A Bayesian Approach for Constructing Implied Volatility Surfaces through Neural Networks

Article *in* The Journal of Computational Finance · January 2000

DOI: 10.21314/JCF.2000.053

3 authors, including:

Marco Avellaneda
New York University
134 PUBLICATIONS **7,354** CITATIONS
SEE PROFILE

Fabio Stella
Università degli Studi di Milano-Bicocca
110 PUBLICATIONS **1,253** CITATIONS
SEE PROFILE

A Bayesian approach for constructing implied volatility surfaces through neural networks

M. Avellaneda

Courant Institute of Mathematical Sciences, 251 Mercer Street, New York, New York 10043, USA

A. Carelli

Risk Management and Research, Banca Intesa, Via Clerici 4, 20121 Milan, Italy

F. Stella

Università degli Studi di Milano – Bicocca, DISCo, Via Bicocca degli Arcimboldi 8, 20126 Milan, Italy

In this paper the authors present a new option pricing scheme which deals with a nonconstant volatility for the price of the underlying asset. The main feature of the proposed pricing scheme consists of exploiting recent developments about Bayesian learning within the artificial neural networks framework. Indeed, the Bayesian learning approach allows the data to speak for itself, i.e., to make a few general assumptions about the process to be modeled and to exploit all the available data concerning the price of traded options for modeling the implied volatility surface. The nonparametric model of the implied volatility surface, obtained through an infinite feedforward neural network and by exploiting the Bayesian formulation of the learning problem, is used within the proposed option pricing scheme. This pricing scheme relies upon the Dupire formula, which maps the implied volatility surface to the corresponding local volatility function. Numerical experiments for the case of the USD/DM over-the-counter options are presented together with a graphical analysis of the resulting smiles which attest to the effectiveness of the overall approach to option pricing.

1. INTRODUCTION

The Black–Scholes option pricing formula (see Black and Scholes 1973) assumes that the risk-free interest together with the volatility of the underlying asset remain fixed at given and constant levels over the life cycle of the option. This simplifying assumption, which can be considered true for short maturity options, has been under severe judgement from practitioners in the last decade, especially in cases where the maturity increases. Indeed, the observed *volatility smile effect* (i.e., options written on the same underlying asset usually trade with different implied volatilities) witnesses that such an assumption is too restrictive and less plausibly verified as the maturity increases. Furthermore, the presence of a time effect is suspected. Indeed, in the Foreign Exchange market, options with longer maturities typically trade at higher implied volatilities than options

with shorter maturities. This evidence is not consistent with the constant volatility assumption of the Black–Scholes option pricing formula and may be motivated by the presence of *fat tails* for the risk-neutral distribution (i.e., extreme values for the prices are more likely than expected according to the lognormal probability distribution model). A lot of effort has been devoted to addressing the Black–Scholes option pricing simplification either by allowing the interest rate and volatility to be time dependent or by introducing an interest rate and volatility that are stochastic. However, these approaches suffer from both theoretical and practical problems. In particular, the stochastic volatility approach, which has been pursued by many researchers (see Heston 1993; Dupire 1994; Avellaneda *et al*. 1997; Derman and Kani 1998; Fouque, Papanicolaou, and Sircar 2000), uses a stochastic volatility model which requires the practitioner to cope with complex and difficult to verify assumptions about the underlying price process.

In this paper the authors propose a *data-driven option pricing scheme* which overcomes several of the above-mentioned limitations by leaving the data to speak about the underlying price process. Indeed, the proposed pricing scheme allows us to price complex option contracts, as well as options traded on illiquid markets, by exploiting only the information coming from the market data. The main idea underlying the pricing scheme is to *let the data speak for itself*, that is, to make a few general assumptions about the process to be modeled and to exploit all the available data coming from the prices of traded options to extract information about the underlying price process (i.e., the local volatility function) through the implied volatility data associated with the option contract. This approach relies upon the Dupire (1994) formula which maps the implied volatility surface to the corresponding local volatility function. The implementation of this mapping requires the availability of the implied volatility for any pair consisting of a strike and a time to maturity in a given range and therefore cannot be directly applied to the usually available implied volatility data. This limitation can be overcome by approximating the implied volatility surface. Unfortunately, this is not an easy task for the following two main reasons:

1. The implied volatility surface is thought to be very complex;
2. Option price data suffers from paucity (i.e., few data points can be collected for different strikes and maturities even in more liquid markets).

While the first reason suggests that a *nonparametric approach* should be appropriate to cope with the implied volatility approximation task, the second one warns about the possible *overfitting* of the few data points at hand. In order to deal with these problems, the authors investigate the use of feedforward neural networks (FNNs) (White 1989; Bishop 1995) together with the Bayesian learning scheme (McKay 1992). Indeed, while FNNs allow us not to *a priori* restrict the complexity of the function to be approximated, the Bayesian learning scheme carefully evaluates the extent to which the available flexibility/complexity, offered by the nonparametric model (FNN), is used. Numerical approaches to option pricing using artificial neural networks as well as nonparametric models have

been proposed and investigated by other researchers (see, e.g., Hutchinson, Lo, and Poggio 1994; Yacine 1996).

Once an estimate for the local volatility function is available, the underlying price process is fully described and therefore the option pricing process can be carried out. The pricing process can be accomplished using one of a number of approaches, namely, binomial trees, trinomial trees, or Monte Carlo scenario generation. Of these, the authors recommend the use of the trinomial tree approach (see Avellaneda and Paras 1996), which has the maximum flexibility in that it allows the modeling of an underlying price process with both stochastic volatility and stochastic drift.

The rest of the paper is organized as follows. The proposed option pricing scheme is presented and the main computational steps discussed in Section 2. Section 3 is devoted to the numerical evaluation of the option pricing scheme performed using data from the over-the-counter USD/DM currency options market. Finally, in Section 4, conclusions and directions for further research are presented.

2. THE OPTION PRICING SCHEME

The proposed pricing scheme, which is shown in Figure 1, consists of the following five steps:

1. implied volatility modeling (IVM);
2. option price computation (OPC);
3. local volatility computation (LVC);
4. model discretization (MD);
5. general contingent claims pricing (GCCP).

These are described in detail in the following sections.

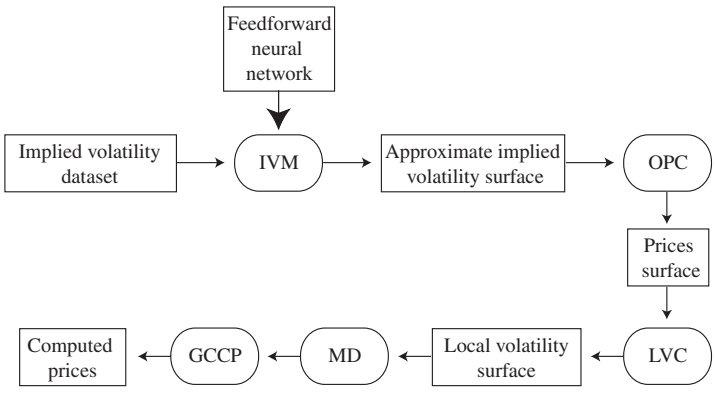

FIGURE 1. Pricing scheme: the boxes represent input or computed quantities, while rounded boxes are the computational steps of the pricing scheme.

2.1 *Implied Volatility Modeling (IVM Step)*

This section describes how FNNs, together with Bayesian learning, can be exploited to solve the problem of implied volatility surface approximation. An FNN (see Figure 2) is a layered structure consisting of computing units, named *artificial neurons* (circles), and connections between neurons, named *synapses* (directed links) (Bishop 1995). Artificial neurons of the input layer are associated with the independent variables or input variables, while the output neurons are associated with the response variables or output variables. Synapses are oriented connections linking the neurons from the input layer to the neurons of the hidden layer and the neurons from the hidden layer to the output neurons. The network's structure implies that the information flows from neurons in the lower layers to those in the higher layers and cannot flow between the neurons in the same layer or from neurons in a higher layer to those in a lower layer.

Without loss of generality, here and in the rest of the paper, artificial feedforward neural networks with only one response variable are considered. The FNN's graphical structure dictates that, for any given input, the corresponding output value is obtained through propagation from the input layer to the hidden layer and then from the hidden layer to the output neuron. The strength of the synapse from neuron i to neuron j is determined by means of a real value named *weight*. Furthermore, each neuron j from the hidden layer (and eventually the output neuron) are associated with a real value named the *neuron's bias* or *threshold* and with a nonlinear function named the *transfer* or *activation function*.

In order to formally describe the structure of a FNN and how, given an input vector, the network's output is computed, consider the FNN shown in Figure 2 consisting of I input variables, m neurons in the hidden layer, and a single output neuron.

