CITICORP Global Derivatives Research

Implied Binomial and Trinomial Trees

Dominick Samperi

February 1995
Revised March 15

Contents

1 Introduction

The constant volatility assumption that was made by Black and Scholes[2] in order to derive their well-known option pricing formula does not apply in today's market. Observed return volatility of market indices tends to vary with time and index level.

For many indices the Black-Scholes volatility implied by out-of-the-money put options is higher than that for out-of-the-money calls. For example, Figure 2 shows the Black-Scholes volatility implied by call options on the S&P500 index, as a function of strike and the time to maturity, as of 21 October 1994. (The actual call and put option prices as of this date are shown in Figure 1.)

Rubinstein[30] has noted that the implied volatility for options on major market indices has shown a strong strike-dependence since the time of the 1987 market crash. His investigations suggest that a sharp decline in the market will result in an increase in implied volatility (a phenomenon that he calls "crash-o-phobia").

These observations illustrate the need for a consistent option pricing methodology that properly incorporates the dependence of volatility on time and index level. The primary focus of this work is on the formulation and testing of such a methodology, but before we begin we will clarify the business motivation for this research.

1.1 Black-Scholes as an Interpolation Tool

In practice traders continue to use the Black-Scholes formula for the purpose of quoting option prices in terms of the implied volatility that would yield these prices. This practice is inconsistent with the original Black-Scholes model since the volatility is assumed to be constant in this model.

In effect traders are using the Black-Scholes formula as an interpolation tool. For example, given the observed market prices (implied volatilities) of two exchange traded European options with different strikes and the same time to maturity, the interpolated price for an OTC European option with some intermediate strike (and the same maturity) can be computed by interpolating the observed implied volatilities and using the Black-Scholes formula.

This practice may seem harmless, but it implicitly recognizes the fact that the Black-Scholes formula does not incorporate the true dynamics of the underlying index (it is based on the assumption that the index volatility is constant), so hedge parameters (Delta, for example) that are computed

using this formula are likely to be contaminated by model error. This can result in decreased hedging efficiency. (For a discussion of related issues see Crouhy and Galai[9].)

1.2 Average Volatility

Since the volatility of the underlying index typically varies with time and index level, the Black-Scholes implied volatility for a particular option can be viewed as an average volatility that is expected to be realized over the life of this option.

Applying such averaged volatilities to the problem of pricing more complicated options like barrier options can lead to mispricings.[1] In particular, the probability of hitting a barrier can be very sensitive to the way volatility varies with index level, and this information is lost in the averaging process.

2 Objective

The comments in the last section illustrate the need for a consistent option pricing methodology that properly factors in the dependence of index (return) volatility on time and index level. This is currently a very active area of research on the street and in academia, and many approaches have been suggested using a variety of techniques (stochastic volatility models, GARCH models, jump processes, implied diffusions, optimal control, PDE inverse techniques, filtering/smoothing, neural networks, etc.).

2.1 Implied Trees

In this work we focus on the implied diffusion approach. Our primary focus is on the problem of constructing an implied binomial or trinomial tree (or discretized diffusion) of asset prices that is consistent with observed market prices of options on this asset (and with the observed term structure of interest rates and dividend yields).

2.2 Local Volatility and Implied Volatility

The local volatility in our implied trees will vary from node to node in order to price European options of all strikes and maturities on the same

[1] The averaged volatility might be used in one of the standard formulas for pricing barrier options, or it might be used in a Monte Carlo simulation.

tree.[2] It can be shown that if all European options are priced correctly, then the (risk-neutral) movement of the underlying index (the dynamics of the underlying process in a risk-neutral world) is completely captured by the implied tree,[3] and it can (in principle) be used to price more complicated contingent claims like American options and path-dependent options.

2.3 The Correct Volatility to Use for Exotic Options

The correct volatility to use for exotic options will be determined automatically from the implied tree (based on fitting the tree to observed prices of liquid European options). Note that in this situation there is no single number that can be identified as the correct (implied) volatility to use for a particular exotic option.

One reasonable proxy for a Black-Scholes implied volatility is the average of the local volatilities at all nodes of the implied tree that can be realized given a particular market view. For example, consider a forward start European option, and assume that our market view tells us that spot is likely to be at a particular node of the tree when the option starts (that is, assume we know the appropriate forward). Then the proxy implied volatility is equal to the average of the local volatilities at all nodes in the implied tree that can be reached from the known forward node.

This proxy "implied volatility" probably has less meaning for a path-dependent option like a barrier option since certain parts of the tree may be irrelevant for pricing purposes (because the option is knocked out before a path can enter these regions).

For the general case we can define a true implied volatility (for a particular option, path-dependent or otherwise) by using the implied tree as follows. First compute the price of the option using the implied tree. Then compute the price of the same option using a standard binomial tree (using some constant volatility). By using some iterative procedure one can repeat the second step with different constant volatilities until the two prices agree. The constant volatility that makes the two prices agree is our implied volatility.

Note that this definition of implied volatility requires an implied tree, so this volatility is implied from the price of the option in question, *plus* the

[2] The traditional approach is to assign a different volatility to each option and to use this volatility to construct a standard binomial tree in order to price the option; options with different volatilities assigned must be priced on different trees.

[3] We are using the discrete analog of a result proved in Dupire[15] for the continuous time limit.

price of the European options that were used to construct the tree.

2.4 Risk Management

For risk management purposes a Black-Scholes implied volatility matrix (volatility as a function of strike and tenor, based on observed European option prices) can be maintained, and an implied tree can be constructed using this matrix instead of raw option prices.

A matrix of Vega's (volatility sensitivities, a function of strike and tenor) for a portfolio can be computed by perturbing the volatility matrix, reconstructing the implied tree, and computing the corresponding change in portfolio value.

3 Progress

Variants of the implied binomial tree methodologies documented in Derman and Kani[12] and Rubinstein[30] and the implied trinomial tree methodology sketched in Dupire[15] have been implemented and tested. Input market data may be provided in the form of raw option prices, a Black-Scholes implied volatility matrix, or an explicit implied volatility function (Black-Scholes implied volatility as a function of strike and tenor). The spot term structure of interest rates and dividend yields must also be provided. Using this information our software constructs an implied binomial or trinomial tree that can be used to price options of various types (American, barrier, etc.).

