Neural Networks and the Valuation of Derivatives – Some Insights into the Implied Pricing Mechanism of German Stock Index Options*

Ralf Herrmann† and **Alexander Narr**

November 10, 1997

Abstract

Following an approach proposed by Hutchinson/Lo/Poggio[1994], Kelly[1994] and Malliaris/Salchenberger[1993], we used neural networks to value derivatives. We first examined the ability of the used neural networks to interpolate the Black & Scholes formula and its derivatives. In a second step we trained neural networks on real world data from the Deutsche Terminbörse (DTB). We used about 500,000 trading prices of stock index options on the Deutscher Aktienindex (DAX) to approximate the implied pricing formulas of the market. Looking at the partial derivatives of the implied pricing formular allows us to get deeper insights into the pricing process.

It can be shown that the implied pricing formulas differ markable from the Black & Scholes formula. The results suggest that the implied pricing formulas account for the correlations between interest rates and the DAX and that increasing volatility has a negative impact on in–the–money options. A "crash–o–phobia" phenomenon is observed, too.

*We are grateful for comments of participants at the 1996 meeting of the German Finance Association in Berlin and at the Tenth Annual CBOT European Futures Research Symposium 1997 in London. Of course, the usual disclaimer applies.

†University of Karlsruhe, Institute for Decision Theory and Management Science, Department of Finance and Banking,76133 Karlsruhe, Germany. eMail:`Ralf.Herrmann@wiwi.uni-karlsruhe.de`

1 Introduction

The standard approach to value derivative securities is based on the explicit specification of the stochastic price process of the underlying asset. Much research has been done within this approach, beginning with the seminal articles of Black/Scholes[1973] and Merton[1973]. However, empirical observations, such as fat–tailed return distributions, the smile effect or "crash–o–phobia" can not be explained by standard option pricing models.[1] Furthermore, empirical results are casting doubt on popular parametric specifications.[2] These shortcomings have increased the interest in implied approaches to option pricing. The fundamental idea is to use observed market prices to get information about the beliefes of the market participants, the implied return distributions or the implied pricing mechanism of the market.[3] Such information can be used to improve standard option pricing models, to reveal arbitrage opportunities or to implement hedging strategies. The roots of implied approaches are dating back to Latané/Rendleman[1976], which used market prices of options to calculate implied volatilities. Rubinstein[1994] among others extended this approach by using option prices to calculate implied state–price–densities.[4]

In our study, we followed an approach suggested by Hutchinson/Lo/Poggio[1994], Kelly[1994] and Malliaris/Salchenberger[1993]: Estimating the implied pricing mechanism of DAX-options traded at the Deutsche Termin Börse (DTB) by "training" neural networks on trading prices.

Due to their characteristics neural networks seem to be an appropriate statistical approximation method if one or more of the following conditions are met:

- The patterns looked for, are subtle or deeply hidden in the data available for the estimation.
- The relationship to be discovered exhibit significant unpredictable nonlinearity.
- The partial derivatives of the approximation function are of interest.

In our specific field of application all mentioned conditions are met. Existing evaluation models show a highly nonlinear structure. The derivatives are necessary for the implementation of hedging strategies. Furthermore they allow to get deeper insights into the pricing mechanism of the market and to prove the economic content of the formula.

In contrast to more traditional models, neural networks do not need restrictive assumptions about the function to be estimated, for example assumptions like log–normal distributed stock returns or sample path continuity. Since they do not rely on restrictive assumptions they are very robust to specifications errors parametric valuation methods often suffer from. Due to their adaptive nature they are able to handle structural changes in the pricing process. Futhermore once "trained" they are easy to implement and easy to handle. But they are highly data intensiv. A lot of historical data is necessary to get sufficiently well–trained networks.

To summarize, neural networks provide a powerful nonparametric, data driven pricing method which allows the data to determine the dynamics of the underlying asset and its

[1] "Crash–o–phobia" describes the fact that implied state–prices of index–options for low index levels are higher than expected. This can be interpreted as as additional insurance premium against crashes. For details see Rubinstein[1994].

[2] See for example Lo/MacKinlay[1988].

[3] A literatur overview is given by Mayhew[1995].

[4] Results for German stock index options can be found in Neumann/Schlag[1996].

relation to the prices of the derivatives. In contrast to the main part of empirical studies on neural networks in Finance – we do not want to predict anything and in contrast to such studies we know that a functional relationship between the input factors of our network and the target value — the price of the derivative — definitely exists, but not its specific form.

First examples of successful applications of neural networks for the valuation of derivative securities were provided by earlier studies. Tabular 1 summarizes such studies:[5]

Table 1:
Empirical Studies

Author	Period	Derivative[6]	Model[7]	Network[8]
Boeck et al.[1995]	1993-1994	AO SPI future option(a)	BS	?
Hutchinson/Lo/Poggio[1994]	1987-1991	S&P 500 futures option(a)	L,BS	MLP,RBF,PPR
Kelly[1994]	1993-1994	Stock-Options(a)	CRR	MLP
Lajbcygier et al. [1995]	1993-1994	AO SPI future option(a)	L,BS,BW	MLP
Malliaris/Salchenberger[1993]	1990	S&P 100 OEX option(a)	BS	MLP
Qi/Maddala[1995]	1994-1995	S&P 500 options(a)	BS	MLP
White[1995]	1994	S&P 100 OEX option(a)	BS	ENT

Our study extends the existing research in some directions: First our study is founded on time-stamped intraday data[9] of all traded options, so not restricted to closing prices or to a easy to manage part of the market. Second, we used call as well as put options to be able to compare the formulas of both typs of options. Third, we show how well-suited neural networks are for the interpolation of the derivatives of pricing formulas. Forth, we used the obtained pricing formulas to get deeper insights into the implied pricing mechanism and compared it to the B&S model. Fifth we derived the implied state–price–densities without using any restrictive assumption — like positivity restrictions or the functional form of the state–price–densities (SPD). So demonstrating a new approach for the derivation of implied SPD's .

The remainder ist organized as follows. The next section provides a brief introduction to the theoretical background of our study. Section 3 describes the methodology and the data used, while section 4 provides the empirical results. Section 5 contains a short summary and concluding remarks. The appendix contains the derivation of the partial derivatives of our networks and the estimated network parameters.

[5]Simultaneously and independent to our study Anders/Korn/Schmitt[1996] applied neural networks to DAX options.

[8]american(a), european(e)

[8]Model used for comparison: Black/Scholes(BS)[1973], Barone-Adesi/Whaley(BW)[1987], Cox/Ross/Rubinstein(CRR)[1979], Linear Approximation(L).

[8]Entropy Network(ENT), Multilayer Perceptron(MLP), Projection Pursuit Regression(PPR), Radial Basis Functions(RBF).

[9]Especially we took a lot of care over the preparation of the database and the input parameters, to avoid problems resulting from non-synchronous data or incomplete and small datasets some earlier studies suffer from. Our database exceeds the databases of all earlier studies many times over.

2 Theoretical Background

Object of our investigation are financial derivatives, more specific index options. Options give the holder the right to buy (call option) or to sell (put option) the underlying asset for a certain price (exercise price) by a certain day (expiration day). Option which can be exercised at any time up to expiration are called American options, options which only can be exercised on the expiration itself are called European options. An index option is an option where the underlying is an index. For the remainder of this article only European options are considered. Option prices are generally denoted by O whereas call prices are denoted by C and put prices by P.

2.1 Option–Pricing

It is possible to derive upper and lower bounds for European options without any particular assumption[10] about the factors affecting the option price. In an arbitrage–free market the following conditions have to be met:

$$\begin{array}{lcc} \hline C(t) & \leq & S(t) \\ P(t) & \leq & Xe^{-rT} \\ \hline C(t) & > & \max[S - Xe^{-rT}, 0] \\ P(t) & > & \max[Xe^{-rT} - S, 0] \\ \hline \end{array} \tag{1}$$

X : exercise price,

r : riskless interest rate,

T : time to expiry expressed as a fraction of a year and

S : underlying asset.

This bounds determine a range in which options price should stay, otherwise investors are able to make riskless profits. The fundamental idea introduced by Black/Scholes[1973] and Merton[1973] to get a explicit pricing formula was to replicate options by dynamic hedging strategies. In an arbitrage-free environment such self-financing hedging strategies have to have the same initial value as the replicated option. Although introduced in 1973 and often extended, the standard B&S formula is unquestionable one of the most relevant pricing formula in practice.

Assuming arbitrage-free frictionless markets, continous trading, constant and same risk-free interest rates for all maturities and a geometric Brownian motion with constant drift and variance rate as the stochastic price process for the underlying, the following pricing formulas can be derived[11] for calls (C^{BS}) and puts (P^{BS}):

$$\begin{array}{lcl} \hline C^{BS} & = & S(t)N(d_1) - Xe^{-rT}N(d_2) \\ P^{BS} & = & Xe^{-rT}N(-d_2) - SN(-d_1) \\ \hline \end{array} \tag{2}$$

[10]Except $r > 0$, which in reality is no very restrictive assumption at all.

[11]The correct derivation can be found in Merton[1973].

where

$$
\begin{array}{ccc}
\hline
d1 & = & \frac{\ln(\frac{S(t)}{X})+(r+\frac{\sigma^2}{2})T}{\sigma\sqrt{T}} \\
d2 & = & d1 - \sigma\sqrt{T} \\
\hline
\end{array}
\tag{3}
$$

and

N(.) : the standard cumulative normal distribution function and

σ : standard deviation of the instantaneous rate of return on S.

The local arbitrage argument used by Black/Scholes[1973] was extended by Rubinstein/Leland[1981] to create actually options.[12] The key idea was instead of building a risk-free portfolio by dynamically hedging an option with stock, to create the option by a dynamic strategy which involved the underlying and a risk-free asset. Essential for such a strategy is to know how the replicated option responds to changes in the underlying option variables. Assuming a B&S world the partial derivatives[13] are as follows[14]:

$$
\begin{array}{ccccc}
\hline
\text{Delta}_c & = & \frac{\partial C}{\partial S} & = & N(d_1) \\
\text{Delta}_p & = & \frac{\partial P}{\partial S} & = & N(d_1) - 1 \\
\hline
\text{Gamma}_c & = & \frac{\partial \text{Delta}_c}{\partial S} & = & \frac{n(d_1)}{S\sigma\sqrt{T}} \\
\text{Gamma}_p & = & \frac{\partial \text{Delta}_p}{\partial S} & = & \frac{n(d_1)}{S\sigma\sqrt{T}} \\
\hline
\text{Vega}_c & = & \frac{\partial C}{\partial \sigma} & = & Sn(d_1)\sqrt{T} \\
\text{Vega}_p & = & \frac{\partial P}{\partial \sigma} & = & Sn(d_1)\sqrt{T} \\
\hline
\text{Rho}_c & = & \frac{\partial C}{\partial r} & = & TXe^{-rT}N(d_2) \\
\text{Rho}_p & = & \frac{\partial P}{\partial r} & = & -TXe^{-rT}N(-d_2) \\
\hline
\text{Theta}_c & = & \frac{\partial C}{\partial T} & = & Sn(d1)\frac{\sigma}{2\sqrt{T}} + rXe^{-rT}N(d_2) \\
\text{Theta}_p & = & \frac{\partial P}{\partial T} & = & Sn(d1)\frac{\sigma}{2\sqrt{T}} - rXe^{-rT}N(-d_2) \\
\hline
\end{array}
\tag{4}
$$

This functions show how the prices[15] of calls and puts changes as one of the input parameter changes. Assuming[16] furthermore the existence of state–price–densities (SPD's) and the dynamic completeness of the market we are able to derive the implied prices of "Arrow-Debreu" securities[17], which correspond to the second derivative of the pricing functions with respect to the strike price X evaluated at S_T . This gives us the the price for a security paying DM 1.– if the state falls between S and $S + dS$. In a B&S world the SPD corresponds to the following log-normal distribution:[18]

$$
\text{SPD} = e^{rT}\frac{\partial^2 C}{\partial X^2} = e^{rT}\frac{\partial^2 P}{\partial X^2} = \frac{1}{S_T\sqrt{2\pi\sigma^2 T}} e^{-\frac{\left(\ln(S_T/S_t)-(r-\sigma^2/2)T\right)^2}{2\sigma^2 T}}. \tag{5}
$$

[12]The evolution of the approach is described in Leland/Rubinstein[1988].
[13]The derivation can be found for example in Stoll/Whaley[1993], p. 245ff.
[14]$n(.)$ denotes the density of a standard normal distribution.
[15]Gamma shows how the delta of the option changes.
[16]Note:We do not need any assumption about the price process of the underlying.
[17]Arrow[1964],Debreu[1959].
[18]Breeden/Litzenberger[1978], p. 630.

3 Data and Methodology

3.1 Data

Our study was founded on time-stamped transaction data on the DAX option traded at the DTB, after the S&P 100 and S&P 500 options the most liquid index option worldwide[19]. All data used was provided by the Karlsruher Kapitalmarktdatenbank (KKMDB).

The DAX option[20] is an European style option on the DAX index. The DAX[21] contains the 30 biggest and most liquid German stocks, which together have a market capitalization of over 60 percent on all German stocks listed at exchanges. Together they account for over 75 percent of the total German stock trading volume. The option has a contract size of DM 10 per index point of the DAX and is quoted in points with one decimal place. Its minimum price move is 0.1 point which corresponds to DM 1. The maturity of the contracts range up to 24 months[22]. At least five exercise prices for each contract month are introduced initially. New option series are introduced continuously if the DAX exceeds the average of the third- and second-highest or falls below the average of the third- and second-lowest currently existing strike price. The DAX-option is traded daily between 9:00 a.m. and 5:00 p.m. Because the DAX[23] is a dividend adjusted performance index and the DAX option traded at the DTB is of European type, there were no dividend and early exercise problems. Hence the DAX option has ideal contract specification for the application of the B&S formula.

To get synchronized datasets[24] it was necessary to calculate simultaneous historical DAX values for each trading day from 9:00 a.m. to 5:00 p.m.[25] This was accomplished using IBIS[26] data.[27] Since June 15, 1995 our index correponds with the IBIS-DAX provided by the Deutsche Börse AG[28].

The interest rates used are Frankfurt Interbank Offer (FIBOR) rates. For intermediate times, interest rates were interpolated linear. We used the VDAX[29] as an average weighted volatility expectation of the market.

The period of observation was from January 1, 1995 to December 31, 1995. In the whole period 514,192 trades in DAX options divided up into 266,300 trades in calls and 247,892

[19] DBAG[1996], p. 26.

[20] See DBAG[1996].

[21] For details see DBAG[1995b].

[22] Options with maturities of 18 and 24 months were introduced on March 18, 1996.

[23] See Deutsche Börse AG (1995b).

[24] For each transaction price the corresponding DAX value is needed.

[25] The weighting factors necessary for this calculation were provided by the Deutsche Börse AG.

[26] Integriertes Börsenhandels- und Informationssystem (Integrated Stock Exchange Trading and Information System), an electronic trading system.

[27] For a description of the datasets used, see Lüdecke[1996].

[28] Between December 15, 1993 and June 14, 1995 the IBIS-DAX was only calculated between 8:30 a.m. – 10:30 a.m. and 1:45 p.m. and 5:00 p.m..

[29] The VDAX is a daily calculated DAX–based implied volatility index, which serves as a proxy for the expected stock market volatility. For technical details see DBAG[1995a]. Furthermore we used some other volatility measures — historical 30/90 day volatility and different implied volatility measures based on trading and calendar days as well as a mixed model. The results show that the different volatility measures do not affect the approximation results strongly. Therefore we restrict our presentation on results obtained for the VDAX.

trades in puts take place. Table 2 contains a short descriptive statistic[30] of the initial option data.[31]

Table 2:
Descriptive Statistic of the Trading Data

		S	X	$\frac{S}{X}$	T	r	σ^i	C
Calls	Mean	2141.25	2171.16	0.987	46	0.0446	0.130	36.90
	Std.	98.88	114.89	0.036	50	0.0036	0.031	43.39
	Min	1893.08	1800.00	0.825	3	0.0368	0.005	0.10
	P_1	1917.53	1900.00	0.896	3	0.0386	0.060	0.60
	P_{99}	2308.17	2450.00	1.101	238	0.0522	0.199	221.10
	Max	2373.18	2550.00	1.284	280	0.0540	1.640	844.40
		S	X	$\frac{S}{X}$	T	r	σ^i	P
Puts	Mean	2132.81	2097.62	1.018	47	0.0447	0.149	29.85
	Std.	99.78	117.53	0.039	49	0.0037	0.047	32.74
	Min	1893.08	1800.00	0.838	3	0.0368	0.006	0.10
	P_1	1915.16	1825.00	0.927	3	0.0385	0.068	0.50
	P_{99}	2304.72	2350.00	1.136	237	0.0521	0.237	162.00
	Max	2373.18	2550.00	1.284	280	0.0540	11.330	2111.50

However not all the data was used. But in contrast to other studies we do not exclude options which were difficult to handle, to get a truth view of the total pricing mechanism of the market and not only of a easy to handle part of it. For example if there exists a tendency towards trades on integer prices, which is crucial point for options with low prices, we feel that a neural network should be able to handle it. Hence we excluded only data which we assumed to be errorneous[32]: Options violating arbitrage boundaries[33] and option prices resulting from mistrades.

To get a practicable criterium we excluded all options with an implied volatility higher than 40 percent[34], which we assumed to be a indication of a mistrade.[35] Only 762 trades[36] were excluded due to this criterium. Furthermore 11,257 call prices and 2,092 put prices violating their lower arbitrage boundary[37] were excluded. Hence 97.3 percent of all trading

[30] T_t denotes the number of trading days till maturity, σ^i the implied volatility and P_α the α-percent percentile.

[31] This is important to know for the standardization of the input parameters of the networks and the restriction of the parameters for the interpolation of the B&S formula.

[32] Looking at table 2, examples for such data are easy to detect. Implied volatilities higher than 300 percent and option prices higher than DM 2,000 are a obvious sign for a mistrade.

[33] See (1).

[34] In table 3 the distribution of the implied volatilities is presented. Note: The VDAX as well as the historical 30-day and 90-day volatilities were below 35 percent in the whole period.

[35] Looking more closely at such trades it is obvious that they could be traced back to typos.

[36] 375 call and 387 put prices.

[37] Note: Transaction costs are not taken into account.

prices were used in our study. Looking at table 2 and table 3 it is interesting to note that

Table 3:
Implied Volatility

σ^i	Calls	Puts
[0–0.1)	21,481	12,768
[0.1–0.2)	231,041	225,611
[0.2–0.3)	1,748	6,430
[0.3–0.4)	398	604
[0.4–0.5)	187	214
[0.5–0.6)	80	90
[0.6–)	108	83

implied volatilities of call and puts differ notable[38]. Furthermore the number of arbitrage boundary violations differs strongly, whereas the number of mistrades is nearly the same for calls and puts in the period under consideration. These numbers give first evidence that the pricing formulas used by the market differ from the B&S formula.

3.2 Methodology

We first interpolated the Black & Scholes formula, using simulate option prices to get insights into the ability of the networks to approximate valuation formulas and their partial derivatives. We wanted to know if the choosen network typ is able to interpolate the B&S formula for realistic parameters[39] precisely enough[40].

Proceeding from our results we trained networks on trading prices of DAX options separately for calls and puts. The trained networks are used in out-of-sample test to prove if the networks are overparametrized. Therefore the option data is divided up into a trainings-sample to learn and conduct in-sample tests and a test-sample which serves for out-of-sample tests. The networks were trained within 10,000 iterations each using the Backpropagation algorithm.

We compared the implied pricing formulas with the B&S formulas. Examining the implied partial derivatives and implied state–price densities we are able to get deeper insights into the pricing mechanism and to find out the reasons for differing prices between these two models.