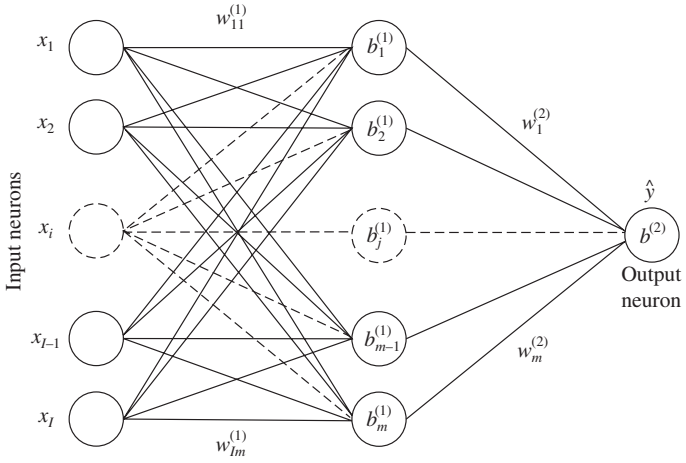

FIGURE 2. Single layer feedforward neural network: a single layer FNN consisting of I input neurons, m hidden neurons, and one output neuron together with its adjustable parameters, i.e., weights and biases.

We let:

- $\boldsymbol{x} = (x_1, \ldots, x_I)$ be the input vector;
- $w_{i,j}^{(1)}$ be the weight for the synapse from the ith input to the jth hidden neuron;
- $b_j^{(1)}$ be the bias associated with the jth neuron of the hidden layer;
- $w_j^{(2)}$ be the weight from the jth hidden neuron to the output neuron;
- $b^{(2)}$ be the bias associated with the output neuron.

Furthermore, let $\mathcal{G}^{(1)}(\cdot)$ and $\mathcal{G}^{(2)}(\cdot)$ be the activation functions associated with hidden and output neurons, respectively. Typical choices for the activation function include the following:

$$\text{logistic sigmoid} \qquad \mathcal{G}(z) = \frac{1}{1 + \mathrm{e}^{-z}} \tag{1}$$

and

$$\text{hyperbolic tangent} \qquad \mathcal{G}(z) = \frac{\mathrm{e}^{z} - \mathrm{e}^{-z}}{\mathrm{e}^{z} + \mathrm{e}^{-z}}. \tag{2}$$

Finally, we let

$$\boldsymbol{\theta} = (w_{1,1}^{(1)}, \ldots, w_{1,m}^{(1)}, \ldots, w_{I,1}^{(1)}, \ldots, w_{I,m}^{(1)}, b_1^{(1)}, \ldots, b_m^{(1)}, w_1^{(2)}, \ldots, w_m^{(2)}, b^{(2)}) \tag{3}$$

and assume that the activation functions $\mathcal{G}^{(1)}(\cdot)$ and $\mathcal{G}^{(2)}(\cdot)$ are given. Then the network in Figure 2 computes the following function of the input vector $\boldsymbol{x}$:

$$\hat{y}(\boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{G}^{(2)}\left(\sum_{j=1}^{m}\left[w_j^{(2)}\mathcal{G}^{(1)}\left(\sum_{i=1}^{I} w_{i,j}^{(1)} x_i - b_j^{(1)}\right)\right] - b^{(2)}\right). \tag{4}$$

FNNs are data-driven models (Hertz, Krogh, and Palmer 1991; Kung 1993) in the sense that they make no assumptions about the nature of the function to be approximated but instead use only the available data to build such an approximation. This computational model has proved to be useful and effective for solving several classes of problems (Fu 1982; Gader *et al*. 1991; Chen, Chen, and Lin 1996).

The main property of FNNs is that they are universal approximators in the sense described by Cybenko (1989), Hornik, Stinchombe, and White (1989), and White (1989, 1990). This property establishes that FNNs, with as many as one hidden layer and sigmoid activation functions, are capable of approximating to any degree of accuracy any mapping between independent variables and response variables as the size of the data set describing the *data generating process* (DGP) and the number of neurons of the hidden layer go to infinity.

Even though today the limitation in the number of available neurons is less keenly felt due to technological enhancements and cost reduction, the problem of paucity of data is still the main concern. Indeed, in actual applications, data are difficult and expensive to get and for this reason it is important that the FNN model be *data-efficient*, that is, squeeze maximum information out of few data.

Unfortunately, the universal approximation property enjoyed by FNNs is an asymptotic result of no help in actual problems in which, quite to the contrary, an increase in the number of parameters too often leads to overfitting and overparametrization. The process through which the approximation is obtained, starting from the available data set, is often called *learning* and requires several complex and interconnected tasks such as:

1. *network structure selection*, i.e., the selection of the *optimal* number of hidden layers together with the selection of the *optimal* number of neurons associated with each hidden layer;
2. *network training*, i.e., the selection of the *optimal* values for the FNN's adjustable parameters;
3. *activation function selection*, i.e., the selection of the functional form for the activation function associated with each neuron.

Cross-validation, i.e., the hold-out method, minimum description length (Rissanen 1996), Vapnik–Chervonenkis dimension (Vapnik 1995, 1998), and Bayesian learning are the main approaches for learning in FNNs. Of these the hold-out method and Bayesian learning are probably the most often utilized in practice.

According to the hold-out method (Stone 1974; Wahba and Wold 1975; Bishop 1995), various networks are trained by minimization of an appropriate error function defined with respect to a *training set*. The performance of the networks is then compared by evaluating the error function using an independent *validation set*, and the network having the smallest error, with respect to the validation set, is selected. Since this procedure can itself lead to overfitting of the validation set, the performance of the selected network should be confirmed by measuring its performance on a third independent set of data called the *test set*.

In contrast, the Bayesian framework (McKay 1992; Bishop 1995) allows us to deal with the above-mentioned tasks contemporaneously without making any strong prior assumption about the complexity of the function to be approximated. The Bayesian formulation of the learning problem in FNNs considers a prior distribution $p(\boldsymbol{\theta})$, over the network's vector $\boldsymbol{\theta}$ of adjustable parameters, defined through the formula (3), which expresses some general properties, such as smoothness, for the function to be approximated, and the likelihood $p(\chi \mid \boldsymbol{\theta})$ of the data set χ given the parameters vector $\boldsymbol{\theta}$, which models the noise process associated with the response variable. Once the data set χ has been observed the posterior probability distribution $p(\boldsymbol{\theta} \mid \chi)$ of the parameters vector $\boldsymbol{\theta}$, given the available data set χ, can be computed, according to the Bayes theorem, as follows:

$$p(\boldsymbol{\theta} \mid \chi) = \frac{p(\chi \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\chi)}. \tag{5}$$

According to the Bayesian framework, a trained network is described in terms of its posterior probability distribution $p(\boldsymbol{\theta} \mid \chi)$. Indeed, the posterior probability $p(\boldsymbol{\theta} \mid \chi)$ of the network's parameters vector $\boldsymbol{\theta}$, given the available data set χ,

implements the learning process and fully describes the network's output over the input space. The main reasons motivating the Bayesian formulation of the learning problem are the following:

1. All the available data points can be exploited for solving the learning problem.
2. The complexity of the function to be approximated should not be strongly constrained in advance but can be automatically selected according to the evidence coming from the available data set.
3. It provides a unifying framework for data modeling and allows us to develop probabilistic models that are well matched to the data and therefore allow us to make *optimal predictions*.
4. Bayesian methods are *mechanistic*: once the model space has been defined, the rules of probability theory give a unique answer which takes into account all the given information.
5. Bayesian inference satisfies the likelihood principle in that inferences depend only on the probabilities assigned to the data and not on properties of other data sets which could have occurred but did not.
6. Probabilistic modeling handles uncertainty in a natural manner. There is a unique prescription for incorporating uncertainty about parameters into predictions of other variables, namely, marginalization.
7. Bayesian model comparison embodies *Occam's razor*, the principle that states a preference for simple models over complex ones.

In particular, points 1 and 2 make it very interesting to evaluate how the Bayesian learning framework in FNNs can be exploited in order to solve the nonparametric estimation problem of the implied volatility surface. Indeed, in such a case, the main problems of the approximation task are the complexity of the function to be approximated, i.e., the implied volatility surface, and the data paucity, i.e., the scarce availability of data points even for more liquid option markets.

Before introducing and describing the FNN model used to approximate the implied volatility surface, a discussion of the preprocessing step concerning the option data is needed. Indeed, it is well known that the available implied volatility data depends on the strike price and time to maturity as well as on the cost-of-carry, i.e., the interest rate minus the dividend rate. The dependence on the cost-of-carry can be misleading and, therefore, transforming the data, i.e., the strike price, in such a way that the drift of the interest rate and the cost-of-carry are both zero is recommended. To clarify this point, let K be the strike price and T be the time to maturity (expressed in years). Then the zero-drift transformation is of the form

$$\tilde{K} = f(K, T, \rho), \tag{6}$$

where ρ and the particular form for the transformation function f depend upon the particular option contract and therefore will be described in Section 3, where

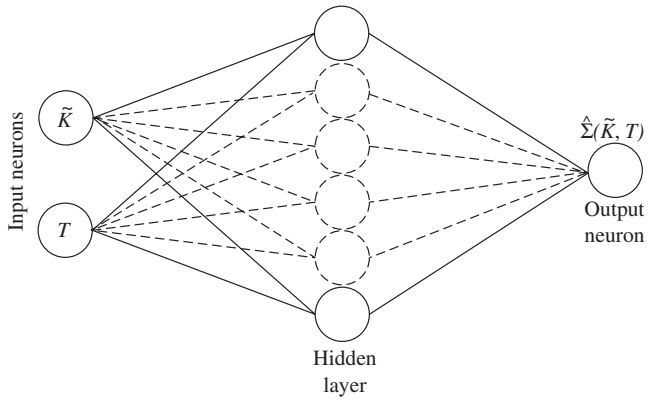

FIGURE 3. Single layer FNN: the FNN model used for the approximation of the implied volatility surface $\Sigma(\tilde{K}, T)$.

the case of USD/DM options is analyzed. According to the transformation (6), the implied volatility, computed on mid-market prices by means of the Black–Scholes formula, is denoted by $\Sigma(\tilde{K}, T)$.