3.1 Classification of Algorithms

The Derman/Kani and Dupire algorithms are explicit in the sense that they use formulas for tree parameters that involve market prices and rates explicitly. On the other hand, the Rubinstein approach is implicit in the sense that market prices only appear in the inequality constraints of an optimization problem (these inequality constraints ensure that the prices of input European options are computed correctly modulo the bid/ask spread).

Although the explicit approaches are easier to implement, they are based on the assumption that the necessary input option prices are perfectly consistent and offer no arbitrage opportunities. There is little room for noise in the input data. Consequently, the main difficulty in using these approaches is in the interpolation and smoothing of input market price information (a two-dimensional interpolation problem).

On the other hand, Rubinstein's implicit approach has a built in slackness (the bid/ask spread) that can allow for some noise in the input data. The main difficulty with this approach is due to the large number of variables that must be solved for in the optimization step (and this problem becomes much greater when one attempts to extend Rubinstein's methodology so that it uses option price information for more than one maturity).

3.2 Sensitivity of Explicit Algorithms

Input data must be provided in the form of a smooth surface that satisfies strong convexity constraints in order to rule out arbitrage opportunities. If raw option prices are provided then a smooth surface must be constructed that satisfies all of the no-arbitrage constraints and is close in some sense to these prices. In this case the problem is further complicated by the need to construct two such surfaces, one for calls and one for puts, with the additional consistency requirement imposed by put/call parity. For this reason it is much easier to supply input option prices in the form an a Black-Scholes implied volatility matrix. This is not really a compromise since traders normally use a volatility matrix to represent their market view.

Our mathematical analysis and numerical results show that the explicit approaches to constructing an implied tree are difficult to use in practice due to the sensitivity of the computed result to small changes in the input data (see Section 6.7).

An input implied volatility surface that is smooth everywhere except for a discontinuity in its second derivative at two points (Figure 13) can lead to conditional probability distributions and a local volatility structure containing discontinuities (Figures 14 and 15).

Noise in an input call/put price surface (a small discontinuity, say) can lead to conditional probability distributions (for fixed times in the future) that oscillate wildly, as is illustrated in Figure 8 (see Section 5 for details). This kind of problem becomes more serious as the number of time steps used is increased.

On the other hand, if the input market data is properly smoothed (Figure 9, for example) and the number of time steps used is not too large (50–100, say), then it is possible to construct an implied tree with conditional probabilities (Figure 11) and local volatilities (Figure 12) that are consistent with the observed smile. In particular, the implied (skewed) conditional probability distributions can be used to compute the price of European options of all strikes and maturities, and these prices will be close to the prices computed using the Black-Scholes formula (with the appropriate strike- and

tenor-dependent implied volatility).

3.3 Rubinstein's Implicit Approach

Note that the irregularities in the (fixed time) conditional probability distributions tend to grow as the Derman/Kani algorithm builds an implied tree in the forward direction starting at the root (Figure 8). On the other hand, the Rubinstein approach tends to smooth out irregularities in the conditional probability distributions as it builds the tree in the backward direction starting from some fixed time T in the future (see Figure 20).

In this sense Rubinstein's approach is more robust, but it should be remembered that this approach only makes use of the market prices of European options that expire at some fixed time T, while the explicit approaches use the market prices for options of all maturities. Furthermore, this approach is based on strong assumptions about path probabilities, assumptions that are not made in the explicit approaches.

In spite of these limitations the Rubinstein approach may result in a useful tool that can be used in practice. Assuming that a time-T conditional distribution can be implied from option prices, Rubinstein's backwards recursive construction of the corresponding implied binomial tree is simple to implement and numerically stable.

The time-T conditional distribution can be implied using Rubinstein's optimization procedure, or by means of some fitting/smoothing technique (see Shimko[33], for example).

The resulting implied binomial tree can be viewed as a modest improvement over the use of a standard (constant variance) binomial tree (note that a standard binomial tree satisfies all of Rubinstein's assumptions). Although this tree may not price all European options accurately (especially those that expire earlier than time T), it may give a more realistic price for path-dependent options like barrier options since it contains more refined information about the time-dependence and state-dependence of local volatility.

3.4 Conclusions

We conclude from these results that the implied tree approach is capable of incorporating the observed market smile into a single tree that can be used to price all options consistently. This is accomplished by permitting the local volatility in the implied tree to vary from node to node so that

European options of all strikes and maturities are priced correctly on the tree.

The Derman/Kani approach tends to over fit the input data, and this leads to unrealistic oscillation in the implied probability distributions. The Rubinstein approach is more stable with respect to noise in the input data, but it does not make use of all of the available market information, and the algorithm is based on assumptions that require further analysis.

The Rubinstein approach may result in a methodology that can be used in practice, but it requires a (risk-neutral) conditional distribution at some fixed time in the future as input.

More accurate and robust algorithms will require new ideas and approaches. We suggest several possible directions for future research in Section 7.

3.5 Root of the Problem

The root of the problem is highlighted in Section 6.7 where we show that the problem of constructing an implied tree (or an implied diffusion) is ill-posed. This term is used by mathematicians to describe problems where small changes in the input data can lead to arbitrarily large changes in the answer. Formulating robust solution techniques for such problems typically requires specialized techniques (regularization, projection, smoothing, etc.).

3.6 Barrier Options

It is important to note that the problem of accurately pricing path-dependent options like barrier options on a tree (implied or not) can be very tricky. The computed price is very sensitive to the placement of the barrier relative to the nodes of the tree, even when a large number of time steps is used. This problem is independent of the problem of constructing an implied tree and is discussed in Samperi[32].

3.7 Overview of Remainder of Paper

In the next section we present some background information that is necessary in order to understand the numerical results discussed in Section 5. Section 6 documents the theoretical basis for the algorithms that we study in the context of continuous time models. Section 7 briefly outlines a few possible alternative directions for future research. Sections 8 through 11 document the implied tree methodologies that we use, with some of the technical details collected in the appendix.

Since the results for a Dupire-style implied trinomial tree are similar to those obtained using a Derman/Kani-style implied binomial tree, and since our implementation of the Derman/Kani algorithm is more flexible, we will not discuss numerical results for implied trinomial trees.

4 Preliminaries

In this section we discuss a few technical details that will help the reader to understand the numerical results contained in the following section. Detailed technical information is contained in the sections following the numerical results.