[38]This difference is significant at a 0.0001 level (Wilcoxon rank-sum test).

[39]We used randomized uniform distributed values within the relevant parameter range (see table 2). Because such simulated data are noise-free, out-of-sample tests do not make sense. We were primarily interested to get a feeling how adding more hidden nodes increases the accuracy of the interpolation and how many learning steps are necessary.

[40]In contradiction to some ealier studies, we trained our networks directly on option prices O and not on the function O/X which would assume a return distribution of the underlying of the option which does not depend on the stock price level (see Merton[1973] theorem 9 and Hutchinson/Lo/Poggio[1994], p. 862). This is a type of assumption we just wished to avoid by using nonparametric valuation methods.

3.2.1 Neural Network

Looking from a statisticans point of view the used neural networks are analogous to nonlinear nonparametric regression models. Due to their inductive nature they are able to infer complex nonlinear relationships between option prices and its determinants. More precisely, they have the ability to approximate any continuous function and its partial derivatives to any degree of accuracy.[41]

The primary goal of our article was to approximate the implied pricing mechanism of the market and to get deeper insights into the implied pricing formula. We did not want to compare the suitability of different network types[42], learning algorithms or methods to find the best network structure[43] for our particular problem.

Hence the results presented were obtained using standard networks. The network of our choice was a multilayer perceptron (MLP) with five inputs units, one hidden layer including five up to eleven neurons and one output neuron. This has got three reasons: First of all, the universal approximation property of MLP's for most classes of linear and nonlinear functions[44], second its proven ability to estimate simultaneously the unknown derivatives of the output function[45] which is essential to get deeper reliable insights into the pricing process of the market. Third, the possibility to compare our results to the results of some earlier studies. As input parameters we used:

- the price of the underlying (S),
- the strike price of the option (X),
- time to maturity (T),
- the riskless interest rate (r) and
- the expected volatility (σ)

Our network is fully connected and has the structure shown in figure 1. Such a MLP has the following functional form:

$$net(\vec{\mathrm{I}}) = f\left[\sum_{j=1}^{J} \omega_2(j,1) f\left[\sum_{i=1}^{5} \omega_1(i,j)\vec{\mathrm{I}}(i) + \theta_1(j)\right] + \theta_2(1)\right] \tag{6}$$

whereas

f	:	smooth monotonically increasing transfer function,
$\omega_1(i,j)$	:	weight between unit i of input layer and unit j of hidden layer,
$\omega_2(j)$	:	weight between unit j of hidden layer and the output unit,
$\vec{\mathrm{I}}(i)$	:	parameter i of input vector $\vec{\mathrm{I}}$,
$\theta_1(j)$	:	bias of unit j of hidden layer and
$\theta_2(k)$	:	bias of the output unit.

[41] Proofs of the universal approximation ability of MLP's are presented in Hornik/Stinchcombe/White[1989] and Hornik/Stinchcombe/White[1990].

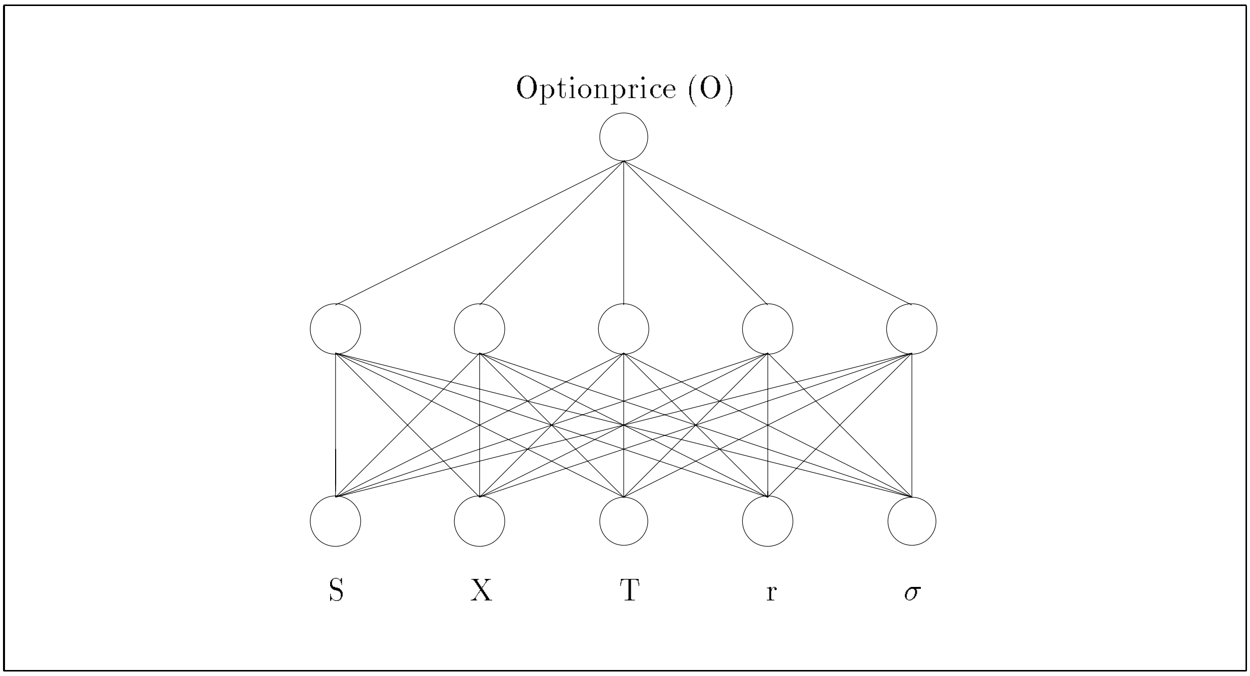
[42] Hutchinson/Lo/Poggio[1994] used three different network types and could not observe any type dominating the others.

[43] For example Anders/Korn/Schmitt[1996] used a model selection strategy to to get some insights into the statistical significance of the input fed into the network.

[44] Hornik/Stinchcombe/White[1989].

[45] Hornik/Stinchcombe/White[1990].

Figure 1:
Structure of the Neural Network



We used a logistic transfer function[46] because there exists a rather simple relationship between the function and its derivatives[47]. To simplify the notation the following functions are introduced:

$$\Omega_1(j) = \sum_{i=1}^{L_1} \omega_1(i,j)\vec{\mathrm{I}}(i) + \theta_1(j) \tag{7}$$

$$\Omega_2 = \sum_{j=1}^{L_2} \omega_2(j) f\Big[\Omega_1(j)\Big] + \theta_2 \tag{8}$$

Using (7) and (8), the first and second partial derivatives of our neural network with respect to the i-th input parameter have the following form:[48]

$$\frac{\partial net(\vec{\mathrm{I}})}{\partial \vec{\mathrm{I}}(i)} = \frac{\partial f[\Omega_2]}{\partial \Omega_2}\Big(\sum_{j=1}^{L_2} \omega_1(i,j)\omega_2(j)\frac{\partial f[\Omega_1]}{\partial \Omega_1}\Big) \tag{9}$$

$$\begin{aligned} \frac{\partial^2 net(\vec{\mathrm{I}})}{\partial \vec{\mathrm{I}}(i)^2} &= \frac{\partial^2 f[\Omega_2]}{\partial (\Omega_2)^2}\Big(\sum_{j=1}^{L_2} \omega_1(i,j)\omega_2(j)\frac{\partial f[\Omega_1]}{\partial \Omega_1}\Big)^2 \\ &+ \frac{\partial f[\Omega_2]}{\partial \Omega_2}\Big(\sum_{j=1}^{L_2} \omega_1(i,j)^2\omega_2(j)\frac{\partial^2 f[\Omega_1]}{\partial (\Omega_1)^2}\Big) \end{aligned} \tag{10}$$

To learn the implied pricing function and its derivatives the networks are "trained" on market prices. Each training set consisted of examples with five input parameters

[46] $f(x) = \frac{1}{1+\exp(-x)}$. Friedman[1994] confirms that this is the most popular choice and remarks that the particular choice of the transfer function is seldom crucial.

[47] $\frac{\partial f(x)}{\partial x} = f(x)(1-f(x))$ and $\frac{\partial^2 f(x)}{\partial x^2} = \frac{\partial f(x)}{\partial x}(1-2f(x))$.

[48] The derivation of the formulas and the adjustements necessary for standardized data are presented in Appendix B.

$\vec{\mathrm{I}} = (X, S, \sigma, t, r)$ and the corresponding option price O – either the theoretical B&S value or the trading price.[49] We do not use any statistical inference method to test the statistical significance of the input parameters, because looking at the price bounds of options we know that there must be a functional relationship between the used input parameters[50] and the option prices.

To find the network parameters the following cost–function is minimized using the Backpropagation-Algorithm[51].

$$\min \sum_{i=1}^{N} (net^i(\vec{\mathrm{I}}) - O^i)^2 \tag{11}$$

"Training" and "learning" means nothing more then improving the estimates of the ω's and θ's with respect to this cost–function. To increase the effectiveness of the learning algorithm and minimize the effect of different dimensions of the input parameters we standardized the input parameters and the corresponding option price using the following formula

$$g(i) = \frac{i - \min(i)}{\max(i) - \min(i)}. \tag{12}$$

All connection weights were initially randomized and during the training process determined. We trained our networks in batch mode. To implement the networks we used the Stuttgart Neural Network Simulator (SNNS).[52]

3.2.2 Performance Measures

Commonly used measures[53] for the approximation quality of neural networks are the mean error (ME), the mean percentage error (MPE), the residual mean squared error (RMSE), the mean absolute error (MAE), the mean absolute percentage error (MAPE) and R^2

[49] Besides we trained networks on the time value of options and furthermore only on that part of the time-value which differs from the corresponding B&S value. So we tried out a new stepwise approach. The idea was to reduce the complexity of the problem by putting the linear part of the evaluation formula out of consideration rsp. that part which is explained by the B&S formula (see Boeck et al.[1995]). But the results for DAX options were not very encouraging (see Appendix C) The best fit was obtained for option prices.

[50] This does not hold for σ, but looking at the asymmetric pay-off pattern of calls and puts it is easy to motivate a potential influence of the volatility of the underlying on option prices, too.

[51] The
algorithm is based on a simple gradient descend. For an introduction see Hertz/Krogh/Palmer[1991], p. 115ff.

[52] See Zell et al.[1995].

[53] Note: All performance measures used in our study are founded on the difference between the non-standardized option prices of the network and the observed market price or the theoretical B&S price! The efficiency of neural networks can be increased by standardizing input and target values. It is important to note that there is quite a difference between using the standardized values or the origin data to measure the performance of the networks. Reliable information is only provided by origin data because otherwise the error measured also depends on the method used to standardize the data. Using neural networks to evaluate derivatives the error of interest ist the difference between the estimated prices and the target prices. Using instead a function like O/X changes the error term considerable. Using O/X instead of O leads in the case of the DAX option to ME's, MSE's and RMSE's which are about 2000 times lower than the real error of interest.

defined as:

$$
\begin{array}{lcc}
\hline
\text{ME} & = & \frac{1}{T}\sum_{t=1}^{T}(\hat{O}_t - O_t) \\
\text{MPE} & = & \frac{1}{T}\sum_{t=1}^{T}\frac{\hat{O}_t - O_t}{O_t} \\
\text{MAE} & = & \frac{1}{T}\sum_{t=1}^{T}|\hat{O}_t - O_t| \\
\text{MAPE} & = & \frac{1}{T}\sum_{t=1}^{T}\frac{|\hat{O}_t - O_t|}{O_t} \\
\text{RMSE} & = & \sqrt{\frac{1}{T}\sum_{t=1}^{T}(\hat{O}_t - O_t)^2} \\
R^2 & = & \frac{\sum_{t=1}^{T}(\hat{O}_t - \overline{O})^2}{\sum_{t=1}^{T}(O_t - \overline{O})^2} \\
\hline
\end{array}
\tag{13}
$$

ME, MAE and RMSE measure the absolute difference between the output and the target value while MPE and MAPE standardize this error. R^2 measures the closeness of variation between the output function of network and target values. ME and the MPE are able to detect a pricing bias whereas MAE, RMSE, MAPE measure the dispersion of the output values around the target values.

4 Empirical Results

4.1 Learning the Black & Scholes Formular

We trained the networks on theoretical B&S values on an uniform distributed parameter region separately for calls and puts. Table 4 summarizes the interpolation results of networks with five and eleven nodes.

Table 4:
Results B&S Interpolation
Option-Prices

Nodes	Calls				Puts			
	ME	MAE	RMSE	R^2	ME	MAE	RMSE	R^2
5	0.458	3.484	4.93	0.998	-0.012	2.469	3.66	0.999
11	-0.051	1.340	2.02	0.999	-0.028	1.051	1.62	0.999

The average mean error of a MLP with eleven nodes is DM -0.051 for Calls and DM -0.028 for Puts. The MAE is about DM 1.00 for both options and reveals that no bias is observable. Increasing the number of nodes from 5 to 11 more than halves the RMSE. This is also visible in figure 2. It is interesting to note that the pricing error is already the same for the whole parameter range and puts are better interpolated.

The improved results for networks with 11 nodes are also reflected by the partial derivatives. Looking at table 5 we can observe similar results for put and call options. For networks with 11 nodes the error terms are clearly lower. For all first derivatives the mean error of such networks is only about 1 percent of its average value. Looking at rho the results visualized in figure 3 are even more impressive.[54] Comparable results are

[54]The results presented are obtained for options with a strike price of 2100 furthermore assuming a volatility of 0.125 and interest rates of 4.5 percent.

Table 5:
Results B&S Interpolation
Partial Derivatives

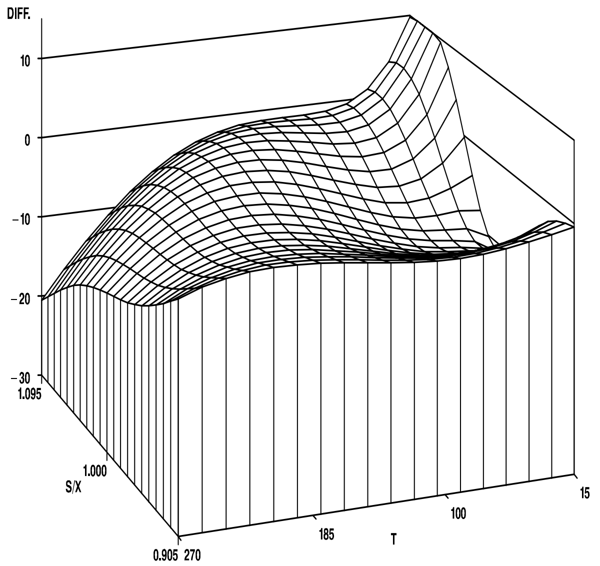
Derivative	Nodes	Calls				Puts			
		ME	MAE	RMSE	R^2	ME	MAE	RMSE	R^2
Delta	5	-0.00067	0.042	0.065	0.963	0.00370	0.027	0.048	0.980
	11	-0.00151	0.018	0.031	0.992	0.00107	0.013	0.026	0.994
Vega	5	15.29	56.37	80.47	0.854	4.36	43.18	62.40	0.906
	11	3.16	29.31	44.69	0.954	0.81	22.91	34.98	0.971
Theta	5	0.0212	0.079	0.154	0.617	-0.0019	0.061	0.140	0.669
	11	0.0072	0.043	0.118	0.763	0.0020	0.035	0.110	0.794
Rho	5	10.12	85.45	124.06	0.820	5.05	75.56	114.51	0.874
	11	5.50	30.69	44.98	0.975	-4.18	35.16	35.16	0.988
Gamma	5	-0.00002	0.0012	0.0016	0.566	0.00012	0.0012	0.0021	0.581
	11	-0.00043	0.0008	0.0012	0.894	-0.00022	0.0009	0.0013	0.885
SPD	5	-0.00042	0.0016	0.0020	0.243	-0.00039	0.0014	0.0026	0.252
	11	-0.00084	0.0011	0.0015	0.492	-0.00074	0.0011	0.0015	0.531

observed for the other derivatives, too. Especially in a parameter region for S/X between 0.9 and 1.1 the derivatives are interpolated very well by MLP's with 11 nodes.

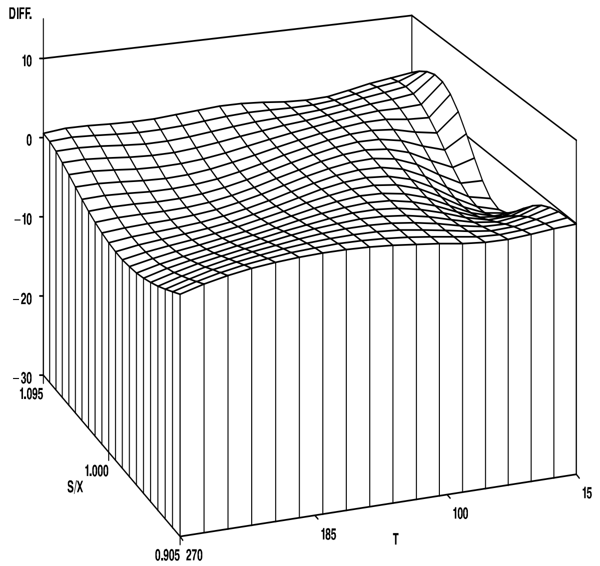
Figure 2:
B&S Interpolation-Error

Calls

5 Nodes

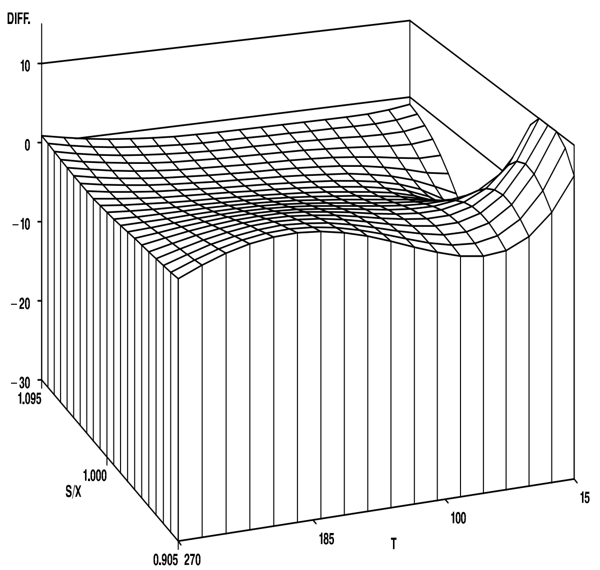


11 Nodes



Puts

5 Nodes



11 Nodes

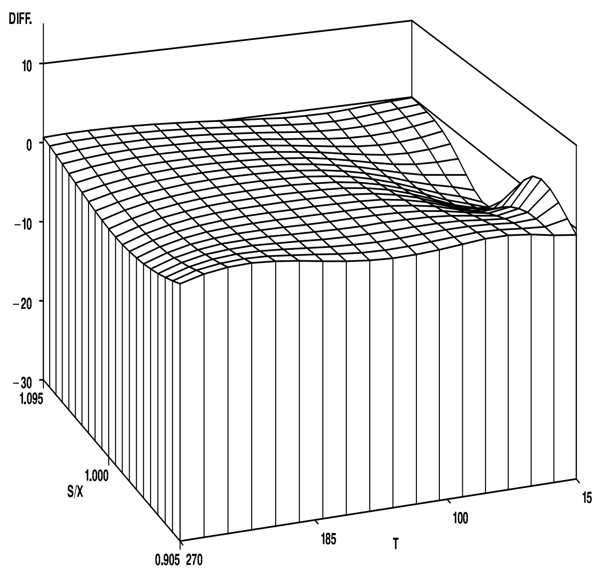
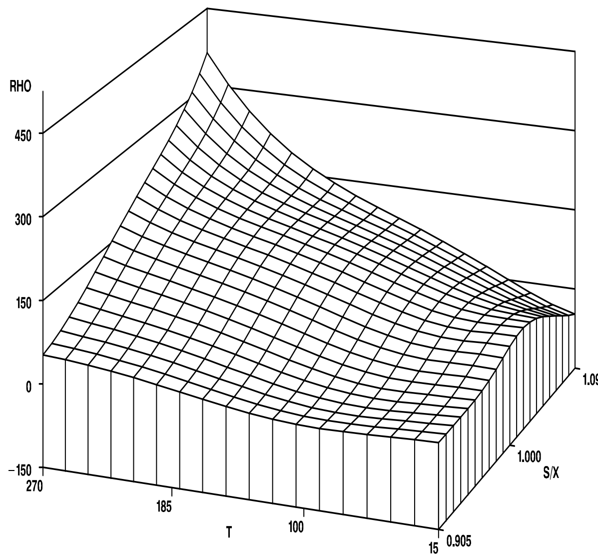