The FNN model used to approximate the implied volatility surface $\Sigma(\tilde{K}, T)$ (Figure 3) consists of two input neurons ($I = 2$) associated, respectively, with the adjusted strike price $\tilde{K}$ and time to maturity T, 20 hidden neurons ($m = 20$) with logistic activation function (1), and one output neuron, associated with the approximated implied volatility surface $\hat{\Sigma}(\tilde{K}, T)$, with logistic activation function (1). Let

$$\boldsymbol{\theta} = (w_{1,1}^{(1)}, \ldots, w_{1,20}^{(1)}, w_{2,1}^{(1)}, \ldots, w_{2,20}^{(1)},\ b_1^{(1)}, \ldots, b_{20}^{(1)},\ w_1^{(2)}, \ldots, w_{20}^{(2)},\ b^{(2)}) \tag{7}$$

be the network's adjustable parameters vector. Then the FNN in Figure 3 computes the following function of the input variables $\tilde{K}$ and T:

$$\hat{\Sigma}(\tilde{K}, T) = \left[1 + \exp\left(-\sum_{j=1}^{20}\left\{w_j^{(2)}\left[\frac{1}{1 + \exp(-w_{1,j}^{(1)}\tilde{K} - w_{2,j}^{(1)}T + b_j^{(1)})}\right] - b^{(2)}\right\}\right)\right]^{-1}. \tag{8}$$

The learning problem has been formulated according to the approach named *infinite networks* introduced by Neal (1996). With this approach the prior distribution of the FNN's adjustable parameters vector $\boldsymbol{\theta}$ is assumed to be of the form

$$p(\boldsymbol{\theta}) = \frac{1}{Z_{\boldsymbol{\theta}}}\exp\left(-\tfrac{1}{2}\alpha\|\boldsymbol{\theta}\|_2^2\right), \tag{9}$$

where $Z_{\boldsymbol{\theta}} = \int \exp(-\frac{1}{2}\alpha\|\boldsymbol{\theta}\|_2^2)\,\mathrm{d}\boldsymbol{\theta}$ is a normalization factor which ensures that $\int p(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta} = 1$, and α, called a hyperparameter, allows us to automatically control the complexity of the FNN. The $1/\alpha$ term represents the variance of the Gaussian distribution associated with the network's adjustable parameters and controls the penalty associated with the network's complexity/flexibility.

The likelihood has the following form:

$$p(\chi \mid \boldsymbol{\theta}) = \frac{1}{Z_\chi} \exp\left(-\tfrac{1}{2}\beta \sum_{n=1}^{D} [\hat{\Sigma}(\tilde{K}_n, T_n, \boldsymbol{\theta}) - \Sigma(\tilde{K}_n, T_n)]^2 \right), \tag{10}$$

where $\hat{\Sigma}(\tilde{K}_n, T_n, \boldsymbol{\theta})$ represents the output for a network with parameters vector $\boldsymbol{\theta}$, $Z_\chi = \int \exp\{-\frac{1}{2}\beta \sum_{n=1}^{D} [\hat{\Sigma}(\tilde{K}_n, T_n, \boldsymbol{\theta}) - \Sigma(\tilde{K}_n, T_n)]^2\} \, \mathrm{d}\boldsymbol{\theta}$ is a normalization factor, β is the hyperparameter controlling the extent to which the FNN's degrees of freedom are used to fit the data points, and D represents the number of data points used to accomplish the learning process. In particular, $1/\beta$ represents the noise level associated with the unknown function to be approximated.

Once the prior distribution and the likelihood have been specified, it is possible to obtain the posterior distribution $p(\boldsymbol{\theta} \mid \chi)$ by substituting (9) and (10) in (5):

$$p(\boldsymbol{\theta} \mid \chi) = \frac{1}{Z_S} \exp\left(-\tfrac{1}{2}\beta \sum_{n=1}^{D} [\hat{\Sigma}(\tilde{K}_n, T_n, \boldsymbol{\theta}) - \Sigma(\tilde{K}_n, T_n)]^2 - \tfrac{1}{2}\alpha \|\boldsymbol{\theta}\|_2^2 \right), \tag{11}$$

where

$$Z_S = \int \exp\left(-\tfrac{1}{2}\beta \sum_{n=1}^{D} [\hat{\Sigma}(\tilde{K}_n, T_n, \boldsymbol{\theta}) - \Sigma(\tilde{K}_n, T_n)]^2 - \tfrac{1}{2}\alpha \|\boldsymbol{\theta}\|_2^2 \right) \mathrm{d}\boldsymbol{\theta}.$$

The posterior distribution (11) fully describes the FNN model which approximates the implied volatility surface and therefore is the result of the learning process.

The task of computing the posterior distribution (11) is extremely complex and can be solved through both approximation methods, as proposed by McKay (1992), or by means of direct sampling through the Markov chain Monte Carlo (MCMC) approach, as proposed by Neal (1996).

2.2 *Option Prices Computation (OPC Step)*

This computational step allows one to go from the approximated implied volatility $\hat{\Sigma}(\tilde{K}, T)$, for each adjusted strike price $\tilde{K}$ and time to maturity T, to the corresponding call option price C.

For computational purposes, we need to evaluate the approximated implied volatility $\hat{\Sigma}(\tilde{K}, T)$ for the input pairs $(\tilde{k}_l, \tau_r)$ constrained so as to belong to a discrete $d \times d$ square grid $\tilde{\mathcal{K}} \otimes \mathcal{T}$ defined as follows:

$$\tilde{\mathcal{K}} = \{\tilde{k} \in \mathbb{R}^+ : \tilde{k}_{\min} \leqslant \tilde{k} \leqslant \tilde{k}_{\max}\}, \qquad \mathcal{T} = \{\tau \in \mathbb{R}^+ : 0 \leqslant \tau \leqslant T\}.$$

Notice that the grid $\tilde{\mathcal{K}} \otimes \mathcal{T}$ consists of all the ordered pairs $(\tilde{k}_l, \tau_r)$ where $l, r = 1, \ldots, d$. According to this discretization, the computation of the call option prices is performed on a $d \times d$ grid via the Black–Scholes formula, i.e., for each pair $(\tilde{k}_l, \tau_r) \in \tilde{\mathcal{K}} \otimes \mathcal{T}$, the corresponding call option price is computed as follows:

$$c_{lr} = S\Phi(d_1) - \tilde{k}_l \Phi(d_2) \quad \forall\, l, r = 1, \ldots, d, \tag{12}$$

with

$$d_1 = \frac{\ln(S/\tilde{k}_l) + \frac{1}{2}\tau_r \hat{\Sigma}(\tilde{k}_l, \tau_r)^2}{\hat{\Sigma}(\tilde{k}_l, \tau_r)\sqrt{\tau_r}}, \qquad d_2 = \frac{\ln(S/\tilde{k}_l) - \frac{1}{2}\tau_r \hat{\Sigma}(\tilde{k}_l, \tau_r)^2}{\hat{\Sigma}(\tilde{k}_l, \tau_r)\sqrt{\tau_r}}, \tag{13}$$

where S represents the market spot price associated with the underlying security and $\Phi(\cdot)$ represents the standard normal cumulative distribution function.