4.1 Implied Binomial Trees

A binomial tree (or discrete diffusion) of asset prices is shown in Figure 3. The price at each node is shown above the node, and the transition probabilities are shown above the branches to which they apply. For example, the (conditional) probability that the price will move to S_2^1 at level 2 (time $t_2 = 2\Delta t$) given that it was S_1^0 at level 1 (time $t_1 = \Delta t$) is p_1^0.

The forwards from level 2 to level 3 are also shown. To rule out arbitrage opportunities these forward prices must fall between neighboring asset prices at level 3, as is suggested by the diagram.

In a standard Cox, Ross, Rubinstein[7] (CRR) binomial tree all of the transition probabilities are equal, and prices that appear in the same horizontal row in Figure 3 are equal (for example, $S_1^1 = S_3^2$). This is no longer true for an implied binomial tree, so a more faithful picture would be obtained by moving the nodes in Figure 3 (other than the root node) in the vertical direction to reflect the actual asset prices (never permitting two nodes to collide), resulting in a twisted binomial tree.

4.2 Arrow-Debreu Prices

Focusing our attention on the nodes at level 2 (time t_2) in Figure 3, note that we have the following nodal probabilities:

$$\begin{aligned} Q_2^0 \equiv p(S_2^0|S_0^0) &= (1-p_0^0)(1-p_1^0) \\ Q_2^1 \equiv p(S_2^1|S_0^0) &= (1-p_0^0)p_1^0 + p_0^0(1-p_1^1) \\ Q_2^2 \equiv p(S_2^2|S_0^0) &= p_0^0 p_1^1 \end{aligned}$$

Here Q_2^0 is the probability that the lower most state at level 2 will be realized, and similarly for Q_2^1 and Q_2^2. Of course, we have:

$$Q_2^0 + Q_2^1 + Q_2^2 = 1$$

Now consider an "option" that pays one dollar if the lower most state at level 2 is realized and pays zero otherwise. The price of such an option can be written:

$$\lambda_2^0 = e^{-rt_2} Q_2^0$$

This is just the discounted value of the risk-neutral expected value of the payout at time t_2 (at level 2). Similarly, we can define an option that pays one dollar only if the middle state is realized. Its price is

$$\lambda_2^1 = e^{-rt_2} Q_2^1$$

Finally, we can define an option that pays only if the upper most state is realized, with price:

$$\lambda_2^2 = e^{-rt_2} Q_2^2$$

These state contingent claim prices at level 2 are known as Arrow-Debreu prices. Note that the value of a position in all three claims is

$$\lambda_2^0 + \lambda_2^1 + \lambda_2^2 = e^{-rt_2}(Q_2^0 + Q_2^1 + Q_2^2) = e^{-rt_2}$$

This should not be surprising since this position pays a dollar for sure at level 2, so it is equivalent to holding a bond that pays a dollar at time t_2.

Arrow-Debreu prices can be defined in the same way at any level m. If Q_m^j is the probability that state S_m^j will be realized at level m, then the corresponding Arrow-Debreu price is

$$\lambda_m^j = e^{-rt_m} Q_m^j$$

In our numerical results we select a few values of m and show the graph of Arrow-Debreu prices as a function of index level for each fixed value of m. The legends in these graphs show m translated into days.

4.3 Pricing Options using Arrow-Debreu Prices

Arrow-Debreu prices can be used to price contingent claims on the tree as follows. For simplicity we consider the case of a European call option with strike K that expires at time t_m (level m). The general case presents no new difficulties and is developed in later sections.

If the state realized at time t_m is S_m^j (more precisely, this is the index level corresponding to the state), then the call option pays $\max(S_m^j - K, 0)$. But we know the fair price of a claim that pays one dollar if this state is realized and pays zero otherwise, namely, λ_m^j. This means that the fair price of this "piece" of the option payout (that is, the piece corresponding to this single state) is $\max(S_m^j - K, 0)\lambda_m^j$.

Unlike an Arrow-Debreu state option which pays only in one state at level m, the real option can pay in more than one state, so we must add the fair value of each of its possible payouts, yielding the option price:

$$C(K, t_m) = \sum_{j=0}^{m} \max(S_m^j - K, 0)\lambda_m^j$$

Note that interest rate discounting is implicit in the use of the Arrow-Debreu prices.

4.4 Local Volatility

Given that we start at the level m node with price S_m^j in Figure 3, the expected value of the return after one step (to level $m + 1$) is:

$$E[R] = p_m^j \log(\frac{S_{m+1}^{j+1}}{S_m^j}) + (1 - p_m^j) \log(\frac{S_{m+1}^j}{S_m^j})$$

and the variance is:

$$\text{var}[R] = p_m^j(\log(\frac{S_{m+1}^{j+1}}{S_m^j}) - E[R])^2 + (1 - p_m^j)(\log(\frac{S_{m+1}^j}{S_m^j}) - E[R])^2$$

The corresponding local volatility (standard deviation per unit time) is therefore computed as follows:

$$\sigma = 100\sqrt{\frac{\text{var}[R]}{\Delta t}}$$

Here Δt is the time step size.

This is the local volatility that we show in cross sections in our numerical results.

4.5 Probability Distributions

Interpreting the graph of Arrow-Debreu prices at level m as a probability distribution for the index at level m can be misleading (for example, when

one tries to compute the area under the curve). It is not simply a matter of removing the interest rate discount factor.

To see this note that if $p(S_m^j|S_0^0)$ is the underlying probability distribution function, then it is related to the Arrow-Debreu price λ_m^j as follows:

$$e^{rt_m}\lambda_m^j = p(S_m^j|S_0^0)(S_m^{j+1} - S_m^j)$$

(Alternatively, one might want to center the interval that the probability applies to around S_m^j.) This shows that the probability distribution is obtained by removing the discount factor from the Arrow-Debreu price and dividing by the size of the price interval that the probability applies to.

5 Numerical Results

In this section we present the the results of several numerical experiments. We illustrate the results that can be expected when the Derman and Kani[12] or Rubinstein[30] methodologies for the construction of an implied binomial tree are used in practice.

The Dupire[15] methodology for constructing an implied trinomial tree has also been implemented and tested. The results are similar to those obtained when applying the Derman and Kani approach, so we will confine our attention to the binomial case (with the exception of the next subsection).