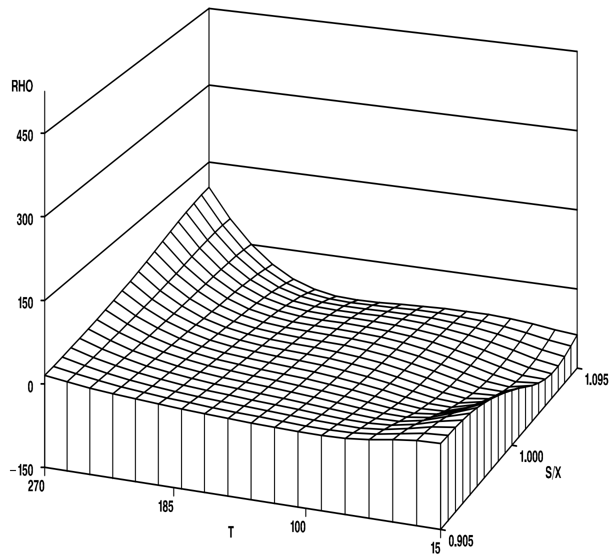
Figure 3:
B&S Interpolation Rho-Error

Calls

5 Nodes

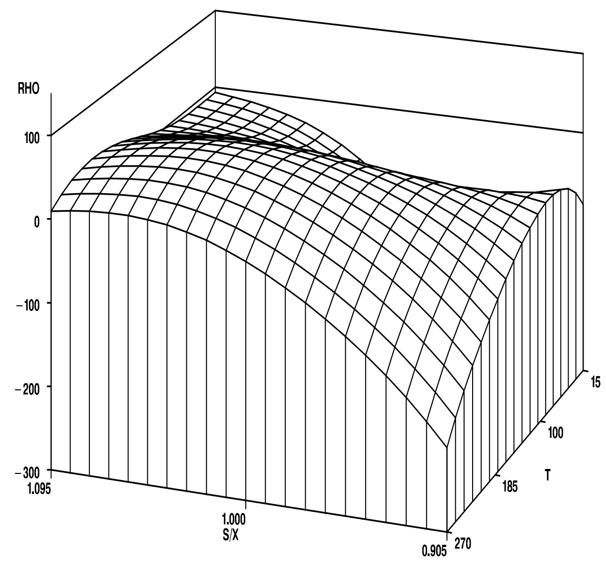


11 Nodes

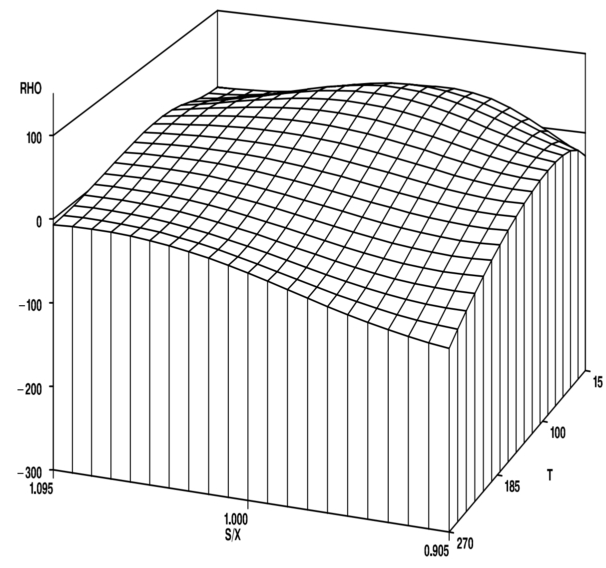


Puts

5 Nodes



11 Nodes



4.2 The Market Formulars

4.2.1 The Implied Valuation Formulars

Assuming a market pricing formula with a complexity similar to the B&S model the used networks should be able to approximate the pricing formula and its derivatives sufficienly well. Looking at the error terms this can be confirmed (see table 6).

Table 6 compares the ability to "explain" market prices between our ANN's and the Black & Scholes formula. Before training the network we divided up the trading prices into two equal subsamples. The first subsample (denoted as in) we used to train our networks, the second (denoted as out) served for testing the pricing formular.

The table shows that the networks are able to explain the market prices far better than the Black & Scholes model.[55] This applies for in–sample tests as well as out–of–sample tests. The results of the out–of–sample tests are sometimes even slightly better than the in–sample results. This shows that there are no signs of overfitting even for networks with eleven nodes.[56]

The average error of a network with eleven nodes is about 15 Pfennige for calls compared to nearly 3.00 DEM for the Black & Scholes formula and about 4 Pfennige for puts compared to 66 Pfennige in the case of the Black & Scholes formula. It is interesting to note that the networks as well as the Black & Scholes formula are better in explaining put prices than call prices.

The implied pricing function of calls (figure 4) is an increasing function of moneyness and time to expiration but shows higher prices[57] compared to the B&S model on nearly the whole parameter range. Only prices of in-the-money calls are below the B&S prices. Strongest deviations are observable for out-of the money option with a long time to maturity. This may be explained by the crash–o–phobia phenomenon shifting the state–price–densities of states near the money to out–of–the–money states. We will come back to this point later.

The price deviation is a monotonous decreasing function of time to maturity (T) and an increasing function of moneyness (S/X). The same is valid for the percentage price difference (figure 4). The strong increase for short term out-of the money options may be explained by the very low theoretical B&S prices and due to the fact that the networks are able to learn the minimum tick size which

[55] Restricting our out-of-sample tests to options meeting the following often used conditions:

- $0.85 \leq \frac{O}{X} \leq 1.15$.
- Maturities $>$ 10 days.
- O $>$ DM 10.

improves the results furthermore. For example for 11 nodes for calls the ME reduces to -0.015, the MAPE to 0.080 and the MPE to -0.001, whereas for puts the ME reduces to 0.116, the MAPE to 0.059 and the MPE to 0.003. This is interesting to note for the interpretation of the implied pricing functions.

[56] The different results compared to the study of Anders/Korn/Schmitt[1996] can be explained by three factors. First by the larger data samples, Ssecond by the inclusion of options with short time to maturity as well as deep-in and deep-out-of-the-money options which are difficult to handle and third by the different target values.

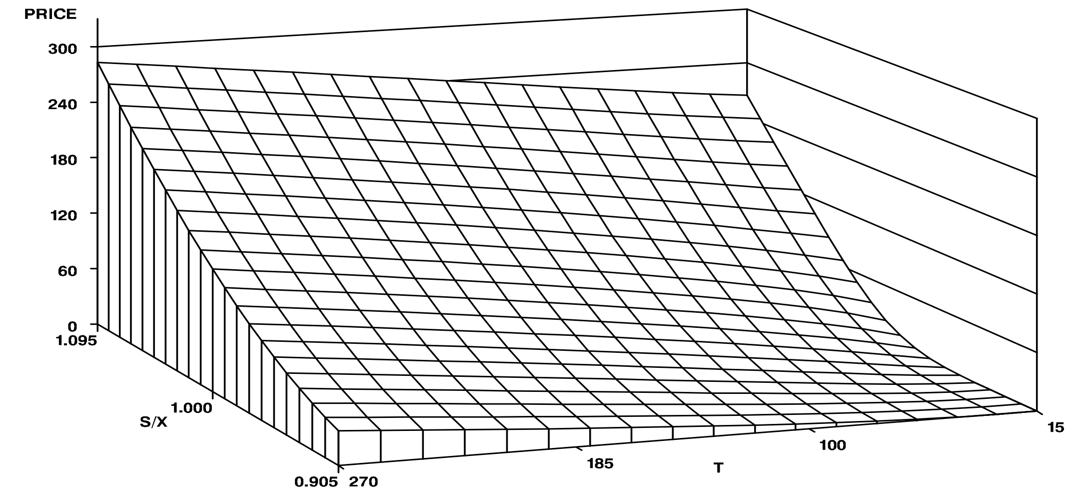
[57] Recognizable by negative differences in the of figure 4.

Table 6:
Results Market Price Approximation
Option Prices 1995

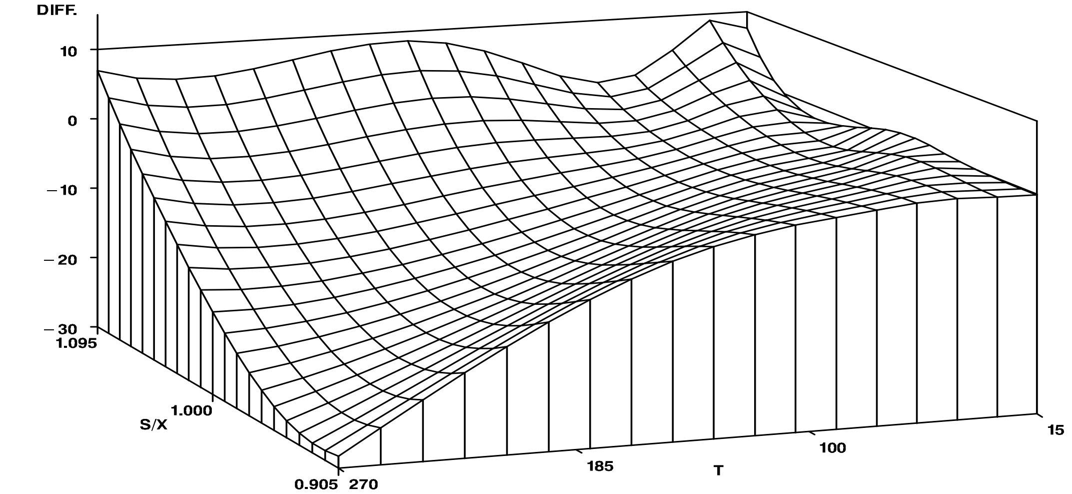
Nodes	Sample	Calls						Puts					
		ME	MAE	MAPE	MPE	RMSE	R^2	ME	MAE	MAPE	MPE	RMSE	R^2
5	in	-0.403	2.58	0.220	0.049	3.64	0.991	0.049	1.70	0.210	0.052	2.37	0.994
	out	-0.407	2.58	0.219	0.048	3.64	0.991	0.053	1.70	0.210	0.052	2.37	0.994
6	in	-0.636	2.63	0.214	0.026	3.74	0.991	0.291	1.84	0.219	0.061	2.52	0.993
	out	-0.640	2.63	0.213	0.025	3.73	0.991	0.295	1.84	0.219	0.061	2.54	0.993
8	in	-0.226	2.47	0.210	0.038	3.47	0.992	0.028	1.75	0.204	0.045	2.42	0.994
	out	-0.228	2.47	0.209	0.037	3.47	0.992	0.030	1.74	0.204	0.047	2.43	0.994
11	in	-0.152	2.32	0.197	0.036	3.31	0.992	-0.035	1.72	0.205	0.044	2.39	0.994
	out	-0.150	2.32	0.196	0.036	3.29	0.992	-0.031	1.71	0.205	0.045	2.39	0.994
B&S	all	2.944	3.50	0.330	0.317	4.13	0.995	-0.655	2.85	0.264	0.029	3.96	0.984

Figure 4:
Call-Options

Implied Pricing-Function



Difference to B&S



Percentage Difference to B&S

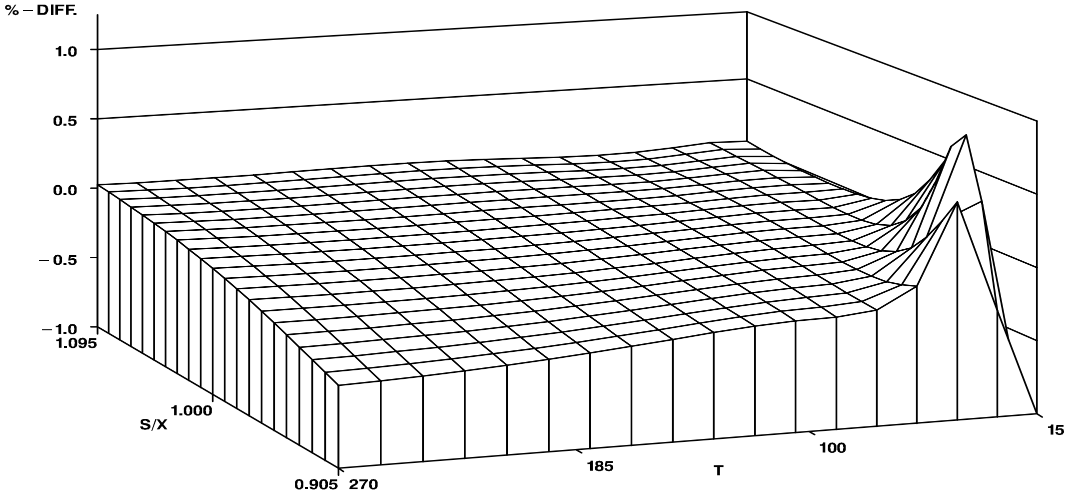
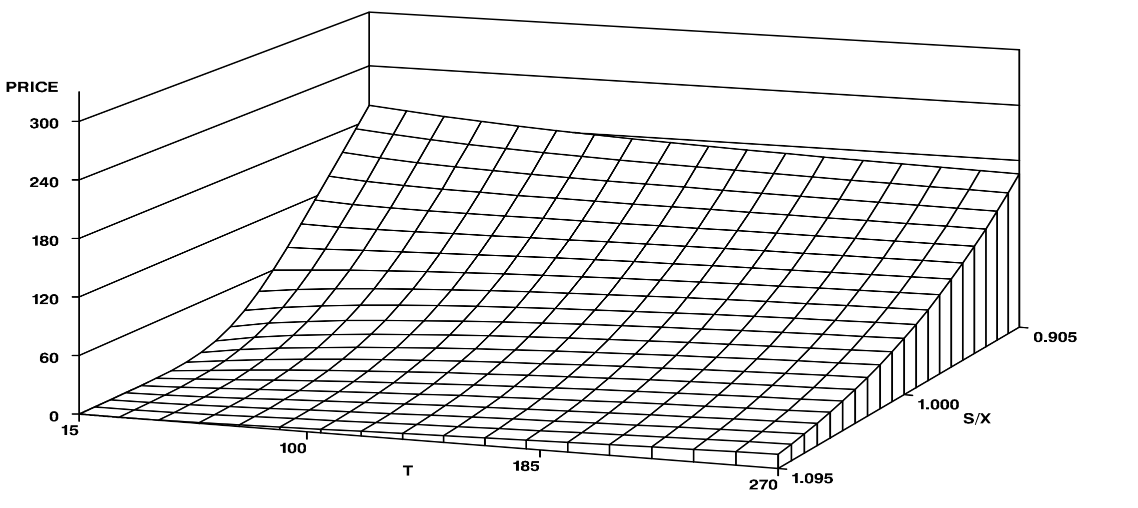
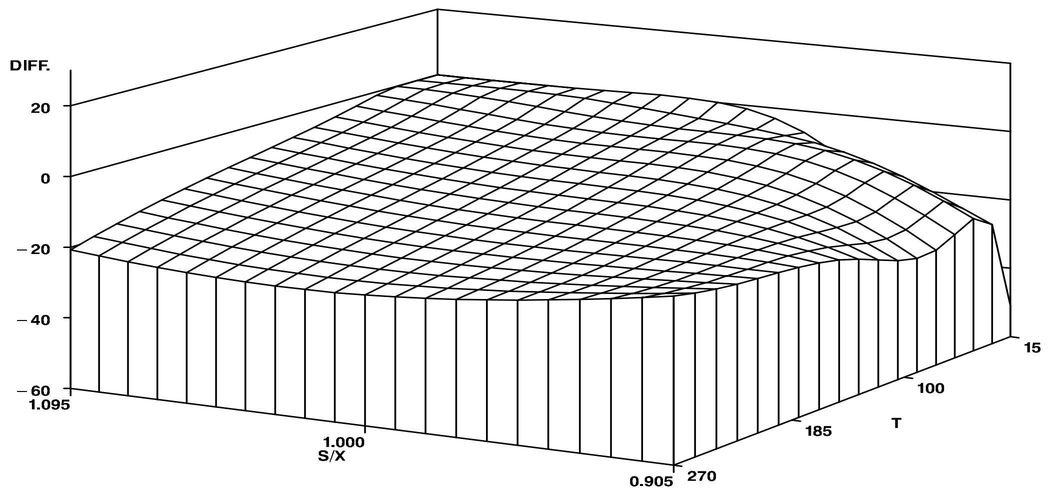


Figure 5:
Put-Options

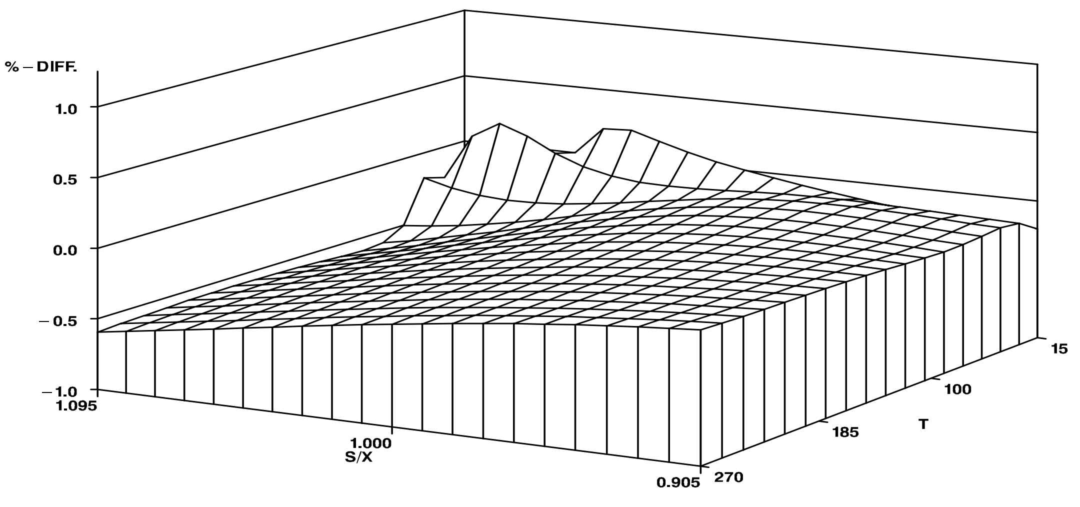
Implied Pricing-Function



Difference to B&S



Percentage Difference to B&S



results in a very high percentage price difference. The same effect is observable for out-of the money put options (lower graph on the right-hand side of figure 5).

The implied pricing function of puts (figure 5) is an increasing function of moneyness but not always an increasing function of time to expiration. In-the-money puts show a slightly negative relationship[58]. On the whole parameter range the implied pricing function provides higher prices. Short term in-the money options account for the highest deviation. The percentage error is an increasing function of moneyness and seems to be independent to time to maturity[59].

4.3 The Implied Partial Derivatives

Looking at the partial derivatives of the pricing formulas we are able to obtain insights into the pricing process and to search for the reasons of differing option prices. The results are presented for average parameter values[60].

Deltas of call options show a similar structure but are lower compared to the B&S deltas. This yields for gamma which is lower with the exception of short-term out-of-the money options, too.