2.3 *Local Volatility Computation (LVC Step)*

This section describes how the Dupire formula can be exploited for computing the local volatility function by means of the approximated implied volatility surface. Indeed, by assuming that the underlying security S follows a risk-neutral process of the form

$$\frac{\mathrm{d}S}{S} = \mu(t)\,\mathrm{d}t + \sigma(S, t)\,\mathrm{d}W, \tag{14}$$

where $\mu(t)$ represents the cost-of-carry and $\sigma(S, t)$ is the local volatility function, Dupire (1994) has shown that the conditional probability distribution of the price and the local volatility function can be computed as explicit functions of the derivatives of the price function with respect to the strike price and the time to maturity. Notice that, according to the Dupire approach, the volatility $\sigma(S, t)$ is assumed to be a deterministic function of both the time t and the underlying price S. The Dupire formula can be viewed as a useful tool for computing the local volatility function starting from the option prices. In order to clarify this point, let $C(S, \tilde{K}, T)$ be the call option price associated with the underlying price S, the adjusted strike price $\tilde{K}$, and the time to maturity T, and assume that the cost-of-carry is zero. Then the Dupire formula states that

$$\sigma(S, t) = \sqrt{\frac{\frac{\partial}{\partial T} C(S, \tilde{K}, T)}{\frac{1}{2}\tilde{K}^2 \frac{\partial^2}{\partial \tilde{K}^2} C(S, \tilde{K}, T)}} \tag{15}$$

and therefore allows us to compute the local volatility σ_{lr} associated with any given pair $(\tilde{k}_l, \tau_r) \in \tilde{\mathcal{K}} \otimes \mathcal{T}$, as follows:

$$\sigma_{lr} \equiv \sigma(s_l, t_r) = \sqrt{\frac{g_{lr}}{\frac{1}{2}\tilde{k}_l^2 f_{lr}}} \quad \forall\, l, r = 1, \ldots, d, \tag{16}$$

where g_{lr} represents the first derivative of the call option price with respect to the time, while f_{lr} represents the second derivative of the call option price with respect to the adjusted strike price, and where both derivatives are evaluated for the pair $(\tilde{k}_l, \tau_r)$.

Notice that even if the new pairs (s_l, t_r) range over the same set $\tilde{\mathcal{K}} \otimes \mathcal{T}$ as the pairs $(\tilde{k}_l, \tau_r)$, they assume a completely different meaning. Indeed, they represent, respectively, the level of underlying S (not the adjusted strike price $\tilde{K}$) and the

time t (not the time to maturity T). The importance of this transformation lies in the fact that it allows a shift from a measure of global volatility (implied volatility) to a measure of local volatility and, therefore, to distinguish each node of the grid according to its proper local volatility value describing the underlying security process (14).

2.4 *Model Discretization (MD Step)*

The model discretization step allows one to compute the prices of general contingent claims whose underlying process is described by (14). In principle, this computation could be performed using several approaches like binomial trees, trinomial trees, and Monte Carlo scenario generation. Of these, the authors recommend the use of the trinomial tree approach (Avellaneda and Paras 1996).

The trinomial tree (Figure 4) achieves the maximum flexibility, i.e., it allows one to model an underlying price process (14) with both stochastic volatility and stochastic drift. Stability, calibration, and convergence conditions (Avellaneda, Levy, and Paras 1995) allow for the definition of the trinomial tree parameters in such a way as to guarantee that the discrete approximation of the price process effectively assigns to each distinct level of S and t the corresponding volatility value. The parameters used for such a computation are expressed through the following formulas (Rebonato 1999; Avellaneda and Laurence 2000):

$$p_{\mathrm{u}} = \tfrac{1}{2}p\big(1 - \tfrac{1}{2}\sigma_{\max}\sqrt{\Delta t}\big), \qquad p_{\mathrm{m}} = 1 - p, \qquad p_{\mathrm{d}} = \tfrac{1}{2}p\big(1 + \tfrac{1}{2}\sigma_{\max}\sqrt{\Delta t}\big), \tag{17}$$

$$s_{\mathrm{u}} = \exp\big(\sigma_{\max}\sqrt{\Delta t}\big), \qquad s_{\mathrm{m}} = 1, \qquad s_{\mathrm{d}} = \exp\big(-\sigma_{\max}\sqrt{\Delta t}\big), \tag{18}$$

where Δt represents a (small) time interval between successive shocks, measured in years, while $\sigma_{\max} \equiv \max_{lr}\{\sigma_{lr}\}$ is the maximum value of the local volatility to be modeled.

In order to better clarify the contents of the model discretization, the basic steps for determining the option price from the local volatility are described in detail. To this end, the contract is characterized in terms of its time to maturity T, measured in years, its underlying price S, the number of periods N, and the adjusted strike price $\tilde{K}$. The quantity $\Delta t = T/N$ can be computed, and therefore a new $(2M + 1) \times (N + 1)$ $(M = \lfloor 6\sqrt{N} \rfloor)$ grid describing the dynamics of the prices from the current time $t = 0$ to maturity $t = T$ according to formulas (17) and (18) is defined. This grid is a function of two indices: $\eta = 1, \ldots, N + 1$ and

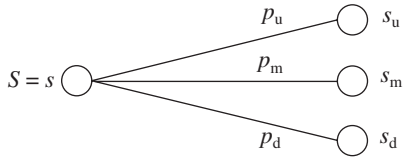

FIGURE 4. Trinomial tree: a representation of the trinomial tree (single period).

$\lambda = 1, \ldots, 2M+1$. The first index represents the time t_η while the second index identifies the price s_λ of the underlying security. Each node of the grid is distinguished by a pair (s_λ, t_η) whose values are defined as follows:

$$s_1 = \exp\bigl(-M\sigma_{\max}\sqrt{\Delta t}\bigr) < \cdots \cdots < s_\lambda = \exp\bigl[-(M-\lambda+1)\sigma_{\max}\sqrt{\Delta t}\bigr] < \cdots \cdots < s_{2M+1} = \exp\bigl(M\sigma_{\max}\sqrt{\Delta t}\bigr), \quad (19)$$

$$t_1 = 0 < \cdots < t_\eta = (\eta-1)\Delta t < \cdots < t_{N+1} = T. \quad (20)$$

Each node of the grid is associated with its own local volatility. It is worth observing that the nodes of this new grid are not necessarily coincident with those belonging to the grid $\tilde{\mathcal{K}} \otimes \mathcal{T}$. Consequently, a linear interpolation between the local volatility values $\sigma_{lr} \equiv \sigma(s_l, t_r)$ $(l, r = 1, 2, \ldots, d)$, computed on the discrete $d \times d$ square grid $\tilde{\mathcal{K}} \otimes \mathcal{T}$, is needed to implement the corresponding mapping on the values $\hat{\sigma}_{\lambda\eta} = \sigma(s_\lambda, t_\eta)$ to assign them to the new $(2M+1)\times(N+1)$ grid for pricing purposes.

Another extrapolation problem has already been introduced for the local volatility surface in the price direction. Indeed, the levels of the underlying price s_λ are not *a priori* controllable because they are defined using the formula (19), and so they can lie outside the range $\tilde{\mathcal{K}}$. This problem is different from that of the time extrapolation on the implied volatility surface, which we solved through a direct extrapolation on the forecasting function defined by means of the neural network. The extrapolation of the local volatility in the price direction is required for nodes that lie outside the domain $\tilde{\mathcal{K}}$. According to the particular nature of the problem to be solved (see the numerical experiments section), the authors adopted a solution that assumes constant volatility levels for those prices lying outside the domain $\tilde{\mathcal{K}}$. This policy is particularly important because it avoids the introduction of *subjective* biases.

Pricing contingent claims on the trinomial tree is a straightforward application of the backward induction rule. Indeed, the value of the option for any given time t_η and underlying value s_λ is obtained by discounting the expected value of the option payoff at time $t_{\eta+1}$ for the underlying prices $s_{\lambda+1}$, s_λ, and $s_{\lambda-1}$ with respect to the probability measure $(p_{\mathrm{u}}, p_{\mathrm{m}}, p_{\mathrm{d}})$. The probabilities p_{u}, p_{m}, p_{d} are functions of the local volatility of the node $\hat{\sigma}_{\lambda\eta} = \sigma(s_\lambda, t_\eta)$ according to the following formulas:

$$p_{\mathrm{u}} = \frac{\hat{\sigma}^2_{\lambda\eta}}{2\sigma^2_{\max}}\bigl(1 - \tfrac{1}{2}\sigma_{\max}\sqrt{\Delta t}\bigr), \quad p_{\mathrm{m}} = 1 - \frac{\hat{\sigma}^2_{\lambda\eta}}{\sigma^2_{\max}}, \quad p_{\mathrm{d}} = \frac{\hat{\sigma}^2_{\lambda\eta}}{2\sigma^2_{\max}}\bigl(1 + \tfrac{1}{2}\sigma_{\max}\sqrt{\Delta t}\bigr). \quad (21)$$

Starting from $\eta = N$ and moving backward to $\eta = 1$, the contingent claim price $\hat{V}_1^{M+1} = \hat{V}(s_{M+1}, t_1, \tilde{K})$ can be computed according to the following recursive

formula:

$$\hat{V}_{\eta}^{\lambda} = \exp(-r_{\mathrm{DM}}\Delta t)\left\lfloor \hat{V}_{\eta+1}^{\lambda+1} p_{\mathrm{u}} + \hat{V}_{\eta+1}^{\lambda} p_{\mathrm{m}} + \hat{V}_{\eta+1}^{\lambda-1} p_{\mathrm{d}} \right\rfloor, \tag{22}$$

where $\hat{V}_{N+1}^{\lambda} = \hat{V}(s_{\lambda}, t_{N+1}, \tilde{K})$ ($\lambda = 1, \ldots, 2M+1$) represents the payoff to maturity of the considered contingent claim, for the adjusted strike $\tilde{K}$ and for the underlying price s_{λ}. Then, for a call option,

$$\hat{V}_{N+1}^{\lambda} = (s_{\lambda} - \tilde{K})^{+}, \quad \lambda = 1, \ldots, 2M+1 \tag{23}$$

and, in a similar way, for a put option,

$$\hat{V}_{N+1}^{\lambda} = (\tilde{K} - s_{\lambda})^{+}, \quad \lambda = 1, \ldots, 2M+1. \tag{24}$$

For those nodes which lie on the border of the grid, where the tree is binomial, the value is computed as

$$\hat{V}_{\eta}^{1} = 2\hat{V}_{\eta}^{2} - \hat{V}_{\eta}^{3}, \quad \hat{V}_{\eta}^{2M+1} = 2\hat{V}_{\eta}^{2M} - \hat{V}_{\eta}^{2M-1}, \qquad \eta = 1, \ldots, N. \tag{25}$$

In the recursive scheme, the discounting factor $\exp(-r_{\mathrm{DM}}\Delta t)$ appears, given that we are in the real market with the last adjustment.