As the examples will illustrate, the Derman and Kani approach is very sensitive to noise in the input data, making its use in practical situations very difficult. The Rubinstein approach is less sensitive to noise (it tends to smooth out irregularities as the implied tree is constructed), but the optimization step can be tricky.

In the case of the Derman and Kani algorithm noise in the input data tends to result in very irregular probability distributions (and local volatilities) as the implied tree is built in the forward direction. In the case of the Rubinstein methodology, noise (or bad data) results in an optimization step that will not converge (or has inconsistent constraints), so the backwards recursive step never gets executed.

The algorithms used are described in later sections.

5.1 Exact Test Cases

The Derman/Kani and Rubinstein approaches to implied binomial tree construction were tested for correctness by using a standard Cox, Ross, Rubinstein[7] (CRR) binomial tree to compute option prices (assuming constant volatility, interest rate, and dividend yield).

The Derman/Kani approach constructed an implied binomial tree that agreed with the CRR tree used for option pricing to machine precision. All tree parameters at every node were compared (transition probabilities, local volatility, etc.).

To apply the Rubinstein approach an artificial spread around the exact CRR option price was used to create a bid/ask price for options with a few different strike prices and a fixed time to maturity. The resulting implied binomial tree was very close to the CRR tree, and it converged to the CRR tree as the spread was decreased to zero. (Recall that the strong path probability assumptions that are made in the Rubinstein approach are true for a standard CRR tree.)

The Dupire implied trinomial tree approach was tested for correctness by using a Boyle[4]-style trinomial tree to price options (with constant volatility, etc.). Our implementation of the Dupire approach reconstructed the Boyle tree to machine precision.

These results indicate that all three approaches to implied tree construction work perfectly in the simple situation where option prices are observed without noise in a market that values contingent claims based on a CRR binomial tree or a Boyle trinomial tree. The use of these approaches in a real market where observations are often polluted by noise and other imperfections is much more tricky, as the following examples will illustrate.

5.2 Interpolated CRR Prices

The Derman and Kani algorithm requires call and put option prices for a range of strikes (typically around spot) and a range of maturities. At a particular point in time we observe the prices of a small number of options with strikes and maturities in the range of interest, and we must interpolate/extrapolate these discrete observations onto a continuous range of strike and maturity values. The result is a function, $C(T, K)$, giving the price of a call option with a given strike and maturity, and another function, $P(T, K)$, giving the price of the corresponding put option. This interpolation problem is complicated by the requirement that these functions must satisfy several constraints in order to rule out arbitrage opportunities and to prevent instabilities from appearing in the process of constructing the implied tree.

Figure 5 shows an example of such an interpolated call option price function. The input data in this case comes from CRR call option prices computed for a discrete range of strikes and tenors. Tenors in the range $0 \leq T \leq 1$ and strikes in the range $0 \leq K \leq 450$ are sampled; 51 samples in the T direction and 101 samples in the K direction are used. The interest rate and dividend yield (continuously compounded) are zero, the volatility is 35%, and spot is 100. The interpolation is done by means of a bivariate spline, cubic in the T direction, and quartic in the K direction.

Figure 6 shows what the Arrow-Debreu prices should look like if there were no errors introduced by the interpolation process, and Figure 7 shows the actual Arrow-Debreu prices computed using the Derman/Kani approach. Here we see the first example of the kind of instability that is to be expected in the presence of noise. Note that although the distributions for nearby times to maturity appear to be reasonably accurate, irregularities appear in the distributions corresponding to later times to maturity.

In this example the number of time steps is 50. For a larger number of

time steps the irregularities appear sooner and are more pronounced.

K	CRR Price	Der Price	Difference
40	60.03	60	-0.02452
50	50.2	50.08	-0.1261
60	40.85	40.52	-0.3295
70	32.35	31.78	-0.5651
80	24.94	24.25	-0.6858
90	18.75	18.16	-0.5918
100	13.82	13.82	-0.004137
110	10.07	10.07	0.006657
120	7.281	7.273	-0.008631
130	5.231	5.188	-0.04337
140	3.706	3.619	-0.08638
150	2.555	2.493	-0.06203

Table 1: Call option prices with maturity 1 year.

Table 1 shows that in spite of the irregularities in the 365-day distribution, European call option prices computed for a range of strikes around spot are quite close to the exact prices computed from the original CRR tree. Note that the strikes that are used here do not coincide with the strikes that were used to construct the tree (except perhaps for a few isolated cases like $K = 100$), so we are looking at interpolated option prices.[4]

The conditional distributions in the implied tree thus "bend and twist" in such a way that European options of all strikes with the given maturity are priced consistently with the input call and put option prices. It is important to note that other conditional distributions that look quite different could easily lead to nearly the same prices for European options. For example, the *correct* distributions shown in Figure 6 have this property! This simply means that there are too many degrees of freedom in the space of acceptable distributions, another consequence of the ill-posedness that is discussed in Section 6.7.

These extra degrees of freedom also appear in the context of interest rate term structure construction where a unique (continuously compounded) forward rate curve is not determined by observed market rates, and common market conventions (linearly interpolating the observed yields) result in a

[4]Following the neural net analogy discussed in Section 7.9, the implied tree has "learned" about the market from training data in the form of input market prices of options with a few different strikes and tenors.

staircase shaped forward rate curve.

In the interest rate context the errors introduced by staircase forwards tend to cancel out for longer term swaps and similar products, only becoming a serious issue for products that are sensitive to isolated parts of the yield curve (like FRA's). Similarly, we see the same kind of averaging effect in the present context where European option prices are computed fairly accurately in spite of the mathematical anomalies in the conditional distributions that result from our choice of interpolation method (used to construct the input call/put option price surfaces).

K	CRR Price	Der Price	Difference
40	21.44	19.84	-1.604
50	18.38	16.83	-1.548
60	15.32	13.83	-1.492
70	12.26	10.83	-1.436
80	9.203	7.823	-1.379
90	6.287	5.021	-1.266
100	3.859	3.112	-0.7473
110	2.11	1.573	-0.5368
120	1.011	0.6594	-0.352
130	0.4044	0.1882	-0.2162
140	0.1079	0	-0.1079
150	0	0	0

Table 2: Double knock-out with maturity 1 year (80/150).