Vega also has a similar shape but shows negative values for short term in-the-money options. The shape of theta is also very similar but seems to have a weaker impact on in-the-money options whereas the impact on out-of-the money options is stronger. Rho instead differs strongly from the B&S model. Only for short term in the money options their seems to be an impact at all. Within this region the impact is strongly negative.

The delta of puts has also the same shape as B&S delta but is also lower on the whole parameter range. Strongest deviations are observable for short-term in-the-money options. This is reflected by gamma, too. Gamma only shows for short-term in-the money markable deviations from the B&S gamma.

Vega also has a similar shape compared to the corresponding function of the B&S model but shows negative values for short term in-the-money options. The same is valid for theta. The market rho seems to have a weaker impact on put prices. The difference is a decreasing function of time to maturity and a decreasing function of moneyness.

To sum up, for both kinds of options delta has nearly the same shape as the corresponding function of the B&S model but is lower on nearly the whole parameter range. Vega is negative for in-the-money calls and in-the-money puts. Hence the market values an expected volatility increase negative for in-the money options. For in-the money options with a short time to maturity strong positiv deviations from the B&S model are observable. This yields also for market theta. The most interesting results are obtained for rho. The implied partial derivative for both types of options differ strongly from the B&S model. Rho is markable lower for calls (for short-term in-the-money options even negative) whereas rho for puts is less negative (except for short term in-the-money puts). On

[58] In this context it is interesting to mention that the payoff of a put option is bounded whereas the payoff of a call is unbounded.

[59] The increase for out-of-the-money options is due to the ability of the networks to learn the minimum ticksize.

[60] We used average parameter values for X, σ and r to ensure that the fit of the network is best. Visualized are the derivatives of calls and puts with a strike price of 2100. The volatility is assumed to be 0.125 (which corresponds to the annualized daily return volatility of the DAX in 1995) and the interest rate 4.5 percent (average interest rate in 1995 see table 2).

possible explanation of this effect is that the implied pricing formula takes into account the negative correlation between interest rates and the DAX .

Figure 6:
Delta — Call Options

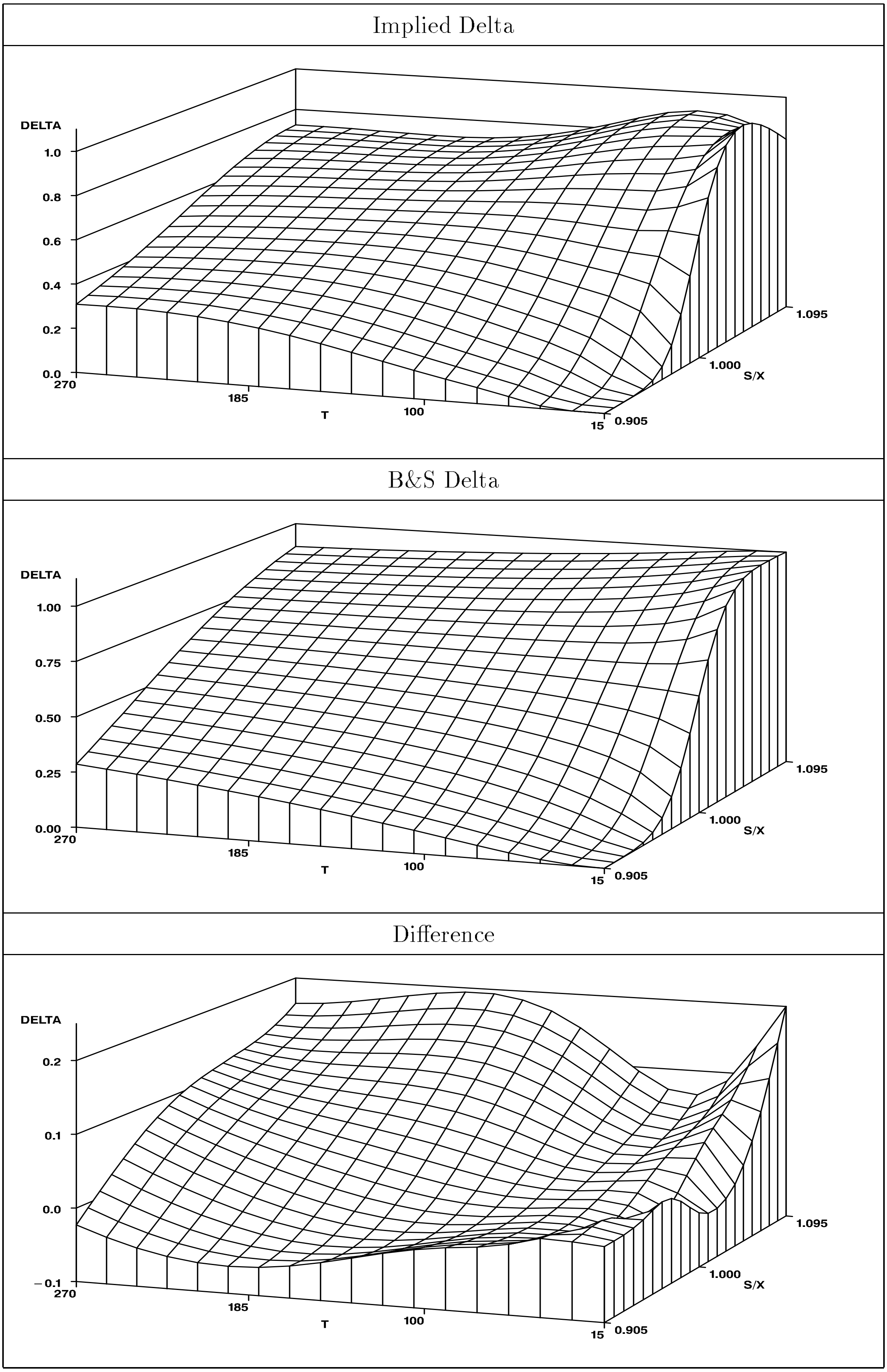


Figure 7:
Vega — Call Options

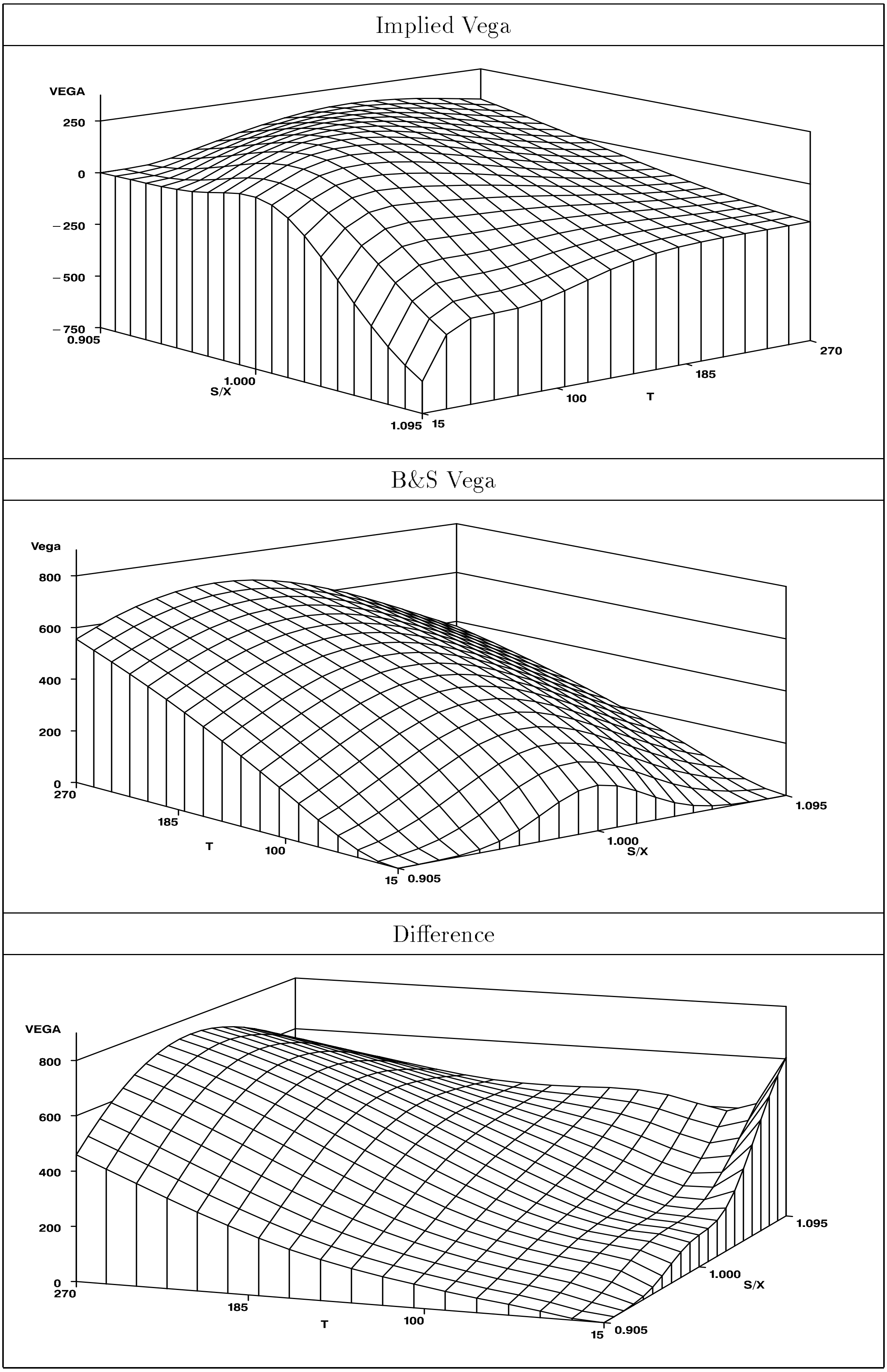
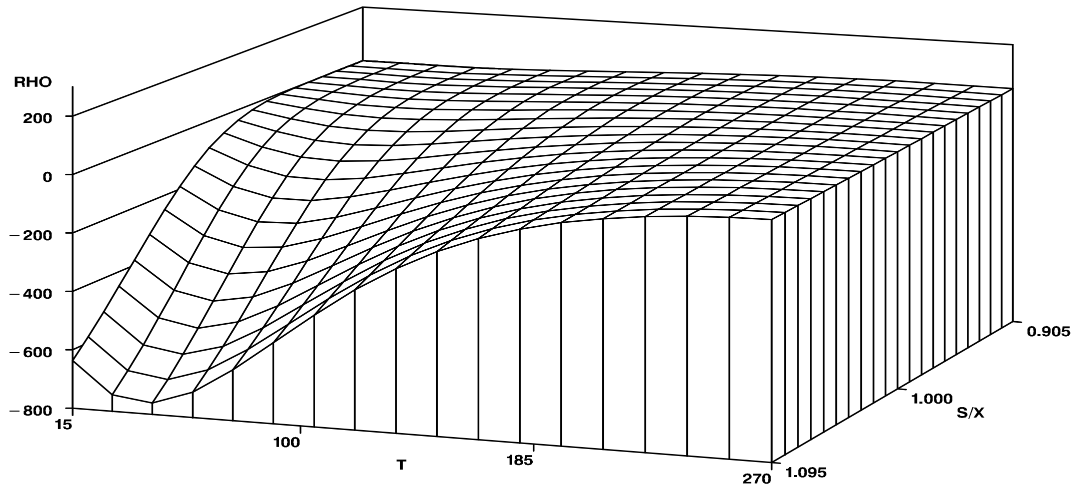
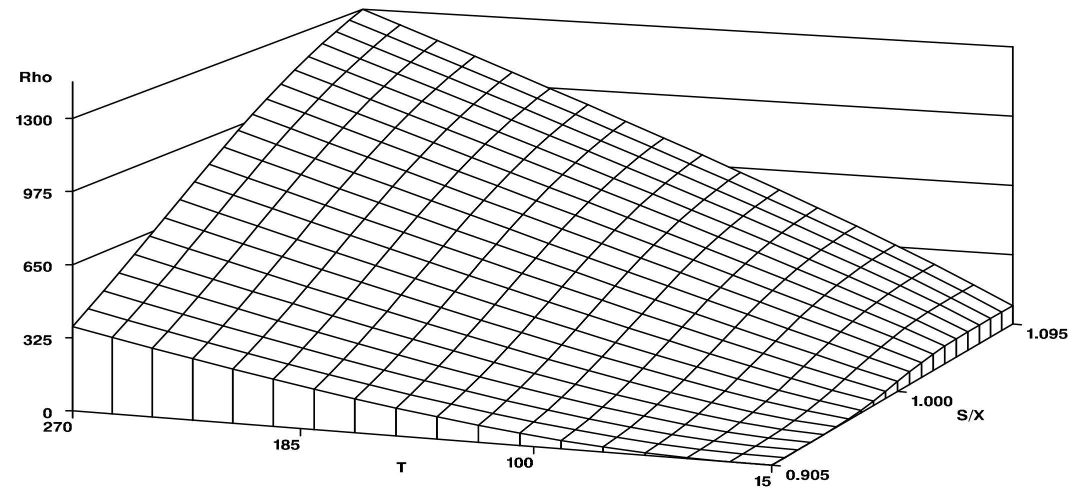


Figure 8:
Rho — Call Options

Implied Rho



B&S Rho



Difference

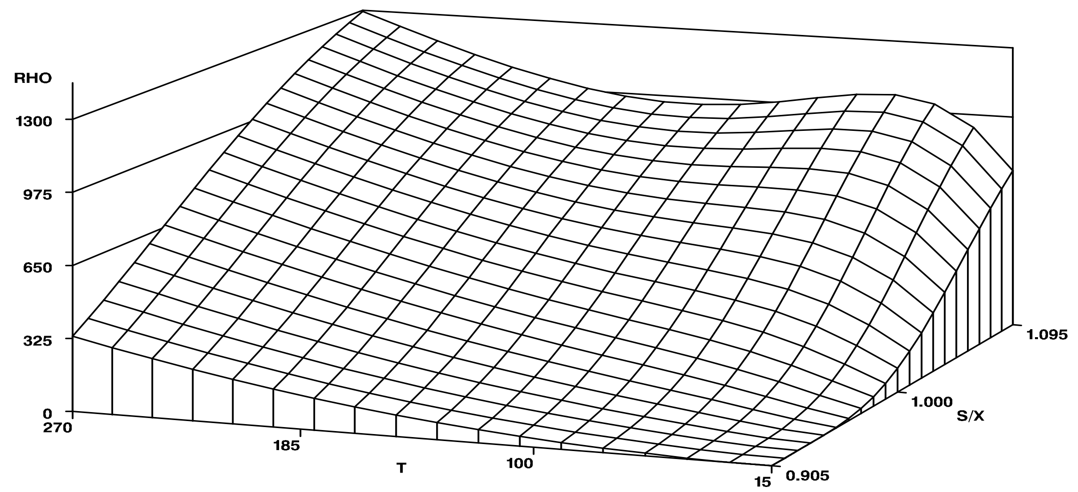


Figure 9:
Theta — Call Options

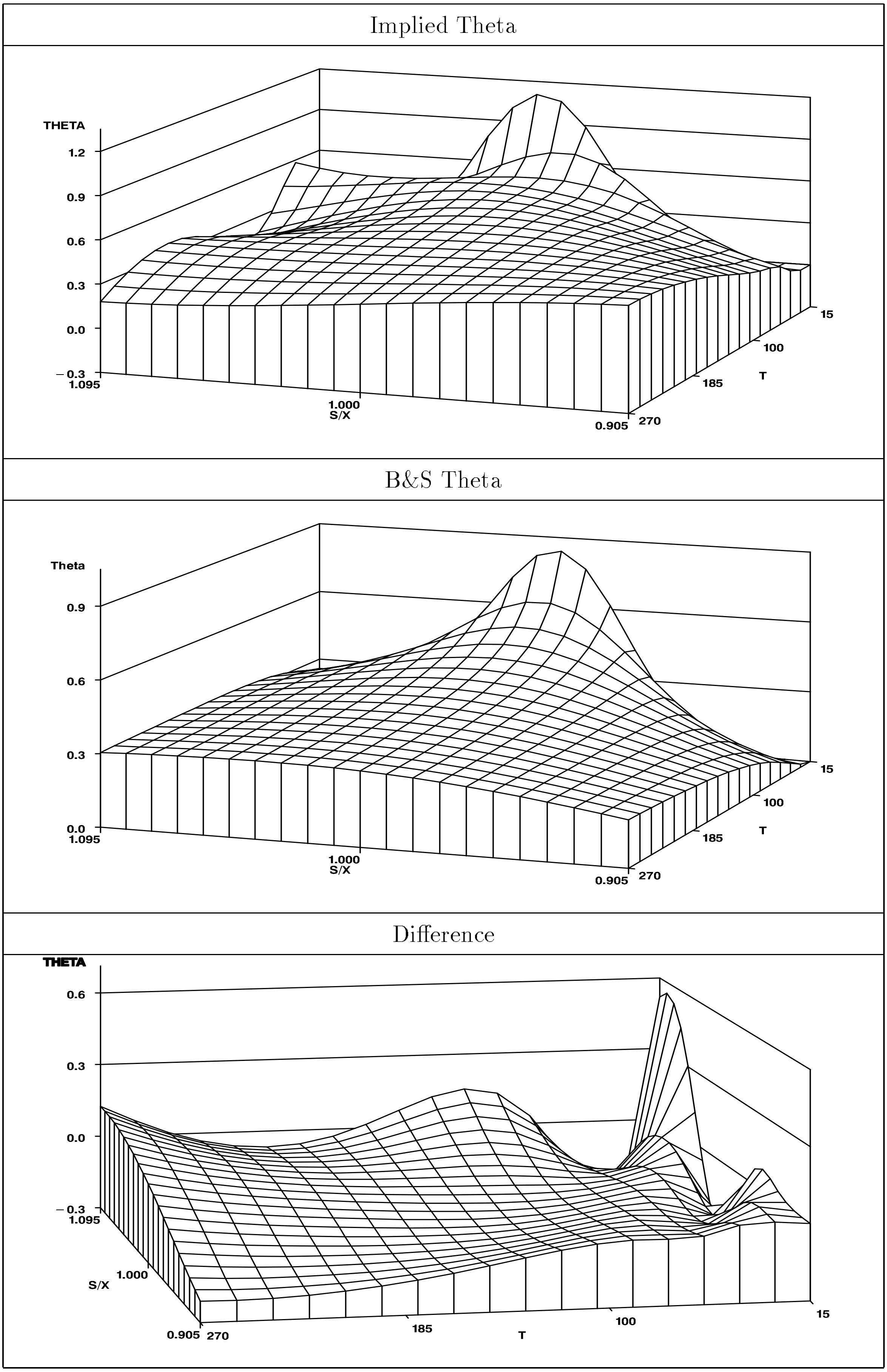
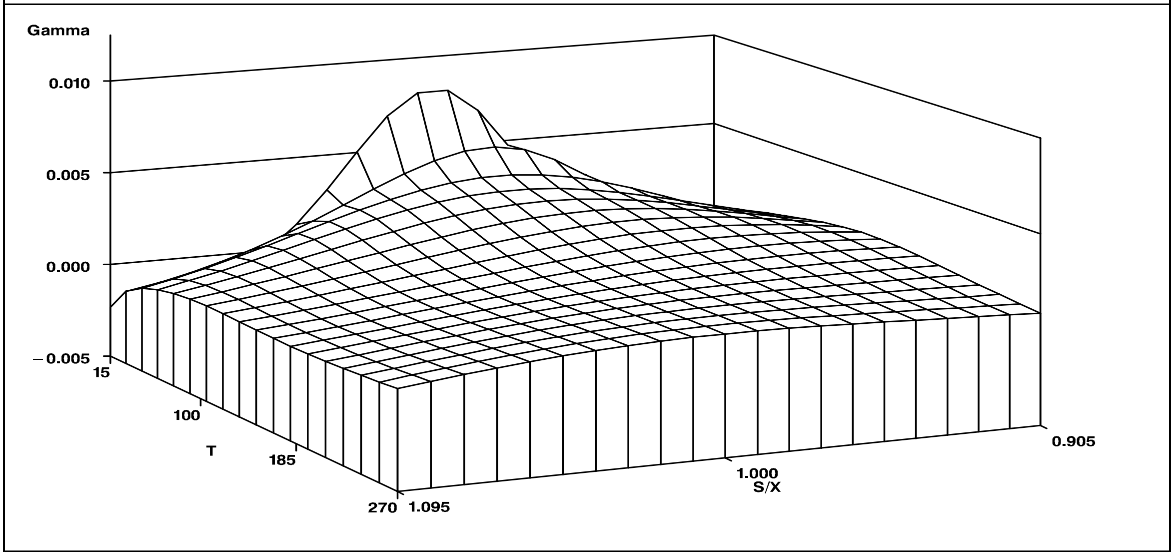
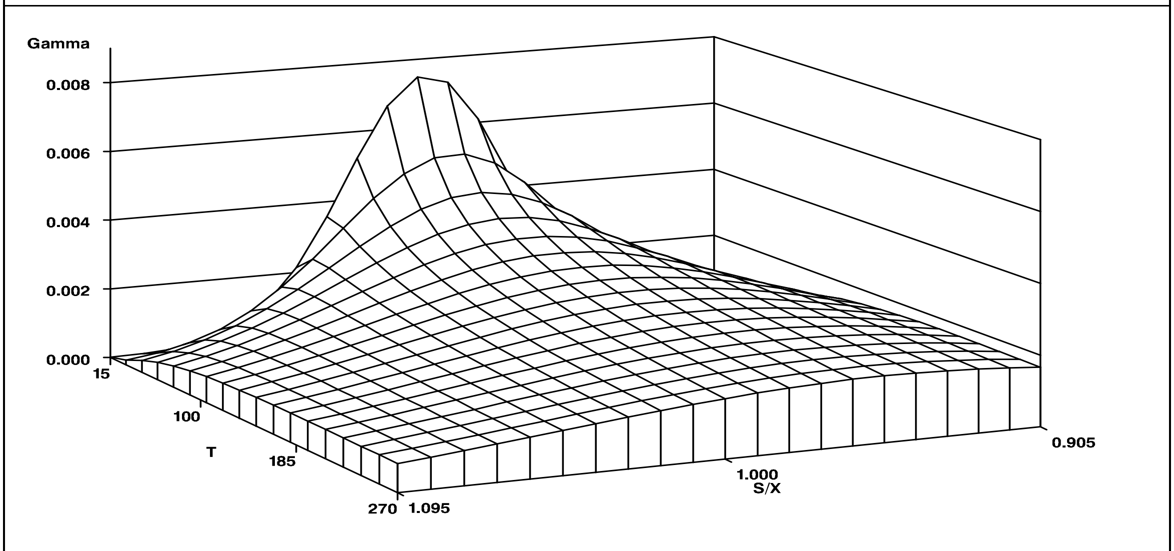