3. USD/DM OPTION PRICING

This section is devoted to the evaluation of the proposed option pricing scheme for USD/DM over-the-counter options, where data for the date of 23 August 1995 is considered. The evaluation of the case under study has been performed using the BLINNBOP software environment, which consists of the following modules:

1. *Bayesian learning in neural networks* (BLINN): this module, written in C, implements the Markov chain Monte Carlo (MCMC) solution procedure for the Bayesian learning problem on FNNs allowing one to perform the IVM step.
2. *Bayesian option pricer* (BOP): this module, which is written in MATLAB, receives the input from the BLINN module and implements the computational steps (OPC, LVC, and MD) to obtain the final option prices.
3. *Pricing analyzer* (PA): this module, which is written in MATLAB, allows the comparison of the original option prices with the option prices obtained using the proposed pricing scheme.

The three BLINNBOP software modules should be used in a strictly sequential fashion. The BLINN module allows us to approach the IVM computational task and to perform several statistical tests for checking the FNN model's significance. Once the FNN model is considered statistically significant, it can be automatically linked to the BOP software module. The BOP module exploits the

FNN model developed through the use of the BLINN module and enables three of the five steps of the option pricing scheme, namely, the OPC, the LVC, and the MD to be performed. At this stage, the BLINNBOP GUI allows for the pricing of general contingent claims. For each contract to be priced, the user must specify the option type (*call*, *put*), the underlying spot price S, the strike price K, the number of periods N, and the time to maturity T. Finally, the PA module, which offers its own GUI, allows one to compare the computed option prices with the prices from the market.

3.1 *The Option Data Set*

This section presents the main characteristics of the option data set used to investigate the performance of the proposed option pricing scheme. Furthermore, the transformation to be applied to the option data set, so that both the drift of the interest rate and the drift of the cost-of-carry are zero, is introduced. The option data set, presented in Table 1, is borrowed from Avellaneda and Paras (1996) and corresponds to the US dollar–Deutschmark

TABLE 1. USD/DEM OTC options: August 1995.

Maturity	Type	Strike	Bid	Ask	Mid	Ivol
	Call	1.5421	0.0064	0.0076	0.0070	14.9
	Call	1.5310	0.0086	0.0100	0.0093	14.8
30 days	Call	1.4872	0.0230	0.0238	0.0234	14.0
	Put	1.4479	0.0085	0.0098	0.0092	14.2
	Put	1.4371	0.0063	0.0074	0.0069	14.4
	Call	1.5621	0.0086	0.0102	0.0094	14.4
	Call	1.5469	0.0116	0.0135	0.0126	14.5
60 days	Call	1.4866	0.0313	0.0325	0.0319	13.8
	Put	1.4312	0.0118	0.0137	0.0128	14.0
	Put	1.4178	0.0087	0.0113	0.0100	14.2
	Call	1.5764	0.0101	0.0122	0.0112	14.1
	Call	1.5580	0.0137	0.0160	0.0149	14.1
90 days	Call	1.4856	0.0370	0.0385	0.0378	13.5
	Put	1.4197	0.0141	0.0164	0.0153	13.6
	Put	1.4038	0.0104	0.0124	0.0114	13.6
	Call	1.6025	0.0129	0.0152	0.0141	13.1
	Call	1.5779	0.0175	0.0207	0.0191	13.1
180 days	Call	1.4823	0.0494	0.0515	0.0505	13.1
	Put	1.3902	0.0200	0.0232	0.0216	13.7
	Put	1.3682	0.0147	0.0176	0.0162	13.7
	Call	1.6297	0.0156	0.0190	0.0173	13.3
	Call	1.5988	0.0211	0.0250	0.0226	13.2
270 days	Call	1.4793	0.0586	0.0609	0.0598	13.0
	Put	1.3710	0.0234	0.0273	0.0254	13.2
	Put	1.3455	0.0173	0.0206	0.0190	13.2

TABLE 2. Market parameters.

Description	Symbol	Value
Spot price	S	1.4885 DM
DM interest rate	r_{DM}	4.27%
USD interest rate	r_{USD}	5.91%

(USD/DM) over-the-counter (OTC) options for 23 August 1995. This data set consists of 25 option prices corresponding respectively to 20, 25, and 50 delta puts and calls, with expiration dates of 30, 60, 90, 180, and 270 days, respectively, i.e., $T = 30,\ 60,\ 90,\ 180,\ 270$. The implied volatility (Ivol) is reported in the last column. The data listed in Table 1 depends on the following market parameters: spot price S, DM interest rate r_{DM}, and USD interest rate r_{USD} whose values are reported in Table 2.

According to formula (6) in Section 2, the following transformation allows one to obtain zero drift and zero cost-of-carry:

$$\tilde{K} = f(K, T, \rho) = f(K, T, r_{\mathrm{USD}} - r_{\mathrm{DM}}) = K \exp[(r_{\mathrm{USD}} - r_{\mathrm{DM}})T]. \qquad (26)$$

The new data set, obtained via the application of the transformation (26) to the data reported in Table 1, is given in Table 3 and represents the data set ($D = 25$) used for solving the learning problem.

3.2 *Implied Volatility Modeling (IVM Step)*

The FNN used to approximate the implied volatility surface $\Sigma(\tilde{K}, T)$ is a single layer neural network consisting of two input neurons ($I = 2$), associated respectively with the adjusted strike price $\tilde{K}$ and the time to maturity T, 20 hidden neurons ($m = 20$) with logistic activation functions (1), and one output neuron, associated with the approximated implied volatility surface $\hat{\Sigma}(\tilde{K}, T)$, with logistic activation function (1).

The approximation of the implied volatility surface, computed via the FNN, is shown in Figure 5. Notice that the learning data set is obtained from the data reported in Table 3 by dividing the implied volatility (Ivol) by 100 and by transforming the time to maturity (Maturity) on a year basis, i.e., dividing the time to maturity by 360.

In order to highlight the systematic behavior of the market with respect to the term structure (changes in volatility due to maturity fluctuations) and the strike structure (changes in volatility due to strike price fluctuations), it is interesting to analyze some sections of the approximated implied volatility surface shown in Figure 5.

The behavior of the approximated implied volatility surface with respect to the time to maturity is depicted in Figure 6 while the cross-section of the approximated implied volatility surface in Figure 7 allows one to investigate graphically the behavior of the approximated implied volatility with respect to the adjusted strike price.

TABLE 3. USD/DM OTC options (zero drift and zero cost-of-carry).

Maturity	Type	Adjusted strike	Bid	Ask	Mid	Ivol
	Call	1.544209	0.0064	0.0076	0.0070	14.9
	Call	1.533094	0.0086	0.0100	0.0093	14.8
30 days	Call	1.489234	0.0230	0.0238	0.0234	14.0
	Put	1.449880	0.0085	0.0098	0.0092	14.2
	Put	1.439065	0.0063	0.0074	0.0069	14.4
	Call	1.566376	0.0086	0.0102	0.0094	14.4
	Call	1.551134	0.0116	0.0135	0.0126	14.5
60 days	Call	1.490669	0.0313	0.0325	0.0319	13.8
	Put	1.435117	0.0118	0.0137	0.0128	14.0
	Put	1.421681	0.0087	0.0113	0.0100	14.2
	Call	1.582877	0.0101	0.0122	0.0112	14.1
	Call	1.564401	0.0137	0.0160	0.0149	14.1
90 days	Call	1.491703	0.0370	0.0385	0.0378	13.5
	Put	1.425533	0.0141	0.0164	0.0153	13.6
	Put	1.409567	0.0104	0.0124	0.0114	13.6
	Call	1.615695	0.0129	0.0152	0.0141	13.1
	Call	1.590892	0.0175	0.0207	0.0191	13.1
180 days	Call	1.494505	0.0494	0.0515	0.0505	13.1
	Put	1.401647	0.0200	0.0232	0.0216	13.7
	Put	1.379465	0.0147	0.0176	0.0162	13.7
	Call	1.649869	0.0156	0.0190	0.0173	13.3
	Call	1.618587	0.0211	0.0250	0.0226	13.2
270 days	Call	1.497608	0.0586	0.0609	0.0598	13.0
	Put	1.387967	0.0234	0.0273	0.0254	13.2
	Put	1.362152	0.0173	0.0206	0.0190	13.2