On the other hand, it should come as no surprise that prices computed for products that are sensitive to isolated parts of the conditional probability distributions will be strongly affected by the anomalies. For example, Table 2 shows the price of a double knock-out option with lower boundary at 80 and upper boundary at 150. Prices computed on a standard CRR tree (Figure 6) and on the tree implied from interpolated prices (Figure 7) are shown. Note that the prices computed on the implied tree are significantly smaller due to the fact that the index stays within the necessary band with a smaller probability (see Figure 7), so it is more likely to be knocked out.[5]

One reason for the acceptance and continued use of staircase forwards in some markets is that this has become a market standard (like the continued

[5]Both the implied and the "exact" price here must be critically analyzed in view of the delicate nature of the problem of pricing a barrier option on a tree (implied of otherwise)—see Samperi[32]

use of the Black-Scholes formula in spite of its known limitations). In the case of implied trees the mathematical anomalies resulting from our choice of interpolation method is a much more serious problem[6] and requires further study.

5.3 Perturbed CRR Prices

For a more dramatic illustration of the instabilities that can appear we consider again the "perfect" situation from Section 5.1 above, where option prices are computed using a standard CRR tree, with zero interest rate and dividend yield, volatility 35%, spot 100, and time to maturity 1 year. But now we will introduce noise by perturbing exactly one price computed from the tree.

We use 200 time steps. At time step $m = 22$ and level $j = 11$ it turns out that the correct CRR call price at this node is 3.5909. The corresponding time to maturity and strike is $T = 40.15$ (days) and $K = 102.506$, respectively. Now we cause the call price at this node to be wrong by one cent (i.e., we add 0.01 to the computed price).

After adding this perturbation it turns out that the Derman/Kani approach still builds a perfect CRR tree (zero error). This is an accident and is explained as follows. Recall that the Derman/Kani approach tries to fix arbitrage opportunities when they are detected by using neighboring nodes to fix the one that appears to be wrong (i.e., would lead to an arbitrage opportunity). It turns out that the trick that they suggest for removing an arbitrage opportunity has the property that in our situation the perturbation is removed exactly, and the correct price at the perturbed node is restored. This means that we are back in the situation of Section 5.1 and the answer is correct to machine precision.

Now we make the perturbation ten times smaller by adding 0.001 to the exact price (at the node in question). This is small enough that an arbitrage opportunity is not produced (at this node), but later on the probability distributions begin to oscillate wildly as shown in Figure 8.

Note that the perturbation was applied at time $T = 40.15$ (days), yet the distributions for maturities out to 146 days look fine. The distribution for maturities later than this become very irregular.

It should be noted that the construction of this implied tree with 200 time steps took about 88 seconds on a SPARC 20. This is due to the fact

[6]For example, ensuring that the surface has one continuous derivative is not good enough—see Section 5.5 below.

that for this example we must build a standard CRR binomial at every node of the implied tree under construction.

5.4 Volatility Matrix

Since it is very difficult to consistently interpolate both call and put option prices in such a way that both functions (of strike and time to maturity) have the right properties (absence of arbitrage constraints on the derivatives, put/call parity, etc.), a simpler alternative is to represent input call and put option prices in the form of a Black-Scholes implied volatility matrix.

Figure 9 shows an example of a trader-supplied implied volatility matrix. More precisely, the trader-supplied volatility matrix has been approximated by a bivariate cubic spline (a function of strike and tenor) in the least-squares sense.[7]

K	CRR Price	Der Price	Difference
400	72.3	72.26	-0.04613
410	63.47	63.49	0.0152
420	54.83	54.83	0.008137
430	46.73	46.71	-0.01942
440	38.95	38.97	0.02182
450	31.67	31.62	-0.04508
460	25.22	25.21	-0.007303
470	19.41	19.4	-0.007056
480	14.32	14.25	-0.0775
490	10.09	10.15	0.05396
500	6.874	6.863	-0.01076
510	4.416	4.014	-0.4021

Table 3: Call option prices with maturity T = 0.5.

Using this smoothed implied volatility surface (and the term structure of interest rates and dividend yields that was observed on the same day) the Derman/Kani approach was used to build an implied binomial tree using 50 time steps. Spot is 464.89. Graphs of the corresponding Arrow-Debreu prices and local volatilities are shown in Figures 11 and 12, respectively. For comparison we show the Arrow-Debreu prices corresponding to a standard CRR binomial tree using the at-the-money implied volatility in Figure 10.

[7] A basis of 38 splines in the T direction and 5 splines in the K direction has been used to fit 50 points in the T direction and 41 points in the K direction. The volatility values at the latter points were obtained using a finite element technique.

Note that the distribution of market-implied Arrow-Debreu prices shows a prominent skew when compared with the distribution for the CRR case, and the market-implied local volatilities are consistent with the shape of the input implied volatility surface (this "consistency" is not always observed—see Section 5.6 below).

Table 3 shows the prices of European call options that expire at time $T = 0.5$ for a range of strikes. The CRR price is the price computed using a CRR tree with the implied volatility corresponding to the given strike and maturity (read from the input volatility surface—Figure 9). The other price is computed from the implied tree. The prices computed using the implied tree appear to give a reasonable interpolation to the prices computed using CRR trees (using a different volatility for each option for the latter).

K	CRR Price	Der Price	Difference
400	45.6	71.1	25.5
410	38.77	62.41	23.63
420	36.73	53.83	17.1
430	30.17	45.78	15.61
440	23.94	38.12	14.18
450	20.31	30.85	10.53
460	15.12	24.51	9.392
470	10.58	18.78	8.198
480	8.347	13.7	5.352
490	5.056	9.675	4.619
500	3.336	6.469	3.133
510	1.619	3.696	2.077

Table 4: Knock-out with upper barrier at 550 (T = 0.5).

It should be clear from Figures 10 and 11 that a knockout option with barrier at 550 should be much more expensive in the real world (that the implied tree is supposed to model) than in a world where options are priced using a standard CRR tree with the at-the-money implied volatility. This observation is verified by the results shown in Table 4. This table shows the CRR and implied tree prices for a knock-out option with boundary at 550 that expires at time $T = 0.5$. This option is much less likely to be knocked out on the implied tree, so it is more expensive in the real world as one would expect.