Figure 10:
Gamma — Call Options

Implied Gamma



B&S Gamma



Difference

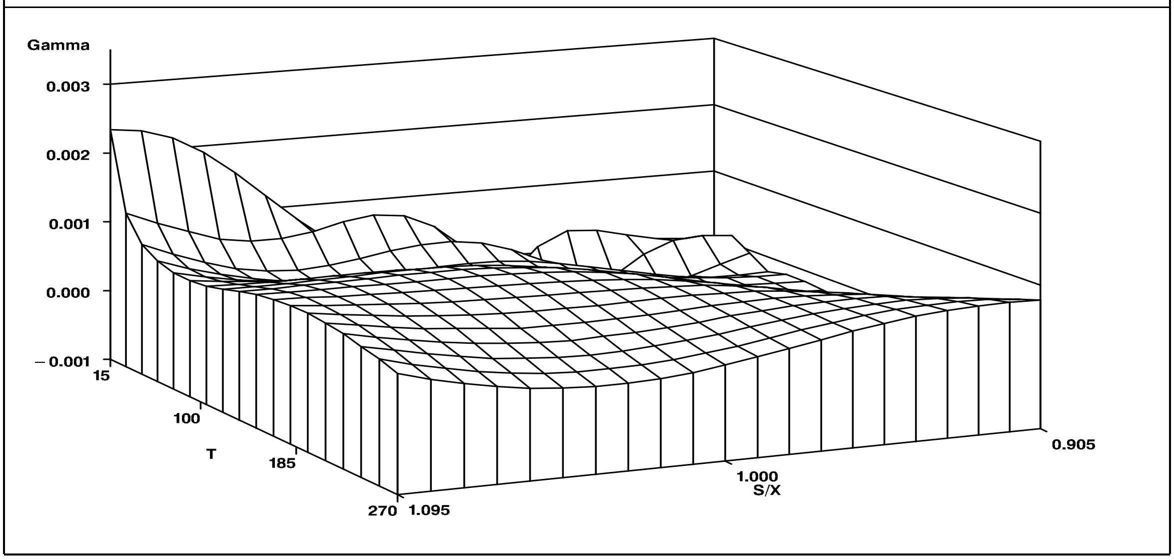


Figure 11:
Delta — Put Options

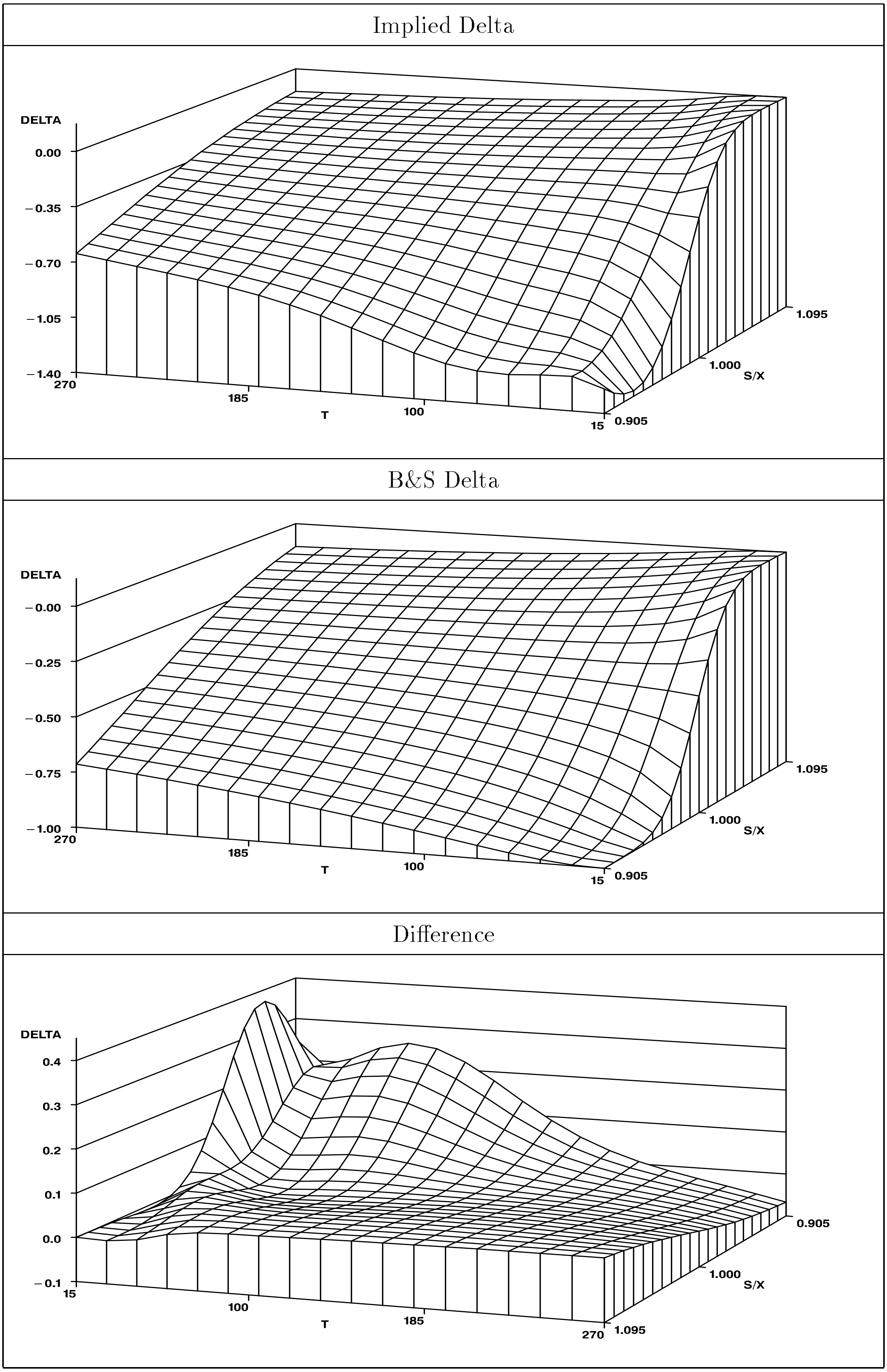
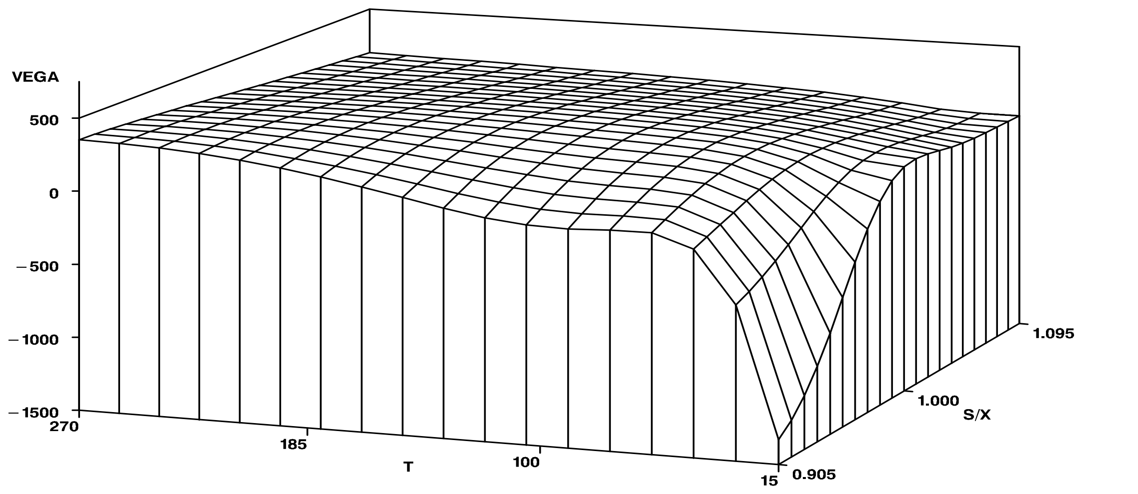
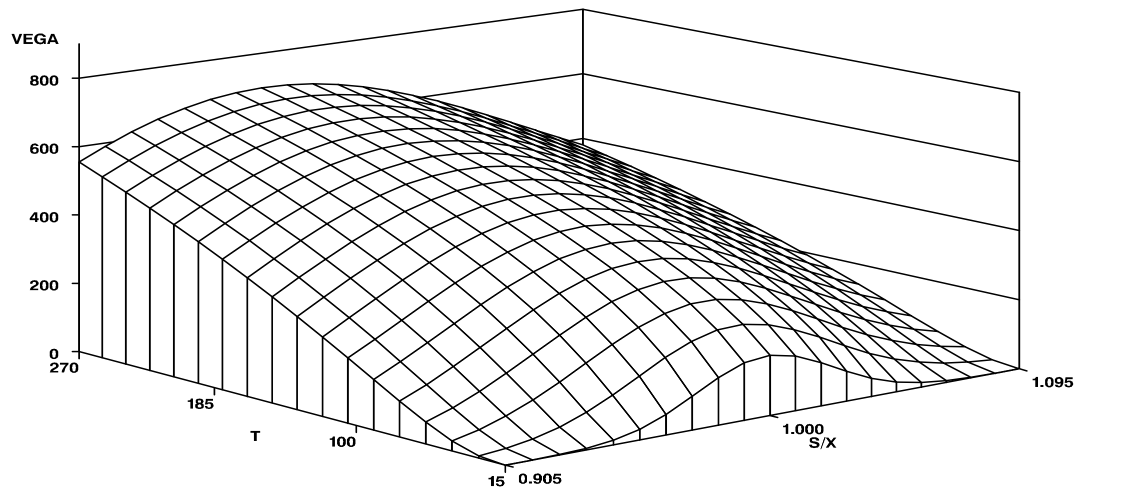


Figure 12:
Vega — Put Options

Implied Vega



B&S Vega



Difference

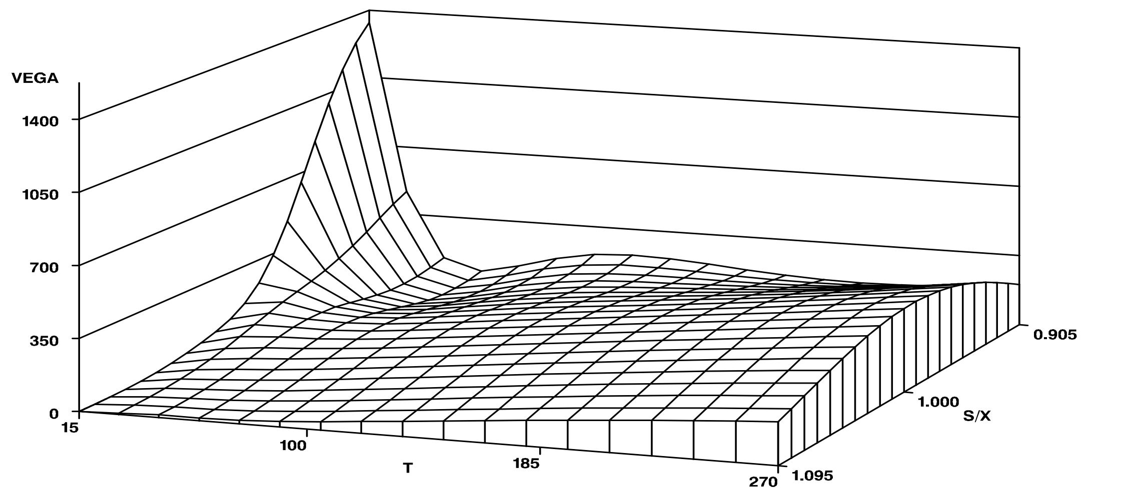
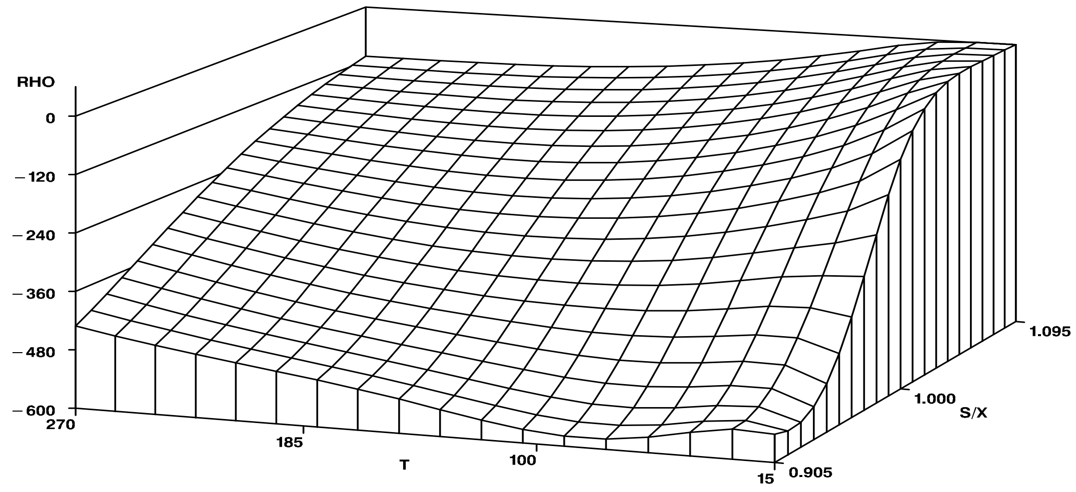
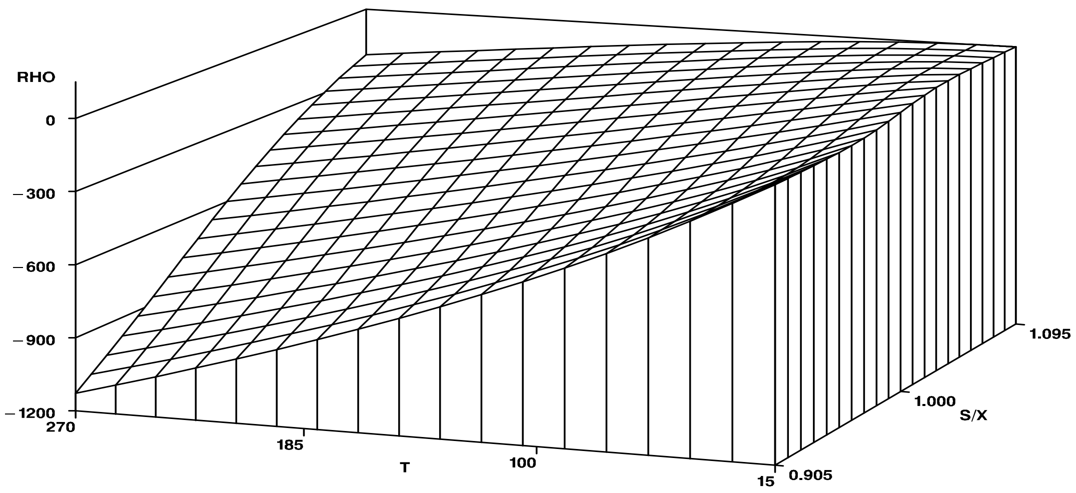