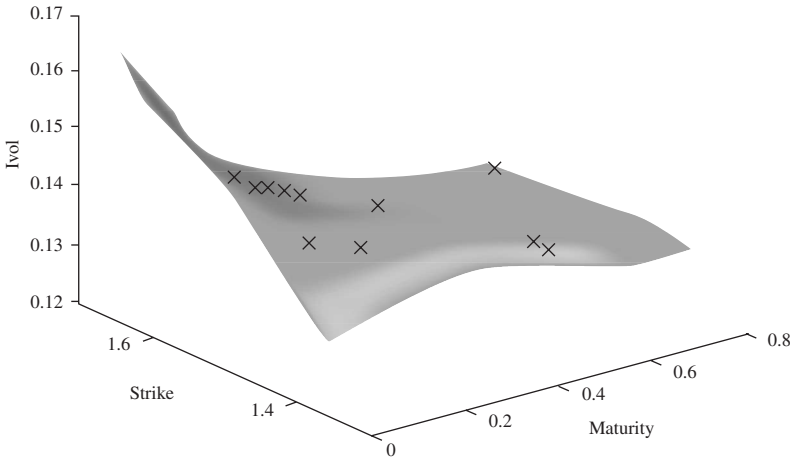

FIGURE 5. Approximated implied volatility surface: the FNN approximated implied volatility surface $\hat{\Sigma}(\tilde{K}, T)$ (mesh) and the corresponding learning set (crosses) for the USD/DM options market.

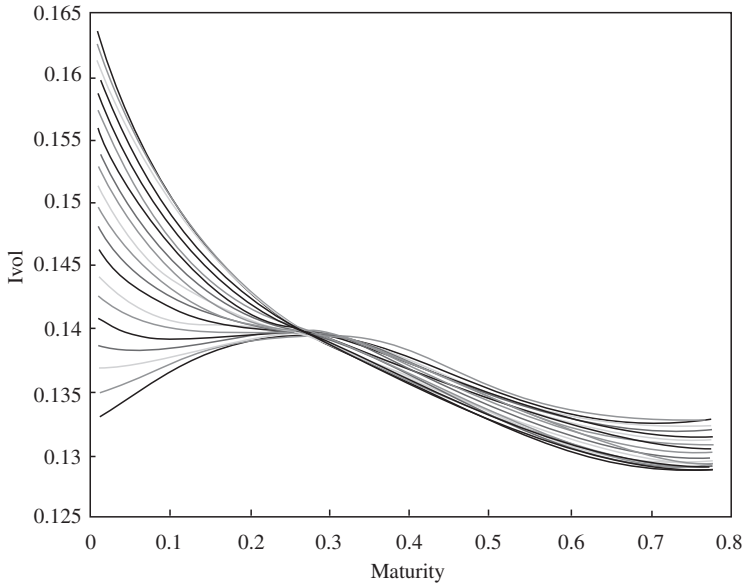

FIGURE 6. Term structure for the approximated implied volatility surface: the term structure highlights a particular shape. Long-term implied volatility is, in general, lower than the short-term one. Moreover, short-term implied volatility shows higher unsteadiness than the long-term one due to big changes in volatility values with respect to different adjusted strike prices.

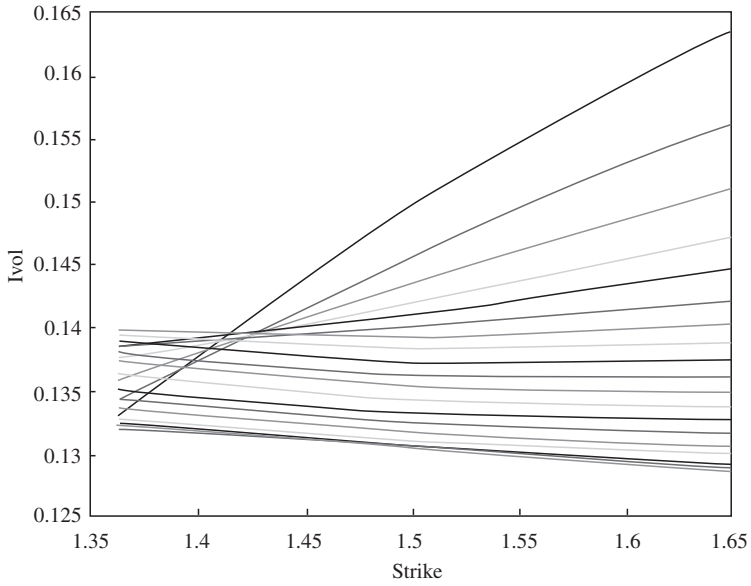

FIGURE 7. Strike structure for the approximated implied volatility surface: the strike structure is characterized by a light smile effect, indeed the implied volatility decreases when the adjusted strike price increases.

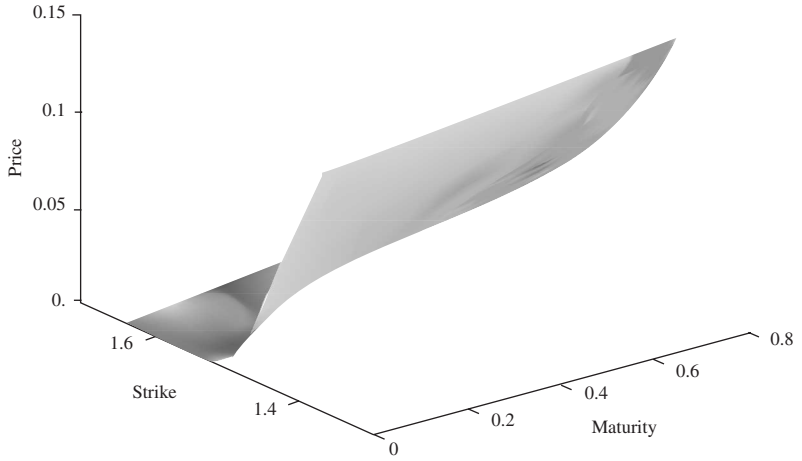

FIGURE 8. Prices surface: the approximated prices surface.

3.3 *Option Price Computation (OPC Step)*

The output of this computational step is the so-called *prices surface* (Figure 8) which has been obtained using a spot price $S = 1.4885$ synchronously observed on the market together with the other data (Tables 1 and 2).

3.4 *Local Volatility Computation (LVC Step)*

The third computational step allows one to compute the local volatility surface together with its first derivative with respect to time to maturity T (Figure 9) and its second derivative with respect to the adjusted strike price $\tilde{K}$ (Figure 10). Both these figures bring valuable information about the numerical properties of the proposed solution for the option pricing problem.

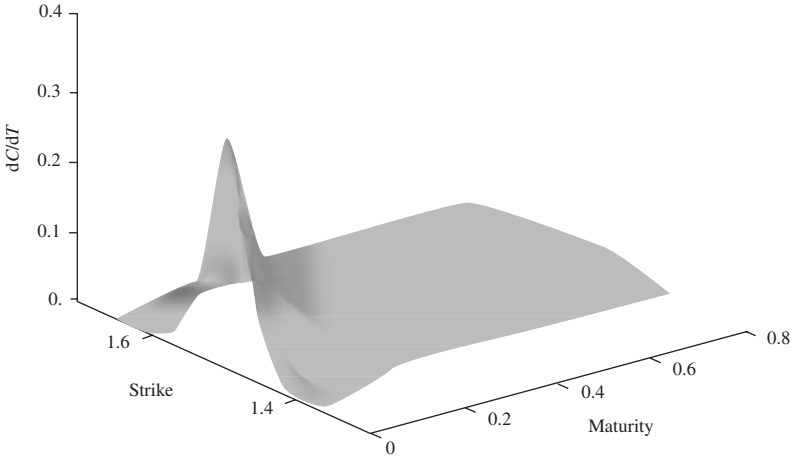

FIGURE 9. First derivative of the call option price $C(S, \tilde{K}, T)$ with respect to time to maturity.

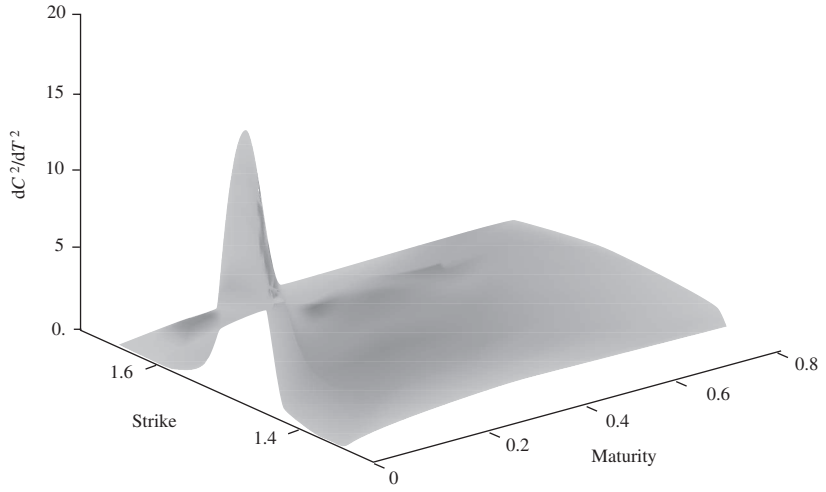

FIGURE 10. Second derivative of the call option price $C(S, \tilde{K}, T)$ with respect to adjusted strike.

