmax

Polynomial degrees for alphas (K and T):

  4      4

Polynomial degrees for betas:

  beta[i: 1 j: 2 l: 3 l': 3]
```


Published online in Wiley InterScience | DOI: 10.1002/wilj
(www.interscience.wiley.com)

For the 1D HDMR model (*hdmr1d*) the parameters include the maximum number of polynomials (polynomial decompositions with respect to *K*) and the number of 1D models (each model is assigned to a fixed *T* value). Here is an example of the *ModelData.txt* file for the *hdmr1d* model:

```
Sigma^2 is approximated (y/n): n

The constant sigma0 is a parameter to be
optimized (y/n): n

Parameters for normalization in HDMR (if -1,
the default value will be used):

      Kmin: 0.6 (*Sinit) // default 0.3
      Kmax: 1.4 (*Sinit) // default 2.0
      Tmin: 0.0 // default 0.0
      Tmax: -1 // default Tmax

Number of polynomials: 10

Number of one-dimensional models: 7
```

Balazs Feil received his Ph.D. degree from University of Pannonia, Hungary, in 2006. He worked on the development and combination of computational intelligence and data mining methods for data based process modelling and optimization. His book, *Cluster Analysis for Data Mining and System Identification* (2007), describes the results of his work. Dr. Feil worked at Imperial College London as a research associate, and currently holds the position of Lecturer at University of Pannonia, Hungary.

Sergei Kucherenko obtained his M.Sc. and Ph.D. degrees from Moscow Engineering Physics Institute in Russia. He has held a number of research and faculty positions in various universities in Russia, the United States and the UK, and has also worked in an inverstment bank. Currently, he holds the position of Senior Research Associate at Imperial College, London. Dr. Kucherenko also provides consultancy services to investment banks and financial companies in the area of MC and Qausi MC simulation, and other advanced numerical techniques used in financial mathematics.

Nilay Shah has been on the academic staff in the CPSE and Department of Chemical Engineering at Imperial College since 1992, with a large research activity in the area of process and system optimization and sensitivity analysis. Prof. Shay received the Beilby Medal and Prize (awarded jointly by the Society of Chemical Industry, the Royal Society of Chemistry, and the Institute of Materials) for contributions to research and practice in process systems engineering in 2005 and was part of a team that won the RAE MacRobert Award and Prize in 2007.

REFERENCES

Andersen, L. and Brotherton-Ratcliffe, R. 1998. The Equity Option Volatility Smile: An Implicit Finite-Difference Approach. *The Journal of Computational Finance*, **1**(2): 5–32.

Ben Hamida, S. and Cont, R. 2005. Recovering Volatility from Option Prices by Evolutionary Optimization, *Journal of Computational Finance* **8**(4): 43–76.

Bouchouev, I. and Isakov, V. 1997. The Inverse Problem of Option Pricing. *Inverse Problems* **13**: 11–17.

Coleman, T. Li, Y. and Verma, A. 1999. Reconstructing the Unknown Local Volatility Function, *Journal of Computational Finance* **2(3)**: 77–102.

Coleman, T., Kim, Y., Li, Y. and Verma, A. 2003. *Dynamic Hedging in a Volatile Market*. Cornell University, Technical Report.

Crêpey, S. 2003. Calibration of the Local Volatility in a Trinomial Tree using Tikhonov Regularization. *Inverse Problems* **19**: 127.

Derman, E. and Kani, I. 1994. Riding on a Smile. *Risk* **7**: 32–39.

Dupire, B. 1994. Pricing with a Smile. *Risk* 7: 18–20.

Egger, H. and Engl, H. 2005. Tikhonov Regularization Applied to the Inverse Problem of Option Pricing: Convergence Analysis and Rates. *Inverse Problems* **21**: 1027–1045.

Feil, B., Kucherenko, S., and Shah, N. 2009. Comparison of Monte Carlo and Quasi Monte Carlo Sampling in High Dimensional Model Representation. Submitted to *Reliability Engineering and System Safety*.

Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T. 2005. *Numerical Recipes*. W.H. Press, Cambridge University Press.

Gatheral G. 2006. *The Volatility Surface: A Practitioner's Guide*. Wiley: Chichester.

Hanke, M. and Rosler, E. 2005. Computation of Local Volatilities from Regularized Dupire Equations. *International Journal of Theoretical and Applied Finance* **8**: 207–221.

Hayes, J. 1974. Numerical Methods for Curve and Surface Fitting. *Bull. Inst. Math. Appl.* **10**: 144–152.

Hayes, J.G. and Halliday, J. 1974. The Least-Squares Fitting of Cubic Spline Surfaces to General Data Sets. *J. Inst. Math. Appl.* **14**: 89–103.

Jackson, N., Süli, E. and Howison, S. 1998. Computation of Deterministic Volatility Surfaces. *Journal of Computational Finance* **2(2)**: 5–32.

Jackwerth, J. and Rubinstein, M. 1996. Recovering Probability Distributions from Option Prices. *The Journal of Finance* **51**: 1611–1631.

Kim, B.-H., Lee, D. and Lee, J. 2006. Local Volatility Function Approximation Using Reconstructed Radial Basis Function Networks. *Lecture Notes in Computer Science–Advances in Neural Networks* **3973**: 524–530.

Kucherenko, S. and Sytsko, Y. 2005. Application of Deterministic Low-Discrepancy Sequences in Global Optimization. *Computational Optimization and Applications* **30(3)**: 297–318.

Li, G., Wang, S.-W., Rabitz, H., Wang, S. and Jaffe, P. 2002. Global Uncertainty Assessments by High Dimensional Model Representations (HDMR). *Chemical Engineering Science* **57**: 4445–4460.

NAG C Library. Curve and Surface Fitting, ftp.nag.co.uk/numeric.

Wilmott P. 2006. *Paul Wilmott on Quantitative Finance*. Wiley: Chichester.

Published online in Wiley InterScience | DOI: 10.1002/wilj
(www.interscience.wiley.com)