Figure 13:
Rho — Put Options

Implied Rho



B&S Rho



Difference

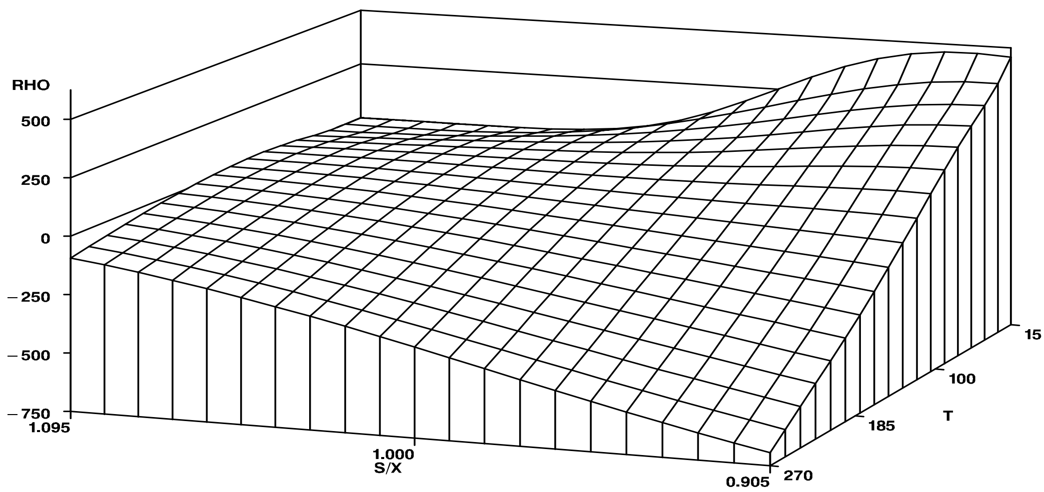


Figure 14:
Theta — Put Options

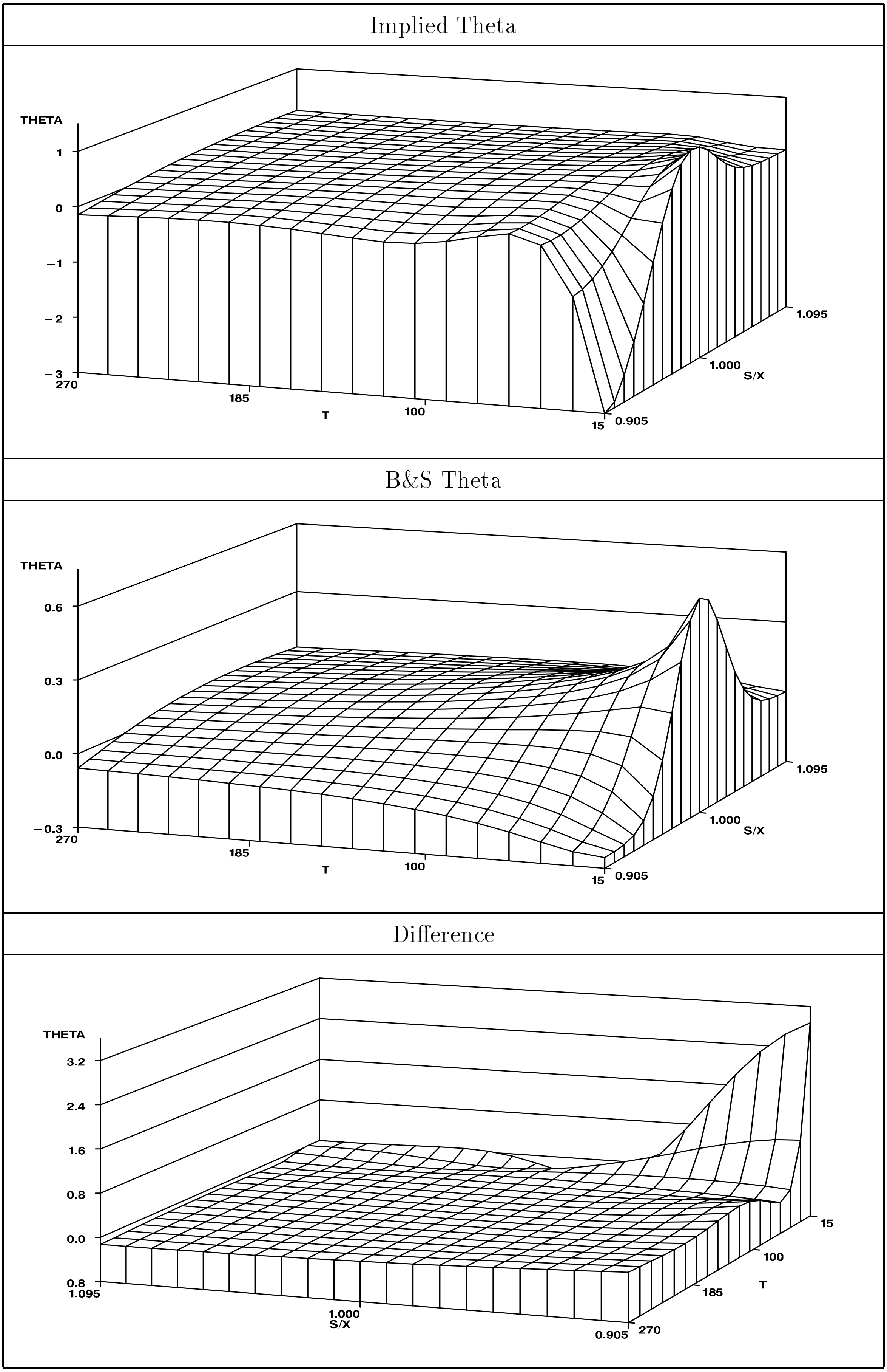
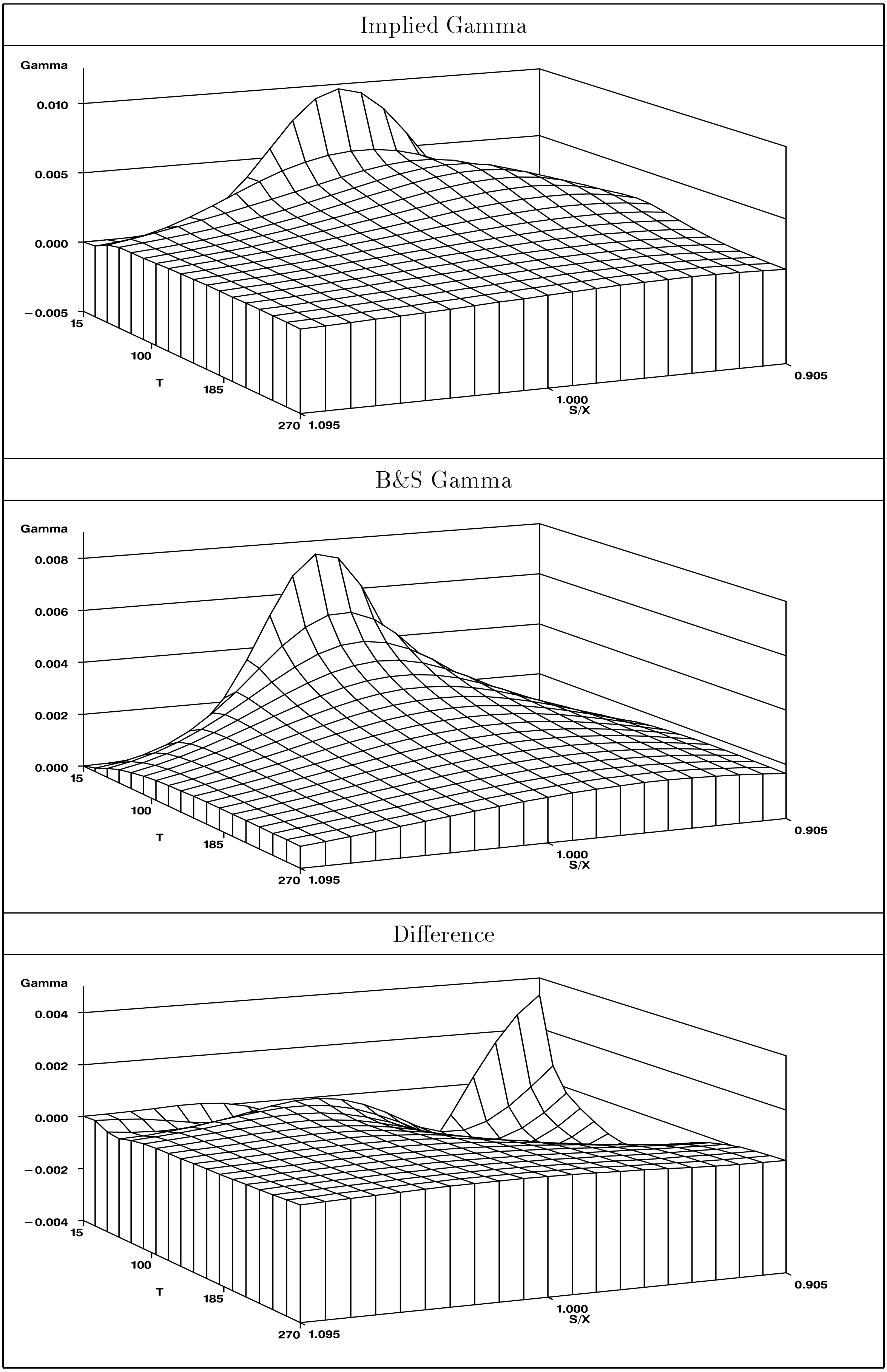


Figure 15:
Gamma — Put Options



4.4 Implied State Price Density

Assuming the existence of a SPD and the dynamic completeness of the markets we are able to derive the implied prices of "Arrow-Debreu" securities. It is important to note that:

- We did not put any restriction, like positivity restrictions or the parametric form of the SPD, on our network.
- Puts only contain information on states below their strike price and calls only on states exceeding their strike price. Hence the networks trained on puts can not contain any information about states above the highest strike price and networks trained on calls about states below the lowest strike price.[61].

The graphs in figure 16 show implied state–price–densities with respect to time to maturity and the future state measured as the future index level divided by the current index level. The value 1 describes the state that will occur if the index level at expiration corresponds to the current index level. The graphs show at least two things:

- First — within the trained parameter region all state prices were positive — not revealing any obvious arbitrage opportunity.
- Second — It is visible how the uncertainty about future states decreases with decreasing time to maturity. This is reflected in the increasing state–prices or in other words the more concentrated density.

That seems to demonstrate the rationality of the pricing mechanism of the market. Further interesting results can be obtained by comparing the implied state–price–density with the theoretical Black & Scholes values. To get a more clearly presentation in figure 18 the SPD's for different maturities are plotted separately. The dashed line corresponds to the theoretical Black & Scholes state prices. Figure 20 the shows the difference to the B&S state prices.

Looking at the used trading data we see that 99 percent of the call options have a moneyness above 0.90 and 99 percent of the puts a moneyness below 1.14.[62] This is clearly reflected by the state prices. Only state prices outside this region show negative values, which results — like mentioned before — out of the fact that the available put and call prices do not contain any information about this states.

The implied state price densities differ markable from the state–price–density of the B&S world (dashed graphs). Furthermore implied distributions of calls and puts differ for short-term options. Whereas the distribution of calls has a symmetric shape for short- and medium-term options (up to 90 days) the implied distributions of puts are shifted to the right. For long–term options the implied distributions of calls are also shifted to the right. What means that the implied function shows lower prices for states around the current index level and higher prices for states with high index levels.

Furthermore in the state prices of puts we observe a phenomenon Rubinstein[1994] calls "crash–o–phobia" which could be explained as a additional insurance premium against strong negative price movements. This premium is observable for medium-term options and seems to increase with increasing time to maturity[63].

[61] A European put option only has a pay off if the price of the underlying at the expiration day is below its strike price and a European call option only if the price is above its strike price.

[62] For short-term options the trained parameter region is even smaller.

[63] Look at the implied distributions of puts with 90 and 180 days to expiration in figure 20.

Figure 16:
State–Price Density

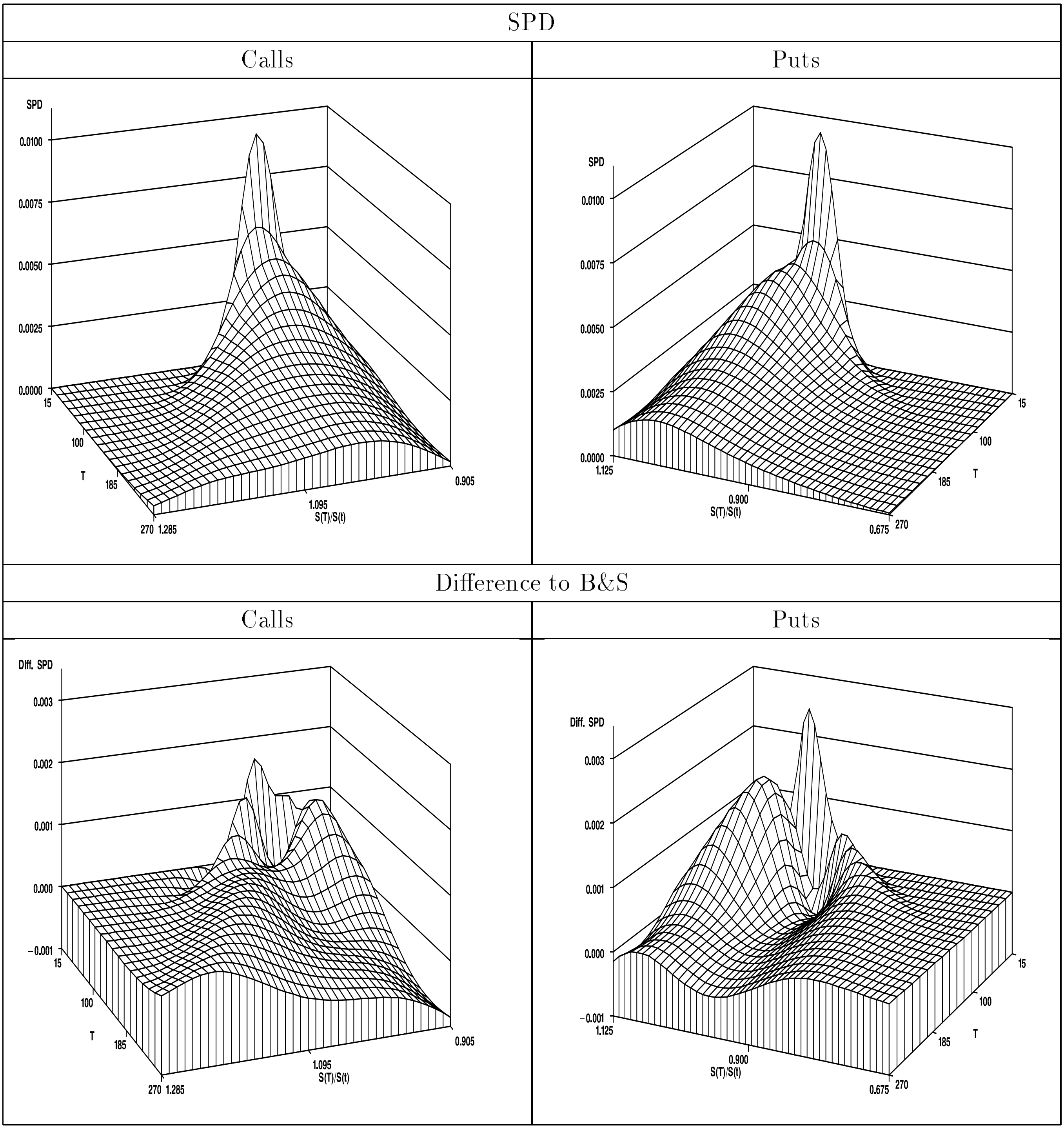


Figure 17:
Implied State–Price–Density

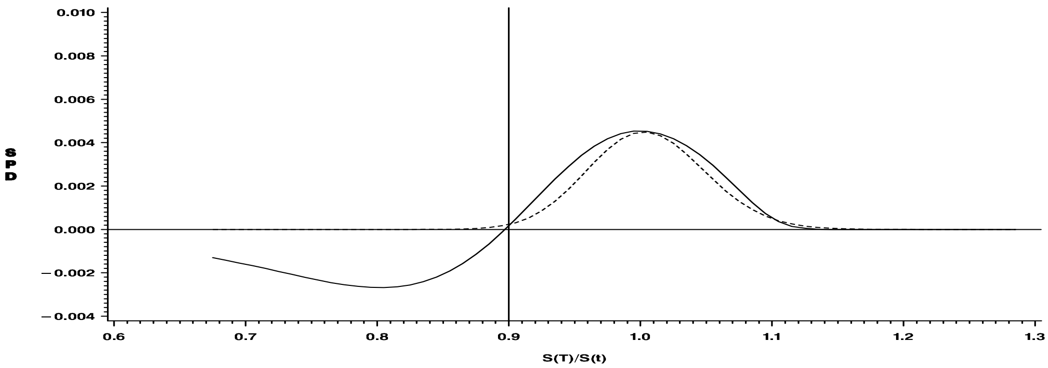
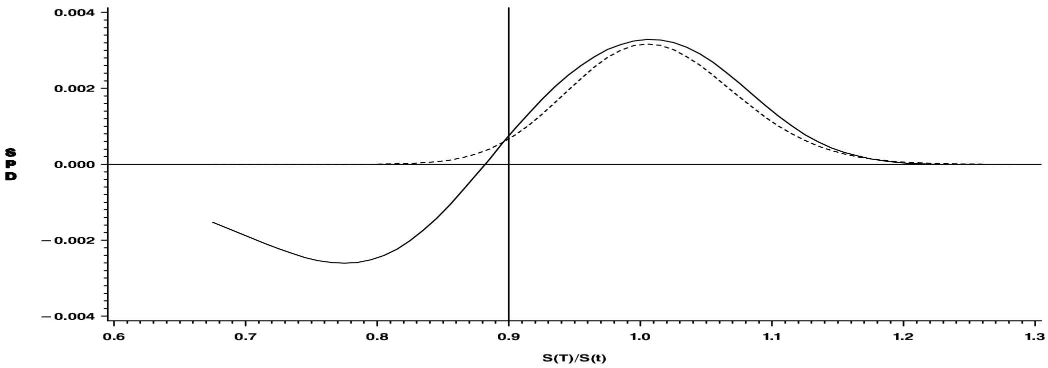
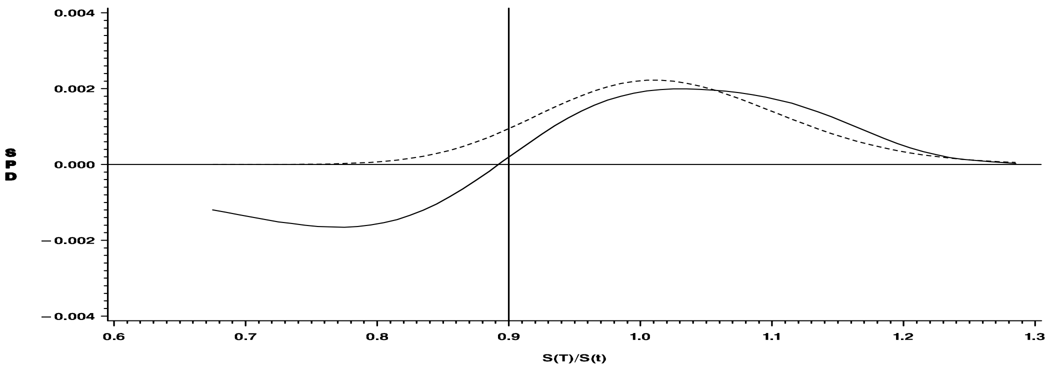
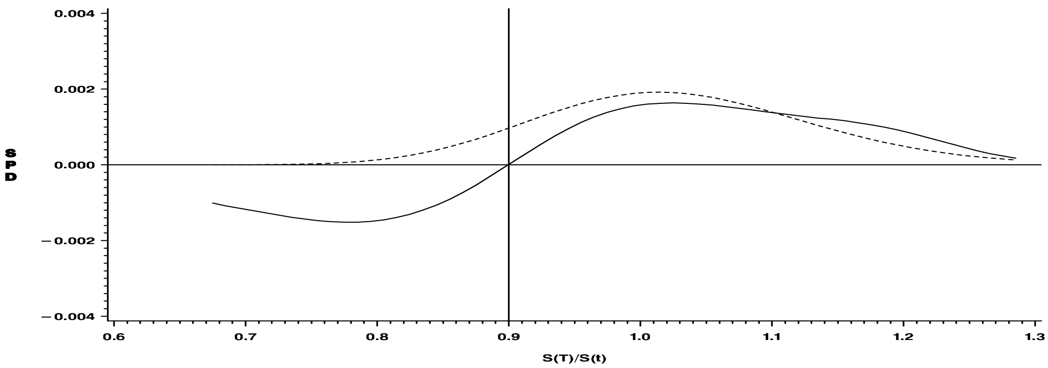
<table>
<tr><th>T</th><th>Calls</th></tr>
<tr><td>45</td><td>
</td></tr>
<tr><td>90</td><td>
</td></tr>
<tr><td>180</td><td>
</td></tr>
<tr><td>240</td><td>
</td></tr>
</table>