In Table 5 we show the prices of European call options that expire at time $T = 0.25$ year approximately (actually, 109.5 days—see Figure 11), that is,

K	CRR Price	Der Price	Difference
400	68.65	68.62	-0.02858
410	59.36	59.37	0.01097
420	50.18	50.2	0.01908
430	41.55	41.54	-0.01475
440	33.24	33.27	0.0339
450	25.54	25.5	-0.04646
460	18.84	18.84	-0.007027
470	13.02	13.02	-0.006707
480	8.277	8.22	-0.05723
490	4.743	4.632	-0.1107
500	2.395	2.326	-0.06933
510	1.037	1.048	0.0109

Table 5: Call option prices with maturity T = 0.25.

the expiration date corresponds to an earlier time slice on the implied tree. The implied and CRR prices are again reasonably close to each other.

K	CRR Price	Der Price	Difference
400	56.94	68.61	11.67
410	48.58	59.36	10.79
420	43.76	50.19	6.437
430	35.71	41.53	5.818
440	29.31	33.26	3.954
450	22.04	25.49	3.448
460	17	18.83	1.834
470	11.42	13.01	1.587
480	7.28	8.215	0.9344
490	3.922	4.628	0.7058
500	2.033	2.323	0.2893
510	0.8359	1.045	0.2093

Table 6: Knock-out with upper barrier at 550 (T = 0.25).

Finally, Table 6 shows the prices of a knock-out option with boundary at 550 that expires at time $T = 0.25$ (approximately). The smaller difference here can be explained by the fact that the option expires sooner (less time to knock-out).

5.5 Almost Piecewise Linear Volatility Surface

For this example we use an input implied volatility matrix in the form of a surface that varies only in the K direction (see Figure 13):

$$V(K) = \begin{cases} V_0 & \text{if } K \geq 100 \\ V_0 + A(100 - K)^2 & \text{if } 100 - H \leq K < 100 \\ V_0 + AH^2 + 2AH(100 - H - K) & \text{if } K < 100 - H \end{cases}$$

For our experiment we set $V_0 = 0.15$, $H = 0.1$, and $A = 0.01$. For a fixed time slice the graph of $V(K)$ is a horizontal line when $K > 100$, a line with a constant (negative) slope when $K < 100 - H$, and these two lines are joined in such a way that the resulting curve has a continuous derivative everywhere. The second derivative is continuous everywhere except when $K = 100$ and when $K = 100 - H$.

K	CRR Price	Der Price	Difference
40	59.37	59.37	-0.0007783
50	49.97	49.96	-0.008702
60	40.59	40.57	-0.02198
70	31.34	31.34	-0.0005483
80	22.39	22.39	0.001551
90	14.14	14.14	0.003379
100	7.249	7.25	0.001079
110	3.28	3.283	0.00327
120	1.258	1.261	0.003517
130	0.4237	0.4217	-0.00204
140	0.1242	0.1247	0.0004653
150	0.03337	0.03329	-8.263e-05

Table 7: Call option prices for PW linear surface.

We assume flat interest rate and dividend term structures with (continuously compounded) rates 6% and 3%, respectively. Taking spot equal to 100, and using 100 time steps, we use the Derman/Kani approach to construct an implied binomial tree. The corresponding Arrow-Debreu prices and local volatilities are shown in Figures 14 and 15, respectively.

This example clearly illustrates the connection between the Derman/Kani approach and the observation made by Dupire that the second derivative of the the surface of call option prices with respect to strike gives the conditional probability distributions (see Section 6.7). Here we see discontinuities in the conditional distributions (of Arrow-Debreu prices) and the local

volatilities precisely where there is a discontinuity in the second derivative of the volatility surface (which implies a discontinuity in the corresponding call and put option price surfaces).

Table 7 shows the prices of European call options that expire in one year for a range of strikes. Note the close agreement between the prices computed using the implied tree and those computed using a CRR tree (using a different volatility for each option in the latter case).

K	CRR Price	Der Price	Difference
40	9.836	35.23	25.4
50	10.47	29.52	19.04
60	9.641	23.8	14.16
70	9.8	18.09	8.285
80	7.951	12.37	4.418
90	4.994	6.812	1.819
100	2.335	2.259	-0.07567
110	0.3844	0.3648	-0.01955
120	0	0	0
130	0	0	0
140	0	0	0
150	0	0	0

Table 8: Double knock-out for PW linear surface (80/120).

On the other hand, we should expect significant differences when we consider options that are sensitive to the probability distributions in a neighborhood of spot. Let us consider a double knock-out option that expires in one year with lower boundary at 80 and upper boundary at 120. Table 8 shows the prices computed for this option on the CRR and implied trees.

Note that the implied tree price is larger when the option is out-of-the-money, and it is smaller when the option is in-the-money (or at-the-money). This can be explained as follows. The large probability associated with spot ($S = 100$) acts like a hill that must be climbed in order to reach the barrier. Therefore, if the option is out-of-the-money the effect of the hill is to make the option more expensive since we must climb the hill in order to get knocked out, while if the option is in-the-money, we are "over the hill," and the option is more likely to hit the barrier, thus it is cheaper.

5.6 Damped Smile Volatility Surface

For this example we use an input implied volatility surface that varies both in the K direction and the T direction (Figure 16):

$$V(K,T) = V_0 + A(K-100)^2(1-T)$$

We set $V_0 = 0.15$ and $A = 0.20/100^2$ (this implies that $V(0,0) = V(200,0) = 0.35$). The means that the at-the-money ($K = 100$) implied volatility is 15% for all maturities, and the implied volatility increases symmetrically and smoothly as K moves away from 100. In the limit as the strike decreases to zero the implied volatility approaches 35%, and the same is true as the strike approaches 200 (and it continues to smoothly increase for $K > 200$).

Note that this surface is smooth everywhere, and that the smile effect decreases with the time to maturity until it vanishes altogether at the one year point.

K	CRR Price	Der Price	Difference
40	59.37	59.37	-5.592e-12
50	49.96	49.96	-9.862e-09
60	40.54	40.54	-8.14e-05
70	31.14	31.14	-0.000191
80	21.95	21.95	-0.0005176
90	13.65	13.65	-6.413e-05
100	7.249	7.249	3.344e-09
110	3.28	3.281	0.001163
120	1.258	1.263	0.00537
130	0.4237	0.4157	-0.008003
140	0.1242	0.1229	-0.001308
150	0.03337	0.02934	-0.004036

Table 9: Call option prices for damped smile.