From Figures 9 and 10, it is possible to conclude that the computational model utilized for the LVC step shares nice numerical properties. Indeed, the two figures show smooth surfaces, suggesting that the derivatives estimated through the finite-difference scheme seem to provide an accurate approximation for the true derivatives. A further graphical analysis (Figure 11) of the nature of the local volatility surface can be performed by exploiting the features of the BOP software module.

Figures 12 and 13 show the behavior of the term and spot structures (the adjusted strike price is replaced by the underlying spot price of the local volatility surface).

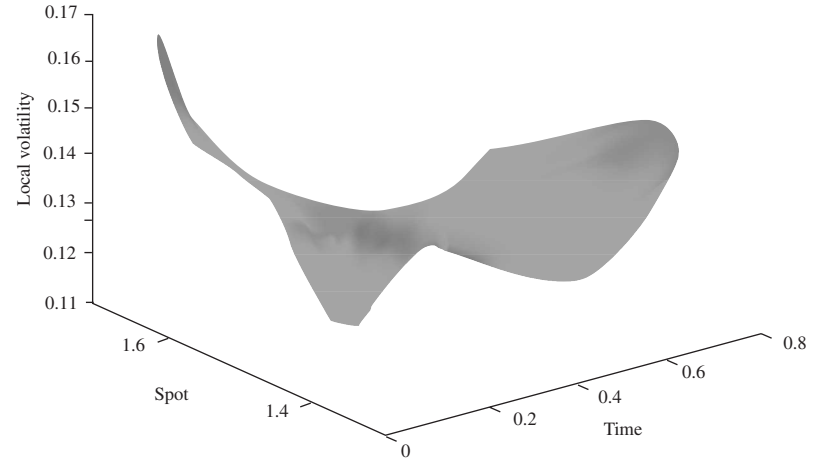

FIGURE 11. Local volatility surface: an approximation of the local volatility surface.

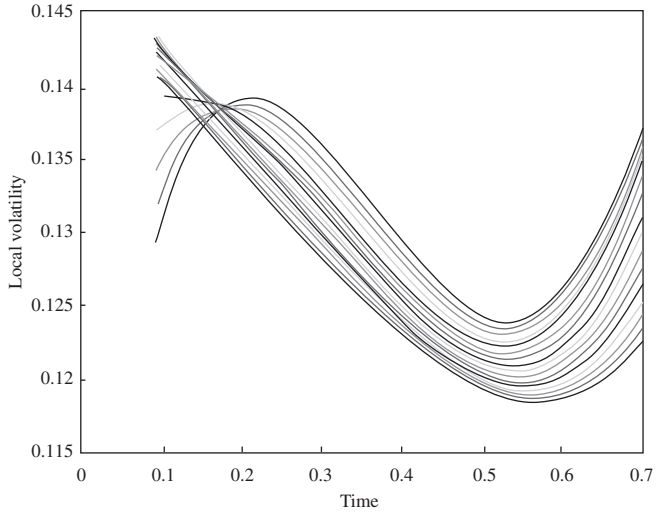

FIGURE 12. Term structure: a cross-section of the term structure for the approximated implied volatility surface depicted in Figure 5.

3.5 *Model Discretization (MD Step)*

The fourth step (MD), performed through the BOP module, allows one to obtain the approximated local volatility surface shown in Figure 14. Notice that in Figure 14 the extrapolation is required for those values lying outside the range $\tilde{\mathcal{K}}$. Indeed, while the initial prices belong to the interval [1.362152, 1.649869], i.e., the minimum and maximum values for the adjusted strike prices reported in

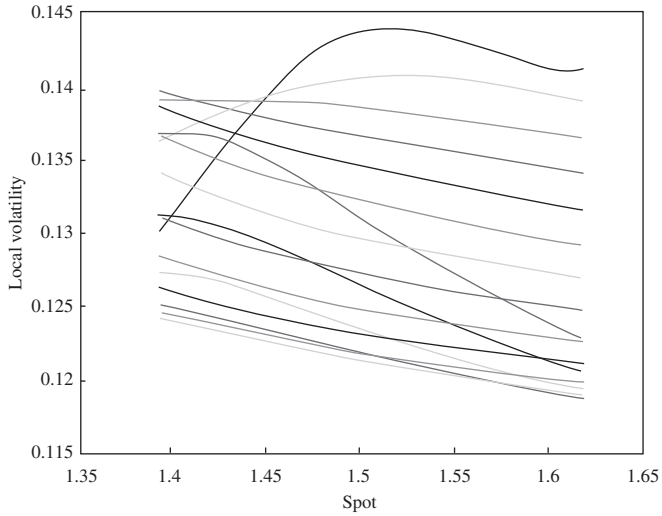

FIGURE 13. Spot structure: a cross-section of the spot structure for the approximated implied volatility surface depicted in Figure 5.

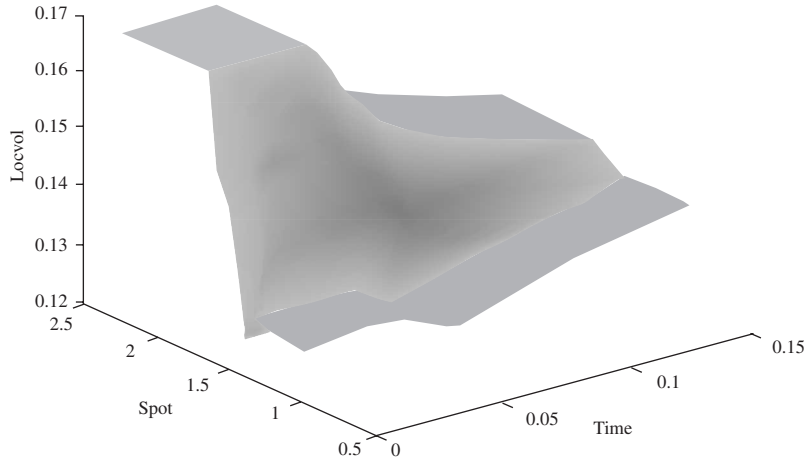

FIGURE 14. Local volatility surface mapping: the approximated local volatility surface mapped on the trinomial stochastic volatility model.

TABLE 4. USD/DM OTC options: Mid versus computed mid-market option prices.

Maturity	Type	Adjusted strike	Mid V	Computed mid $\hat{V}$	δ
	Call	1.544209	0.0070	0.0069720	+0.4
	Call	1.533094	0.0093	0.0091977	+1.1
30 days	Call	1.489234	0.0234	0.0235170	−0.5
	Put	1.449880	0.0092	0.0091448	+0.6
	Put	1.439065	0.0069	0.0068034	+1.4
	Call	1.566376	0.0094	0.0094094	−0.1
	Call	1.551134	0.0126	0.0125370	+0.5
60 days	Call	1.490669	0.0319	0.0321233	−0.7
	Put	1.435117	0.0128	0.0127360	+0.5
	Put	1.421681	0.0100	0.0099400	+0.6
	Call	1.582877	0.0112	0.0111440	+0.5
	Call	1.564401	0.0149	0.0147063	+1.3
90 days	Call	1.491703	0.0378	0.0382158	−1.1
	Put	1.425533	0.0153	0.0152694	+0.2
	Put	1.409567	0.0114	0.0115482	−1.3
	Call	1.615695	0.0141	0.0140436	+0.4
	Call	1.590892	0.0191	0.0193674	−1.4
180 days	Call	1.494505	0.0505	0.0507525	−0.5
	Put	1.401647	0.0216	0.0214272	+0.8
	Put	1.379465	0.0162	0.0160380	+1.0
	Call	1.649869	0.0173	0.0174038	−0.6
	Call	1.618587	0.0226	0.0225322	+0.3
270 days	Call	1.497608	0.0598	0.0606372	−1.4
	Put	1.387967	0.0254	0.0257302	−1.3
	Put	1.362152	0.0190	0.0192280	−1.2

Table 3, the prices computed by means of the trinomial tree range in the interval

$$\left[\exp\left(\lfloor -6\sqrt{N}\rfloor\sigma_{\max}\sqrt{\frac{T}{N}}\right), \exp\left(\lfloor 6\sqrt{N}\rfloor\sigma_{\max}\sqrt{\frac{T}{N}}\right)\right]$$

and therefore, for the case under study, lie outside the interval $\tilde{\mathcal{K}}$.