Figure 18:
Implied State–Price–Density

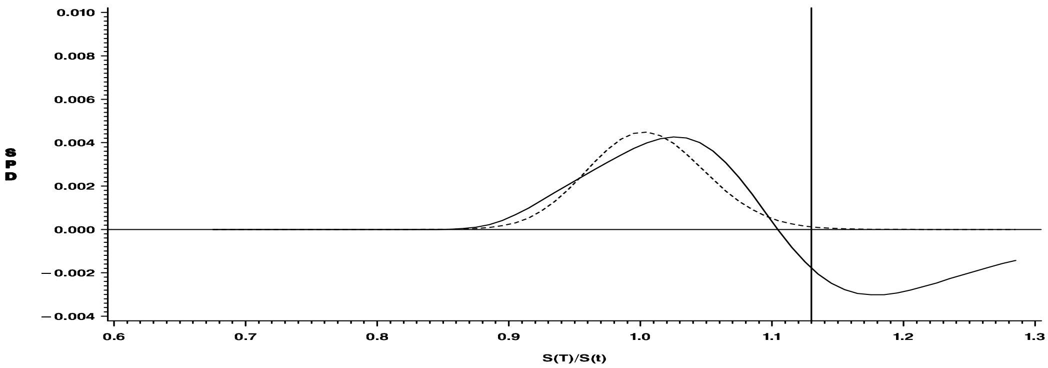
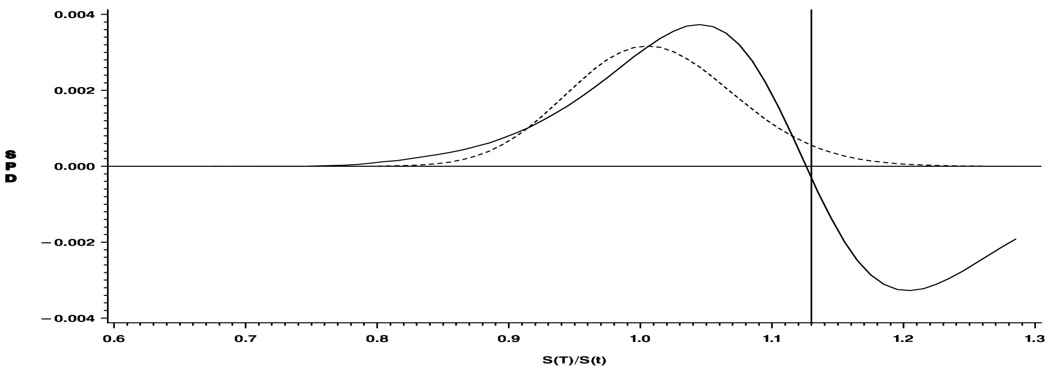
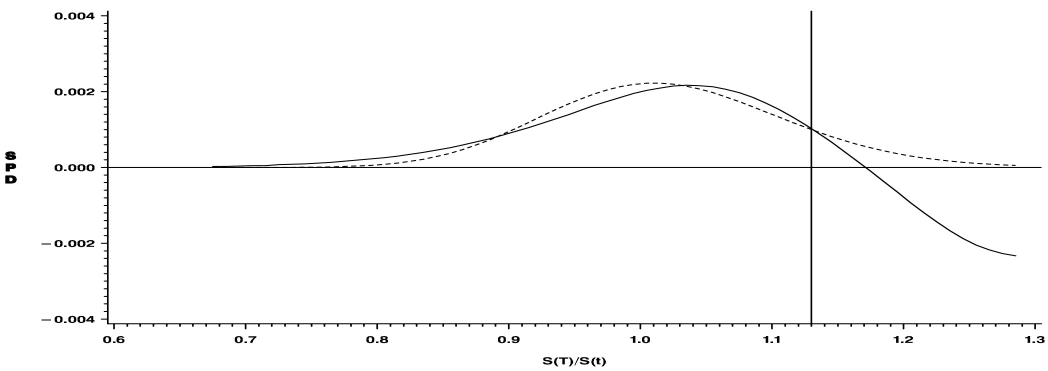
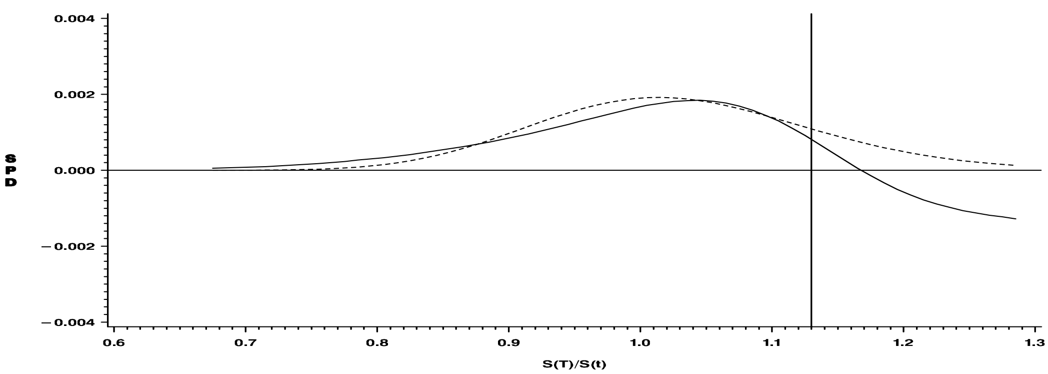
T	Puts
45	
90	
180	
240	

Figure 19:
State–Price–Density Differences to B&S

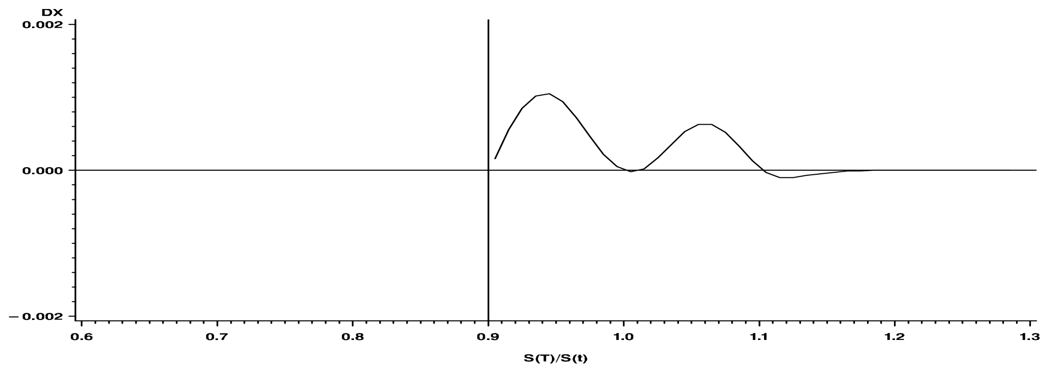
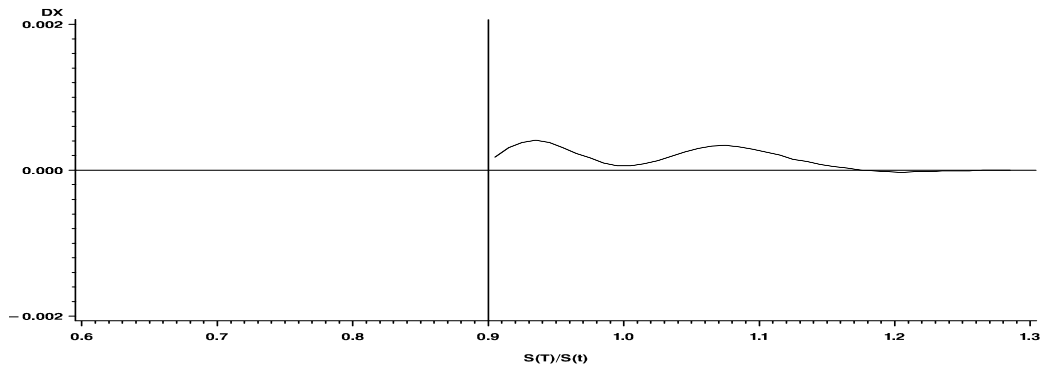
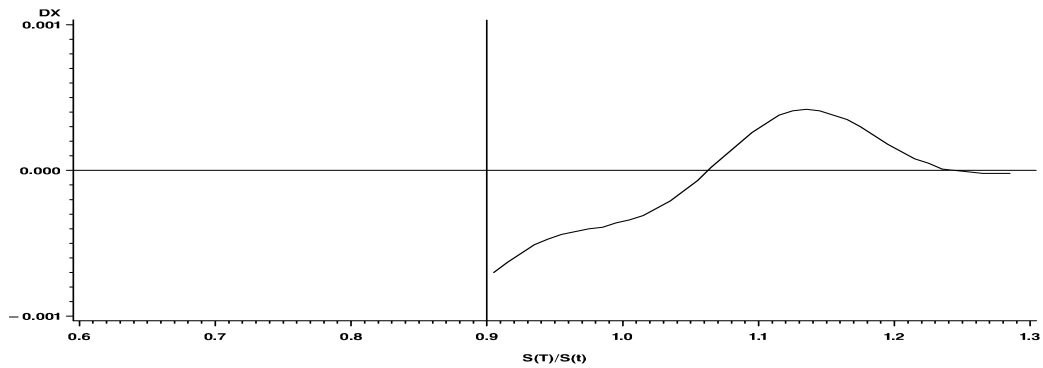
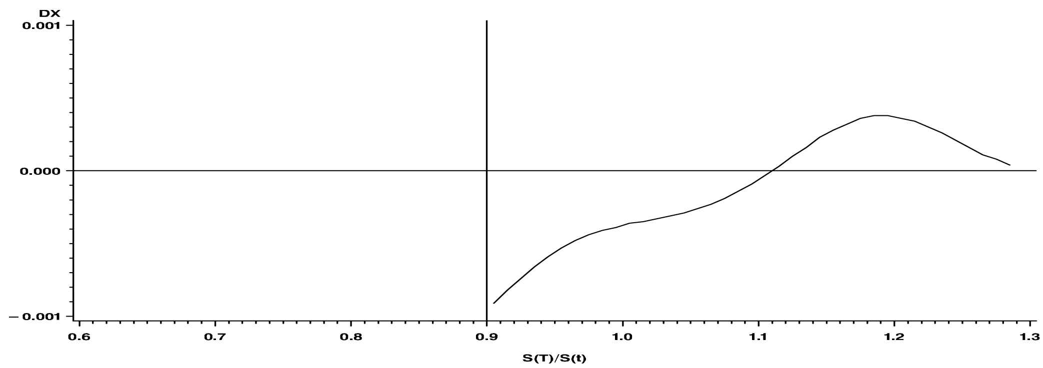
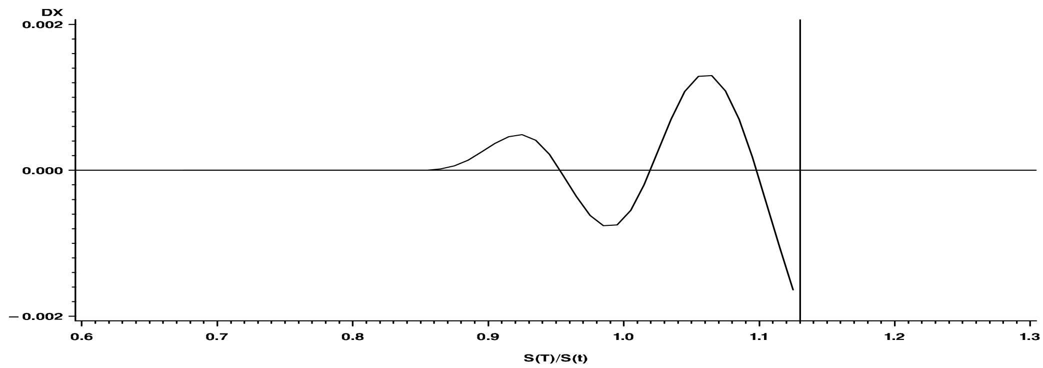
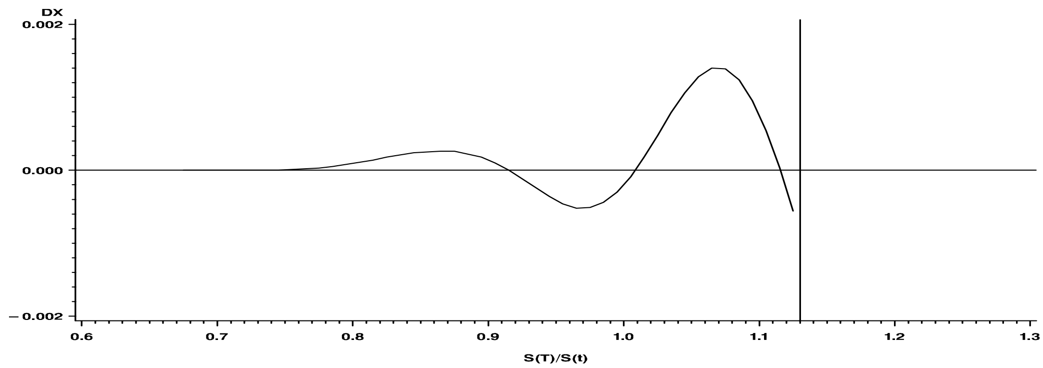
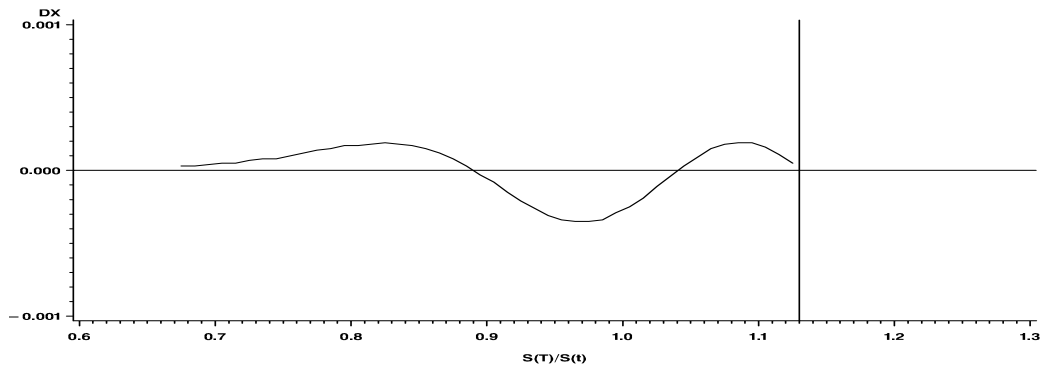
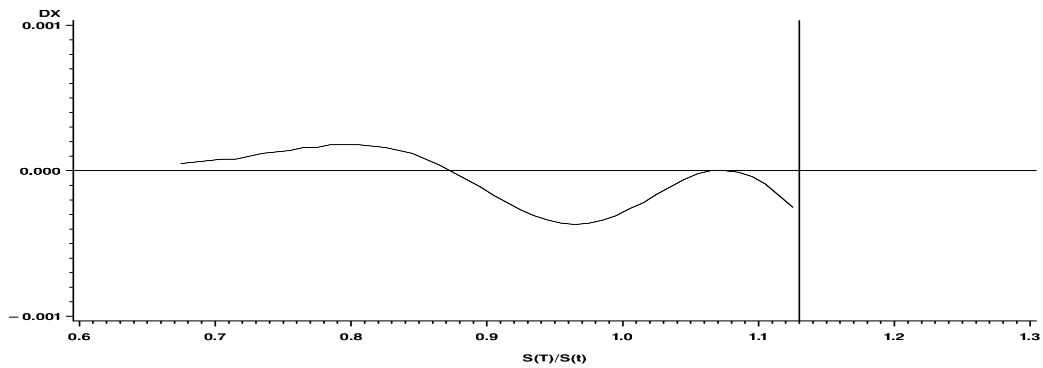
T	Calls
45	
90	
180	
240	

Figure 20:
State–Price–Density Differences to B&S

T	Puts
45	
90	
180	
240	

5 Conclusions

We demonstrated how to extract implied pricing formulas from market data by applying ANN. So getting deeper insights into the pricing mechanism used by the market participants. Looking at the empirical results we see that:

- The used ANN's are able to interpolate the Black & Scholes formula and its partial derivatives very well.
- ANN's are able to explain market prices far better than the Black & Scholes model.
- The implied pricing formula differs significantly from the Black & Scholes formula — even in the case of the DAX options which meet ideal requirements for the application of the Black & Scholes formula.
- Especially the partial derivatives — Rho and Vega — differ significantly from the corresponding derivatives of the Black & Scholes model. This differences may be explained by correlations between the input parameters, which the Black & Scholes model does not account for. Increasing volatility seems to have a negative impact on in–the–money options.
- The other partial derivatives have a similar shape compared to the corresponding B&S derivatives but differ through the strength of their impact.
- The DAX options traded at the DTB exhibit a "crash–o–phobia" phenomenon.

The most interesting question for ongoing research is if we can learn something from the market or looking from a somewhat other point of view if the implied market formulas are really better then the B&S model. Looking at the resulting hedging strategies of well–trained neural networks should give insights if the market prices are fair and the markets efficient.

Once trained, such networks are a very fast and easily handled valuation tool. Hence ongoing research is going to focus on the valuation of derivatives which are hardly, for example only with time-costly methods, valuable. Another interesting field of application are derivatives for which no closed–form valuation formula is available.

The application and the comparison of different network types as well as "learning" algorithms should allow to improve our results. Furthermore in the training process additional input factors could be incorporated to account for example for market mikro structure issues or liquidity. All these tasks will be left for further research.

References

[1] **Anders, U./Korn, O./Schmitt, C. (1996)** *Improving the Pricing of Options – A Neural Network Approach – .* ZEW Discussion Paper 96-04.

[2] **Arrow, K. (1964)** *The Role of Securities in the Optimal Allocation of Risk Bearing.* Review of Economic Studies 31, p. 91-96.

[3] **Barone-Adesi, G./Whaley, R.E. (1987)** *Efficient Analytic Approximation of American Option Values.* The Journal of Finance 42, Vol. 2, p. 301-320.

[4] **Black, F. / Scholes M. (1973)** *The Pricing and Options and Corporate Liabilities.* The Journal of Political Economy, Vol. 3, p. 637-654.

[5] **Boek, C./Lajbcygier, P./Palaniswami, M./Flitman A.(1995)** *A Hybrid Neural Network Approach to the Pricing of Options.* Proceedings ICNN '95, Perth Australia.

[6] **Breeden, D.T. / Litzenberger, R.H. (1978)** *Prices of Sate–contingent Claims Implicit in Option Prices.* Journal of Business, Vol. 51, Nr. 4, S. 621-651.

[7] **Brenner, M. / Galai D. (1981)** *The Properties of the Estimated Risk of Common Stocks Implied by Option Prices.* Working Paper No. 112, University of California-Berkeley.

[8] **Cox, J./Ross, S./ Rubinstein, M. (1979)** *Option Pricing: A Simplified Approach* Journal of Financial Economics 7, p. 229-263.

[9] **DBAG (1995a)** *VDAX.* Deutsche Börse AG, Frankfurt.

[10] **DBAG (1995b)** *DAX.* Deutsche Börse AG, Frankfurt.

[11] **DBAG (1996)** *DAX DTB Deutsche Terminbörse.* Deutsche Börse AG, Frankfurt.

[12] **Debreu, G. (1959)** *Theory of Value.* John Wiley and Sons, New York.

[13] **Friedman, J.H. (1994)** *An Overview of Predicitive Learning and Function Approximation.* In: Cherkassky, V. / Friedman, J.H. / Wechsler, H. (eds.), "From Statistics to Neural Networks", Springer Verlag, Berlin 1994.

[14] **Hertz, J. / Krogh, A. / Palmer, R.G. (1991)** *Introduction to the Theory of Neural Computation.* Lecture Notes Volume I, Addison Wesley Publishing Company.

[15] **Hornik, K./Stinchcombe, M./White, H. (1989)** *Multilayer Feedforward Networks are Universal Approximators.* Neural Networks, Vol. 2, p. 359-366.