We assume flat interest rate and dividend term structures with (continuously compounded) rates 6% and 3%, respectively. Taking spot equal to 100, and using 100 time steps, we use the Derman/Kani approach to construct an implied binomial tree. The corresponding Arrow-Debreu prices and local volatilities are shown in Figures 17 and 18, respectively.

Note that there is no skew in the distributions of Arrow-Debreu prices. This corresponds to the fact that the input implied volatility matrix is symmetric about spot.

On the other hand, the input term structure of implied volatility has resulted in implied local volatility profiles (Figure 18) that "smile" in the opposite direction (that is, that "frown"). This is not surprising when one considers the fact that the implied forward local volatility must average (in a complicated sense) to the correct implied Black-Scholes volatility (from now to time T), and since the latter is declining over time, the local volatilities must decline even more rapidly (away from spot).

A similar phenomenon is observed in the interest rate world where observed market prices for interest rate caps can sometimes imply a decreasing term structure of (implied) volatility.

Table 9 shows the prices of European call options that expire in one year for a range of strikes. Again we observe close agreement between the prices computed using the CRR and implied trees.

K	CRR Price	Der Price	Difference
40	37.48	37.43	-0.05883
50	31.23	31.16	-0.07588
60	24.98	24.89	-0.09294
70	18.73	18.62	-0.11
80	12.48	12.35	-0.1271
90	6.579	6.478	-0.1013
100	2.335	2.282	-0.05302
110	0.3844	0.3657	-0.01867
120	0	0	0
130	0	0	0
140	0	0	0
150	0	0	0

Table 10: Double knock-out for damped smile (80/120).

Let us again consider the double knock-out option that we used in the last subsection (expires in one year with lower boundary at 80 and upper boundary at 120). Table 10 shows the prices computed for this option on the CRR and implied trees.

Here we see much closer agreement for the case of the barrier option than we observed in the last subsection. This can be explained by the fact that the Arrow-Debreu distributions (Figure 17) for this example are not skewed and are very similar to distributions computed for a CRR tree.

On the other hand, the local volatility structure shown in Figure 18 shows that the local volatility to be expected over the life of the option

is generally larger than the at-the-money volatility (15%), making it more likely that the option will hit one of the boundaries (assuming it starts at-the-money). This explains the negative bias in the at-the-money price, and provides a starting point for understanding the negative bias in the other prices (where in/out of the moneyness must also be considered).

5.7 Rubinstein's Results

Using the data contained in Rubinstein[30] we have duplicated Rubinstein's results by using the methodology that he proposed to construct an implied binomial tree. Using the market prices of 16 call options that expire in 164 days, the optimization step results in the conditional distribution of Arrow-Debreu prices at this time. The graph of this distribution is labeled TimeP164 in Figure 19. Spot for this example is 349.19.

This figure also shows the distribution of Arrow-Debreu prices at other times between now and 164 days from now, and it shows the corresponding local volatilities on the same diagram. The local volatility curves suggest that a decline in the market should be associated with an increase in volatility (as Rubinstein has observed).

K	CRR Price	Rub Price	Difference
400	72.48	73.23	0.7525
410	63.88	64.07	0.1924
420	55.47	55.45	-0.02283
430	47.25	47.19	-0.05682
440	39.27	39.19	-0.07324
450	31.61	31.58	-0.0378
460	24.51	24.5	-0.008511
470	18.13	18.15	0.02398
480	12.75	12.75	0.003609
490	8.512	8.469	-0.04259
500	5.496	5.464	-0.03166
510	3.54	3.511	-0.02904

Table 11: Call option prices with maturity T = 0.5.

Note that the local volatility becomes very irregular in the tails of the distribution. This is due to the fact that the transition probabilities are difficult to compute here. The formulas used for building the tree involve dividing by zero when the nodal probabilities become very small.

At present we simply try to assign reasonable values to the transition probabilities when the methodology would otherwise fail. One should then only use parts of the tree where the parameters appear to be stable. The hope is that the probability of the index landing in other parts of the tree is very small.

5.8 Rubinstein's Approach using a Volatility Curve

In order to apply Rubinstein's methodology with the real market data that was considered in Section 5.4 above (the trader-supplied volatility matrix), we simply look at the slice of this matrix corresponding to the largest maturity, $T = 0.5$. (Recall that this methodology does not make use of all available market information.)

K	CRR Price	Rub Price	Difference
400	43.55	68.57	25.02
410	36.92	59.71	22.79
420	32.68	51.38	18.71
430	28.5	43.42	14.92
440	23.88	35.71	11.83
450	19.39	28.39	8.996
460	14.83	21.61	6.776
470	11.42	15.55	4.127
480	7.744	10.43	2.685
490	5.197	6.429	1.232
500	3.035	3.699	0.6645
510	1.535	2.017	0.4815

Table 12: Knock-out option with upper barrier at 550 (T = 0.5).

We used this volatility curve (the slice data is a function of K) to compute call option prices for 20 strikes K between $S - 50$ and $S + 50$ (S is spot). Then we used an artificial spread of 0.05 to create bid/ask prices $(P - 0.05/P + 0.05)$, and we applied the optimization procedure to compute the distribution of Arrow-Debreu prices at time $T = 0.5$.

The backwards recursive procedure was then used to build an implied binomial tree. The corresponding conditional distributions are shown in Figure 20, and the local volatilities are shown in Figure 21

Table 11 shows the prices computed for European call options that expire at time $T = 0.5$ for several values of the strike K. Once again we observe close agreement between the CRR and implied tree prices. Note that the

K	CRR Price	Rub Price	Difference
400	68.68	70.64	1.968
410	59.51	60.86	1.342
420	50.61	51.25	0.6375
430	41.98	41.93	-0.05335
440	33.71	33.09	-0.6202
450	25.92	24.81	-1.108
460	18.88	17.33	-1.555
470	12.81	10.94	-1.867
480	8.01	5.992	-2.019
490	4.579	2.744	-1.835
500	2.441	1.126	-1.315
510	1.265	0.5517	-0.7132

Table 13: Call option prices with maturity T = 0.25.

computed prices do not agree to within the given spread (0.05). This should not be surprising since the spread is used in the optimization step to ensure that the 20 input option prices are correctly computed to this tolerance; options with other strikes may not be priced as accurately, as this example shows.