The numerical comparison between the market and the computed (estimated) prices is given in Table 4. The analysis of the results reported in Table 4, i.e., the quantitative comparison between the market and the computed prices, suggests that the overall option pricing procedure is capable of accurately computing the mid-market option price V. Indeed, the relative error δ, defined as

$$\delta = 100\left(\frac{V - \hat{V}}{V}\right), \tag{27}$$

is very small, ranging from -1.4% (observations 17 and 23) to 1.4% (observation 5). This characteristic indicates that the overall option pricing scheme provides estimates $\hat{V}$ for the mid-market option prices V which are very close to those effectively observed on the real market.

Figure 15, the graph of the mid-market option price V versus the computed mid-market option price $\hat{V}$, very closely resembles a 45° straight line lying in the first quadrant. This shape suggests that, in the case under study, the proposed option pricing scheme does not introduce any systematic bias for the quantity to be estimated, namely, the mid-market option price V.

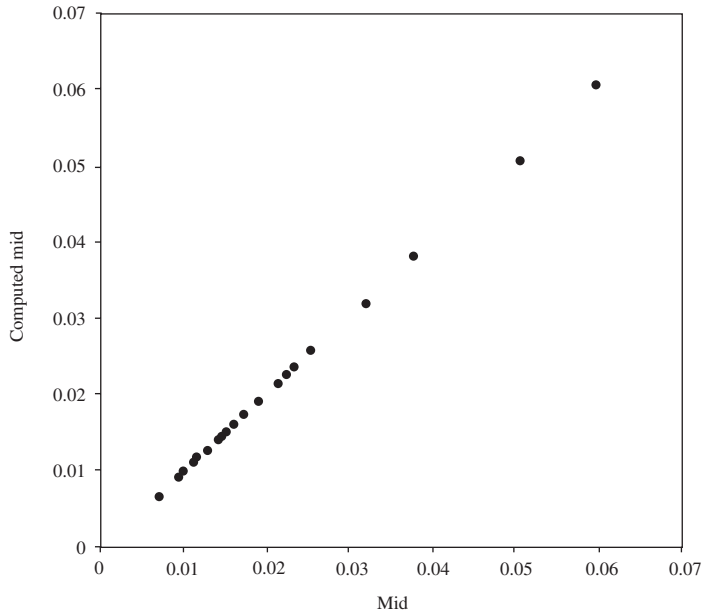

FIGURE 15. Mid versus computed mid-market option price scatter plot: a scatter plot for the comparison between the mid-market option price V and the computed mid-market option price $\hat{V}$.

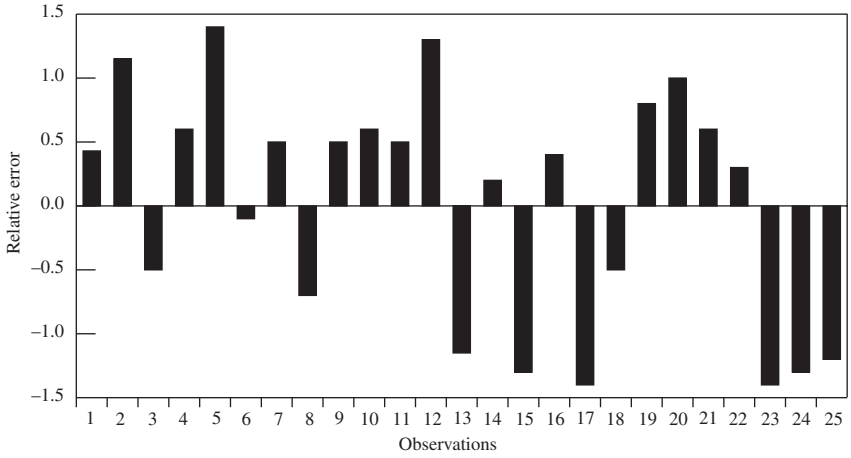

FIGURE 16. Relative error histogram plot: a histogram plot of the relative error δ with respect to the 25 data points (option contracts).

Inspection of Figure 16 does not reveal any systematic pattern for the relative error (27). Indeed, it can be noticed that in 14 out of 25 times the mid-market option price V is underestimated (positive δ values) while in the remaining 11 cases the mid-market option price is overestimated (negative δ values). Furthermore, even the sequence of the relative errors δ along the data point axes does not show any systematic pattern.

Finally, an interesting property of the proposed pricing scheme emerges from the joint analysis of the data reported in Tables 3 and 4. Indeed, the computed mid-market option price $\hat{V}$ is always contained in the bid–ask interval, thus reflecting the intrinsic uncertainty of the option market.

4. CONCLUSIONS AND DIRECTIONS FOR FURTHER RESEARCH

In this paper, the authors propose a new option pricing scheme which exploits recent developments about learning in FNNs coupled with the Dupire formula. Indeed, while the Bayesian learning scheme for FNN allows the data to speak for itself and therefore to develop an accurate approximation of the implied volatility surface, the Dupire formula provides a mathematical way of computing the local volatility function from the implied volatility surface. The numerical experiments, performed using the USD/DM OTC options data, provide strong empirical evidence in favor of the proposed pricing scheme. According to this evidence, it seems to be possible to pass from a set of liquid option prices to a pricing system which allows one to evaluate other derivatives whose prices are not readily available from the market, i.e., illiquid European options, American options, and exotic options. Furthermore, the proposed pricing system can be used for pricing barrier options, where the probability

of striking the barrier is sensitive to the shape of the smile, creating static hedge portfolios, pricing exotic options, and generating Monte Carlo distributions for valuing path-dependent options.

Acknowledgements

The authors are grateful to the anonymous referees whose insightful comments enabled significant improvements to be made to this paper.

REFERENCES

Avellaneda, M., Friedman, C., Holmes, R., and Samperi, D. (1997). Calibrating volatility surfaces via relative-entropy minimization. *Applied Mathematical Finance*, **4**, 37–64.

Avellaneda, M., and Laurence, P. (2000). *Quantitative Modeling of Derivative Securities: From Theory to Practice*. Chapman & Hall/CRC.

Avellaneda, M., Levy, A., and Paras, A. (1995). Pricing and hedging derivative securities in markets with uncertain volatilities. *Applied Mathematical Finance*, **2**, 73–88.

Avellaneda, M., and Paras, C. (1996). Managing the volatility risk of portfolios of derivative securities: The Lagrangian uncertain volatility model. *Applied Mathematical Finance*, **3**, 21–52.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford.

Black, F., and Scholes, M. (1973). The Pricing of options and corporate liabilities. *Journal of Political Economy*, **8**, 637–654.

Chen, W.-Y., Chen, S.-H., and Lin, C.-J. (1996). A speech recognition method based on the sequential multi-layer perceptrons. *Neural Networks*, **9**(4), 655–669.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematical Control Signals Systems*, **2**, 303–314.

Derman, E., and Kani, I. (1998). Stochastic implied trees: Arbitrage pricing with stochastic term and strike structure of volatility. *International Journal of Theoretical and Applied Finance*, **1**, 61–110.

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

Fouque, J., Papanicolaou, G., and Sircar, R. (2000). Mean-reverting stochastic volatility. *International Journal of Theoretical and Applied Finance*, **3**(1), 101–142.

Fu, K. S. (1982). *Syntactic Pattern Recognition and Applications*. Prentice-Hall.

Gader, P., Forester, B., Ganzberger, M., Gilles, A., Mitchell, B., Whalen, M., and Yocum, T. (1991). Recognition of handwritten digits using template and model matching. *Pattern Recognition*, **24**, 421–431.

Hertz, J., Krogh, A., and Palmer, R. G. (1991). *Introduction to the Theory of Neural Computation*. Addison-Wesley.

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

Hornik, K., Stinchombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, **2**, 359–366.

Hutchinson, J. M., Lo, A., and Poggio, T. (1994). A nonparametric approach to pricing and hedging derivative securities via learning networks. *Journal of Finance*, **49**, 851–889.

Kung, S. Y. (1993). *Digital Neural Networks*. Prentice-Hall.

McKay, D. J. C. (1992). Bayesian interpolation. *Neural Computation*, **4**, 415–447.

Neal, R. M. (1996). *Bayesian Learning for Neural Networks*, Lecture Notes in Statistics 118. Springer.

Rebonato, R. (1999). *Volatility and Correlation*. Wiley.

Rissanen, J. (1996). Modeling by shortest data description. *Automatica*, **14**, 465–471.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society* B, **36**, 111–147.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer.

Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley.

Wahba, G., and Wold, S. (1975). A completely automatic french curve: Fitting spline functions by cross-validation. *Communications in Statistics* A, **4**, 1–17.

White, H. (1989). Learning in artificial neural networks: A statistical perspective. *Neural Computation*, **1**, 425–464.

White, H. (1990). Connectionist nonparametric regression: Multilayer feedforward networks can learn arbitrary mappings. *Neural Networks*, **3**, 535–549.

Yacine, A. S. (1996). Nonparametric pricing of interest rate derivative securities. *Econometrica*, **64**, 527–560.