[16] **Hornik, K./Stinchcombe, M./White, H. (1990)** *Universal Approximation of an Unknown Mapping and Its Derivatives Using Multilayer Feedforward Networks.* Neural Networks, Vol. 3, p. 551-560.

[17] **Hutchinson, J. M. / Lo, A. W / Poggio, T. (1994)** *A Nonparametric Approach to Pricing and Hedging Derivative Securities Via Learning Networks.* The Journal of Finance, Vol. 49, p. 227-230.

[18] **Kelly, D. L. (1994)** *Valuing and Hedging American Options Using Neural Networks.* Working Paper, Carnegie Mellon University.

[19] **Lajbcygier, P./Boek, C./Palaniswami, M./Flitman A.(1995)** *Neural Network Pricing of All Ordinaries SPI Options on Futures.* Presented at the 3rd Conference on Neutral Networks in the Capital Markets.

[20] **Latané, H. A. / Rendleman, R. J. (1976)** *Standard Deviations of Stock Price Ratios Implied in Option Prices.* The Journal of Finance, Vol. 31, p. 369-381.

[21] **Leland, H. E. / Rubinstein, M. (1988)** *The Evolution of Portfolio Insurance.* In: Luskin, D.(ed.): Portfolio Insurance A Guide to Dynamic Hedging, John Wiley & Sons.

[22] **Lüdecke, T. (1996)** *The Karlsruher Kapitalmarktdatenbank (KKMDB): The IBIS-Tape.* Institut für Entscheidungstheorie und Unternehmensforschung, Universität Karlsruhe, Discussion Paper No. 190.

[23] **Malliaris, M. / Salchenberger, L. (1993)** *A Neural Network Model for Estimating Option Prices.* Journal of Applied Intelligence, Vol. 3, p. 193-206.

[24] **Mayhew, S. (1995)** *Implied Volatility.* Financial Analysts Journal, July-August, p. 8-20.

[25] **Merton, R.C. (1973)** *Theory of Rational Option Pricing.* Bell Journal of Economics and Management Science, Vol. 4, p. 141-183.

[26] **Min, Q. / Maddaka, G. S. (1995)** *Option Pricing Using Artificial Neural Networks: The Case of S&P 500 Index Call Options.*, Proceedings of the 3rd International Conference on Neural Networks in the Capital Market, World Scientific.

[27] **Neumann, M. / Schlag, C. (1996)** *Martingale Restrictions and Implied Distributions for German Stock Index Option Prices.* Institut für Entscheidungstheorie und Unternehmensforschung, Universität Karlsruhe, Discussion Paper No. 196.

[28] **Rubinstein, M. (1994)** *Implied Binomial Trees.* The Journal of Finance, Vol. 49, p. 771-818.

[29] **Rubinstein, M. / Leland, H.E. [1981]** *Replicating Options with Positions in Stock and Cash.* Financial Analysts Journal, July-August, p. 63-72.

[30] **Stoll, H. R. / Whaley, R. E. (1993)** *Futures and Options.* South–Western Publishing Co., Cincinnati, Ohio.

[31] **White, H (1992)** *Artificial Neural Networks.* Blackwell, Cambridge.

[32] **White, H (1995)** *Option Pricing in Modern Finance Theory and the Relevance of Artificial Neural Networks.* NIPS '95, Denver CO, November 27, 1995.

[33] **Zell, A. et al. (1995)** *SNNS Stuttgart Neural Network Simulator.* User Manual, Version 4.1 , Report No. 6/95, University of Stuttgart, Institute for Parallel and Distributed High Performance Systems (IPVR).

A Network Derivatives

In this chapter the partial derivatives of a MLP with one hidden layer and k output units is derived. To simplify the notation the following functions[64] are introduced:

$$\Omega_1(j) = \sum_{i=1}^{L_1} \omega_1(i,j)\vec{\mathrm{I}}(i) + \theta_1(j) \tag{14}$$

$$\Omega_2(k) = \sum_{j=1}^{L_2} \omega_2(j,k) f\Big[\Omega_1(j)\Big] + \theta_2(k) \tag{15}$$

whereas

L_1	:	number of input units,
L_2	:	number of hidden units,
$\omega_1(i,j)$	:	weight between unit i of input layer and unit j of hidden layer,
$\omega_2(j,k)$	:	weight between unit j of hidden layer and output unit k,
$\vec{\mathrm{I}}(i)$	:	parameter i of input vector $\vec{\mathrm{I}}$,
$\theta_1(j)$	:	bias of unit j of hidden layer and
$\theta_2(k)$	:	bias of unit k of output layer.

The output function of the output unit k of a MLP network with one hidden layer looks using (14) and (15) as follows:

$$net_k(\vec{\mathrm{I}}) = f\Big[\Omega_2(k)\Big] \tag{16}$$

The partial derivatives of the output unit k with regard to input parameter i is:

$$\frac{\partial net_k(\vec{\mathrm{I}})}{\partial \vec{\mathrm{I}}(i)} = \frac{\partial f[\Omega_2(k)]}{\partial \Omega_2(k)} \Big(\sum_{j=1}^{L_2} \omega_1(i,j)\omega_2(j,k) \frac{\partial f[\Omega_1(j)]}{\partial \Omega_1(j)} \Big) \tag{17}$$

$$\begin{aligned} \frac{\partial^2 net(\vec{\mathrm{I}})}{\partial \vec{\mathrm{I}}(i)\partial \vec{\mathrm{I}}(h)} &= \frac{\partial\Big(\frac{\partial f(\Omega_2)}{\partial \Omega_2}\Big)}{\partial \vec{\mathrm{I}}(h)} \Big(\sum_{j=1}^{L_2} \omega_1(i,j)\omega_2(j,k) \frac{\partial f(\Omega_1)}{\partial \Omega_1} \Big) \\ &+ \frac{\partial f(\Omega_2)}{\partial \Omega_2} \Big(\sum_{j=1}^{L_2} \omega_1(i,j)\omega_2(j,k) \frac{\partial\Big(\frac{\partial f(\Omega_1)}{\partial \Omega_1}\Big)}{\partial \vec{\mathrm{I}}(h)} \Big) \end{aligned} \tag{18}$$

Using a sigmoid or logistic transfer function[65] f which

$$\frac{\partial f[x]}{\partial x} = f(x)(1 - f(x)) \tag{19}$$

yields

$$\begin{aligned} \frac{\partial net_k(\vec{\mathrm{I}})}{\partial \vec{\mathrm{I}}(i)} &= f[\Omega_2(k)](1 - f[\Omega_2(k)]) \\ & \Big(\sum_{j=1}^{L_2} \omega_1(i,j)\omega_2(j,k) f[\Omega_1(j)](1 - f[\Omega_1(j)]) \Big) \end{aligned} \tag{20}$$

[64]The function can be interpreted as input of the different units.

[65]Using instead the tangens hyperbolicus function $\frac{\partial f[x]}{\partial x} = 1 - f(x)^2$.

and

$$\begin{aligned}\frac{\partial^2 net(\vec{\mathrm{I}})}{\partial\vec{\mathrm{I}}(i)\partial\vec{\mathrm{I}}(h)} &= \frac{\partial^2 f(\Omega_2)}{\partial^2\Omega_2}\Big(\sum_{j=1}^{L_2}\omega_1(i,j)\omega_2(j,k)\frac{\partial f(\Omega_1)}{\partial\Omega_1}\Big) \\ &\times \Big(\sum_{j=1}^{L_2}\omega_1(h,j)\omega_2(j,k)\frac{\partial f(\Omega_1)}{\partial\Omega_1}\Big) \\ &+ \frac{\partial f(\Omega_2)}{\partial\Omega_2}\Big(\sum_{j=1}^{L_2}\omega_1(i,j)\omega_1(h,j)\omega_2(j,k)\frac{\partial^2 f(\Omega_1)}{\partial^2\Omega_1}\Big).\end{aligned} \tag{21}$$

To take the standardization of the input data and the output data into account we have to modify the formula to get the correct derivatives. The output function of a network with standardized input and target values is given by

$$\vec{\mathrm{I}}(i) = g(\vec{\mathrm{I}}(i)^*) \tag{22}$$

$$net_k^*(\vec{\mathrm{I}}) = g^{-1}(net_k(\vec{\mathrm{I}})) \tag{23}$$

whereas

g	:	function used for standardization,
$\vec{\mathrm{I}}(i)^*$	:	non-standardized input parameter i,
$\vec{\mathrm{I}}(i)$	:	standardized input parameter i,
$net_k^*(\vec{\mathrm{I}})$	:	non-standardized network output and
$net_k(\vec{\mathrm{I}})$	:	standardized network output.

For the derivatives follows:

$$\frac{\partial net_k^*(\vec{\mathrm{I}})}{\partial\vec{\mathrm{I}}(i)^*} = \frac{\partial net_k^*(\vec{\mathrm{I}})}{\partial net_k(\vec{\mathrm{I}})}\frac{\partial net_k(\vec{\mathrm{I}})}{\partial\vec{\mathrm{I}}(i)}\frac{\partial\vec{\mathrm{I}}(i)}{\partial\vec{\mathrm{I}}(i)^*} \tag{24}$$

$$= \frac{\partial g_O^{-1}(net_k(\vec{\mathrm{I}}))}{\partial net_k(\vec{\mathrm{I}})}\frac{\partial net_k(\vec{\mathrm{I}})}{\partial\vec{\mathrm{I}}(i)}\frac{\partial g_i(\vec{\mathrm{I}}(i)^*)}{\partial\vec{\mathrm{I}}(i)^*} \tag{25}$$

and

$$\begin{aligned}\frac{\partial^2 net_k^*(\vec{\mathrm{I}})}{\partial\vec{\mathrm{I}}(i)^*\partial\vec{\mathrm{I}}(h)^*} &= \frac{\partial g_O^{-1}(net_k(\vec{\mathrm{I}}))}{\partial net_k(\vec{\mathrm{I}})\partial\vec{\mathrm{I}}(h)^*}\frac{\partial net_k(\vec{\mathrm{I}})}{\partial\vec{\mathrm{I}}(i)}\frac{\partial g_i(\vec{\mathrm{I}}(i)^*)}{\partial\vec{\mathrm{I}}(i)^*} \\ &+ \frac{\partial g_O^{-1}(net_k(\vec{\mathrm{I}}))}{\partial net_k(\vec{\mathrm{I}})}\Big(\frac{\partial^2 net_k(\vec{\mathrm{I}})}{\partial\vec{\mathrm{I}}(i)\partial\vec{\mathrm{I}}(h)^*}\frac{\partial g_i(\vec{\mathrm{I}}(i)^*)}{\partial\vec{\mathrm{I}}(i)^*} + \frac{\partial net_k(\vec{\mathrm{I}})}{\partial\vec{\mathrm{I}}(i)}\frac{\partial^2 g_i(\vec{\mathrm{I}}(i)^*)}{\partial\vec{\mathrm{I}}(i)^*\partial\vec{\mathrm{I}}(h)^*}\Big)\end{aligned} \tag{26}$$

whereas

g_i	:	function used to standardize input parameter i and
g_O	:	function used to standardize the target value.

More precisely, using g (see (12)) for standardizing the input as well as the target parameters leads to:

$$\frac{\partial net_k^*(\vec{\mathrm{I}},g)}{\partial\vec{\mathrm{I}}(i)^*} = \frac{\max(O)-\min(O)}{\max(\vec{\mathrm{I}}(i)^*)-\min(\vec{\mathrm{I}}(i)^*)}\times\frac{\partial net_k(\vec{\mathrm{I}})}{\partial\vec{\mathrm{I}}(i)} \tag{27}$$

and

$$\begin{aligned}\frac{\partial^2 net_k^*(\vec{\mathrm{I}},g)}{\partial\vec{\mathrm{I}}(i)^*\partial\vec{\mathrm{I}}(h)^*} &= \frac{\max(O)-\min(O)}{(\max(\vec{\mathrm{I}}(i)^*)-\min(\vec{\mathrm{I}}(i)^*))(\max(\vec{\mathrm{I}}(h)^*)-\min(\vec{\mathrm{I}}(h)^*))} \\ &\times \frac{\partial^2 net_k(\vec{\mathrm{I}})}{\partial\vec{\mathrm{I}}(i)\partial\vec{\mathrm{I}}(h)}.\end{aligned} \tag{28}$$

B Network Parameters

Bias (Calls 11 Nodes)	
$\theta_1(1)$	-0.58840
$\theta_1(2)$	0.37417
$\theta_1(3)$	-0.57864
$\theta_1(4)$	-0.17316
$\theta_1(5)$	1.85720
$\theta_1(6)$	-1.02409
$\theta_1(7)$	0.65206
$\theta_1(8)$	-0.97508
$\theta_1(9)$	-0.61033
$\theta_1(10)$	-0.56625
$\theta_1(11)$	-2.01742
$\theta_2(1)$	0.22074

Network Weights (Calls 11 Nodes)					
	i: Weight				
$\omega_1(i,1)$	1: 0.57201,	2:-0.27432,	3: 0.24589,	4: 0.23492,	5: -1.12052
$\omega_1(i,2)$	1: -3.54813,	2: 0.41835,	3:-0.67894,	4: -1.34480,	5: 0.53523
$\omega_1(i,3)$	1: 3.05922,	2:-0.31798,	3:-0.28761,	4: 0.59460,	5: -3.04590
$\omega_1(i,4)$	1: -6.53084,	2:-2.32410,	3: 1.25593,	4: 3.58300,	5: -4.49372
$\omega_1(i,5)$	1:-16.00039,	2:-4.05903,	3:-1.31691,	4: -3.96295,	5: 5.29610
$\omega_1(i,6)$	1: -3.60152,	2:-3.68240,	3: 0.62852,	4: -7.11775,	5: 8.73238
$\omega_1(i,7)$	1: -1.16820,	2:-0.11216,	3:-0.13322,	4: -0.72364,	5: 1.30841
$\omega_1(i,8)$	1: 1.86945,	2: 0.21300,	3:-0.01270,	4: -3.42379,	5: 3.30959
$\omega_1(i,9)$	1: -0.34539,	2:-0.03479,	3:-0.10101,	4: 0.15602,	5: 0.38350
$\omega_1(i,10)$	1: -6.75212,	2:-0.27145,	3:-1.22901,	4: 0.68794,	5: 1.50068
$\omega_1(i,11)$	1:-20.44959,	2:-1.27682,	3:-0.51192,	4:-15.64235,	5: 20.11668
	j: Weight				
$\omega_2(j,1)$	1: 0.96314,	2:-3.30625,	3: 2.35614,	4: 2.25166,	5: 5.40473,
	6: -7.92366,	7:-1.40035,	8:-4.27809,	9: -0.09130,	10: -3.74646
	11: -19.82512				

Bias (Puts 11 Nodes)	
$\theta_1(1)$	-0.45779
$\theta_1(2)$	1.19295
$\theta_1(3)$	3.08627
$\theta_1(4)$	0.20418
$\theta_1(5)$	-0.74546
$\theta_1(6)$	0.59314
$\theta_1(7)$	0.45674
$\theta_1(8)$	0.29647
$\theta_1(9)$	-0.82837
$\theta_1(10)$	0.63285
$\theta_1(11)$	0.23007
$\theta_2(1)$	2.47536

Network Weights (Puts 11 Nodes)					
	i: Weight				
$\omega_1(i,1)$	1: 0.63880,	2:-0.85715,	3:-0.66344,	4: 0.89275,	5: -2.91488
$\omega_1(i,2)$	1: -0.89603,	2:-3.10441,	3:-0.86908,	4: 3.18183,	5: -4.94099
$\omega_1(i,3)$	1:-17.40225,	2:-7.42688,	3:-0.05951,	4: 2.80797,	5: -4.55926
$\omega_1(i,4)$	1: -0.17517,	2:-0.41435,	3:-0.67073,	4:-0.00452,	5: -0.94608
$\omega_1(i,5)$	1: 6.00794,	2: 2.50173,	3:-0.34262,	4: 6.17425,	5: -7.24374
$\omega_1(i,6)$	1: -2.15197,	2: 0.95174,	3: 0.93012,	4: 0.67700,	5: -1.14964
$\omega_1(i,7)$	1: 2.52420,	2:-1.12030,	3:-0.99898,	4: 3.76600,	5: -4.76535
$\omega_1(i,8)$	1: 2.89889,	2:-1.17473,	3: 0.32146,	4:-0.45313,	5: -1.15220
$\omega_1(i,9)$	1: -0.28404,	2:-0.52024,	3:-0.78772,	4: 1.25748,	5: -2.11221
$\omega_1(i,10)$	1:-18.31992,	2:-3.88896,	3:-0.15585,	4:11.90996,	5: -15.27501
$\omega_1(i,11)$	1: -5.82130,	2:-2.08686,	3: 0.04722,	4: 3.18518,	5: -3.86175
	j: Weight				
$\omega_2(j,1)$	1: -2.55937,	2:-5.60405,	3: 5.94110,	4:-0.26057,	5: -2.12908,
	6: -2.35366,	7:-3.44292,	8: 2.43090,	9:-1.90645,	10:-23.76661,
	11: -5.29603				

C Some more Results

Table 7:
Results Market Price Approximation 1995
Time-Value

Nodes	Sample	Calls						Puts					
		ME	MAE	MPE	MAPE	RMSE	R^2	ME	MAE	MPE	MAPE	RMSE	R^2
5	in	5.518	7.53	0.464	0.689	10.41	0.949	0.432	1.99	-0.052	0.364	2.66	0.993
	out	5.519	7.53	0.453	0.692	10.41	0.949	0.430	1.99	-0.049	0.360	2.67	0.993
6	in	6.145	7.30	0.589	0.746	10.31	0.952	0.049	1.77	-0.084	0.263	2.46	0.994
	out	6.155	7.30	0.576	0.750	10.31	0.952	0.047	1.77	-0.083	0.260	2.46	0.994
8	in	2.754	5.144	0.101	0.449	7.854	0.964	-0.736	1.93	-0.111	0.252	2.66	0.993
	out	2.754	5.144	0.091	0.459	7.844	0.963	-0.739	1.92	-0.110	0.249	2.66	0.993
11	in	3.845	5.742	0.189	0.493	8.70	0.959	-0.228	1.784	-0.084	0.244	2.45	0.994
	out	3.844	5.740	0.176	0.503	8.69	0.959	-0.232	1.781	-0.083	0.241	2.46	0.994
B&S	all	2.944	3.50	0.330	0.317	4.13	0.995	-0.655	2.85	0.264	0.029	3.96	0.984

Table 8:
Results Market Price Approximation 1995
Price Difference to B&S

Nodes	Sample	Calls						Puts					
		ME	MAE	MPE	MAPE	RMSE	R^2	ME	MAE	MPE	MAPE	RMSE	R^2
5	in	1.690	3.51	0.211	0.473	4.96	0.985	-0.670	1.42	-0.050	0.151	2.10	0.996
	out	1.690	3.51	0.210	0.470	4.96	0.985	-0.675	1.42	-0.054	0.152	2.11	0.996
6	in	2.329	3.46	0.297	0.450	4.63	0.989	-1.460	1.87	-0.119	0.189	2.61	0.995
	out	2.329	3.46	0.292	0.451	4.63	0.989	-1.464	1.87	-0.122	0.191	2.61	0.995
8	in	2.347	3.75	0.315	0.485	5.19	0.985	-1.091	1.64	-0.092	0.174	2.22	0.996
	out	2.347	3.75	0.311	0.484	5.18	0.985	-1.092	1.64	-0.092	0.177	2.23	0.996
11	in	3.993	4.82	0.530	0.625	6.29	0.983	-0.933	1.59	-0.065	0.182	2.20	0.996
	out	3.993	4.81	0.528	0.624	6.38	0.983	-0.936	1.60	-0.069	0.184	2.21	0.996
B&S	all	2.944	3.50	0.330	0.317	4.13	0.995	-0.655	2.85	0.264	0.029	3.96	0.984