K	CRR Price	Rub Price	Difference
400	55.5	69.89	14.4
410	47.25	60.15	12.9
420	40.85	50.58	9.728
430	34.61	41.3	6.686
440	28.05	32.5	4.449
450	21.83	24.26	2.428
460	15.87	16.81	0.9432
470	11.06	10.46	-0.5999
480	6.848	5.54	-1.309
490	3.95	2.315	-1.635
500	2.013	0.712	-1.301
510	0.9198	0.1455	-0.7743

Table 14: Knock-out option with upper barrier at 550 (T = 0.25).

Let us compare Table 11 with the corresponding result obtained in Section 5.4 using the Derman/Kani approach (Table 3). The CRR prices in

the two tables are not the same because the volatility curve used for the Rubinstein construction has been interpolated (we are not just reading the slice data from the volatility surface), and more importantly, we are using 200 time steps here, while only 50 time steps was used in Section 5.4. Nevertheless, most of the European option prices in the two tables are reasonably close.

More precisely, the Rubinstein implied tree prices for options with strikes less than 450 (spot is 465) are slightly larger than the corresponding Derman/Kani implied tree prices, while the reverse is true for options with strikes larger than 450. This can probably be explained by noting that the bi-modality in the Rubinstein distribution tends to increase the weight of smaller index levels, and this results in larger prices for out-of-the-money options.

Table 12 shows the prices computed for the knock-out option that was considered in Section 5.4 above (knock-out call with boundary at 550). In this case the Rubinstein prices are all smaller than the corresponding Derman/Kani prices (see Section 5.4, Table 4). The difference between the corresponding prices is smaller for in-the-money options than it is for out-of-the-money options. The likely explanation here is that the extra mode in the Rubinstein distribution carries more weight for the knock-out option since its price is not affected by the part of either distribution that extends beyond $S = 550$ (and the Rubinstein distribution carries more weight here—see Figures 11 and 20).

Table 13 shows the prices of European call options that expire at time $T = 0.25$ (an earlier slice of the tree). Recall that the Rubinstein approach does not use input option price information for maturities earlier than the largest in the tree ($T = 0.5$). Nevertheless, when the numbers in Table 13 are compared with the corresponding numbers generated by the Derman/Kani procedure in Section 5.4 (Table 5) we see that there is reasonably close agreement for in-the-money options ($K < 460$). The prices for out-of-the-money options are not as close, for example, when $K = 480$, the Rubinstein price is 5.99 while the Derman/Kani price is 8.22.

It appears from Figures 11 and 20 that the Derman/Kani distribution for this time slice places greater weight on *both* large and small index levels, so the results documented in the preceding paragraph cannot be easily explained.

Finally, Table 14 shows the price of a knock-out option that expires at time $T = 0.25$ with boundary at 550. These numbers can be compared with the corresponding numbers computed using the Derman/Kani approach in Section 5.4 (Table 6). Once again we observe closer agreement between the

in-the-money prices than we see between the out-of-the-money prices.

6 Implied Diffusion Models

In this section we discuss the theoretical basis for the methodology used in this paper. We focus primarily on the continuous time theory and formulate the problem to be solved in three ways.

The first formulation takes the form of a stochastic optimal control problem. The second formulation is the problem of determining a Green's function (or fundamental solution) for a partial differential equation based on a set of integral equations that must be satisfied by this function (one per observed option price). Finally, the third formulation is the problem of solving a partial differential equation inverse problem (that is, find the coefficients based on observations of the solution).

Unfortunately, all of these (roughly equivalent) problems are ill-posed. We explain what this means and why it applies to these problems below.

Applications to the construction of implied trees are discussed in Sections 8 through 11.

6.1 The Fundamental Diffusion Equation

In order to permit index return volatility to vary with time and the level of the index we generalize the Black-Scholes model by assuming that the price process (S_t) follows the equation:

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

where $r(t)$ is the risk free rate at time t (a deterministic function), $\sigma(t, S_t)$ is the local volatility at time t and index level S_t, and Z_t is a standard Brownian motion. More generally, one could consider equations where both r and σ are random path-dependent (nonanticapative) functions, but we will not consider such equations here.[8] The Black-Sholes model is the special case where r and σ are constant.

[8] Equations of this more general form are used in the Heath, Jarrow and Morton[21, 20] interest rate model where the short rate is non-Markovian.

6.2 Solution

Equation 1 is a diffusion equation,[9] and it is well-known (Karatzas and Shreve[25]) that with some mild regularity assumptions on the coefficient functions r and σ it has a unique solution that satisfies $S_0 = S$ (today's spot) with probability 1. For each fixed $T > 0$, S_T is a random variable with the probability distribution $p(S_T, T|S, 0)$.

The solution S_t is a so-called Markov process, meaning that its future behavior is completely determined by its current value (independent of past history), and its probabilistic behavior is completely determined by transition functions of the form $p(S_t, t|S_r, r)$ (the probability of the price being in a neighborhood of the value S_t at time t given that it equals S_r at time $r < t$ is $p(S_t, t|S_r, r)dS_t$). This is what permits the dynamics of this process to be modeled on a tree.

6.3 Risk-Neutral Valuation

Equation 1 describes the risk-neutral dynamics of the price process, and the standard risk-neutral valuation methodology (Duffie[13]) implies that the price of a call option with strike K and maturity T can be written:

$$C(T, K) = E[\exp(-\int_0^T r(t)dt) \max(S_T - K, 0)] \tag{2}$$

To simplify the notation below we will assume that the risk-free rate is constant, so we can write the call option price as follows:

$$C(T, K) = e^{-rT} \int \max(S_T - K, 0) p(S_T, T|S, 0) dS_T \tag{3}$$

There is a similar formula for put options:

$$P(T, K) = e^{-rT} \int \max(K - S_T, 0) p(S_T, T|S, 0) dS_T \tag{4}$$

6.4 Implied Diffusions and Optimal Control

The fundamental problem that we address in this paper can now be described in the continuous time setting.

Assume that we observe the market price of n European call options, C_i (with strike K_i and maturity T_i), $1 \leq i \leq n$, and m European put

[9] More precisely, this is a stochastic differential equation whose solution implies transition probability functions that satisfy an associated diffusion equation—see Section 6.6.

options, P_j (with strike K_j and maturity T_j), $1 \leq j \leq m$. The problem is to determine the function σ in Equation 1 so that the corresponding call and put option prices computed using Equations 3 and 4 are as close as possible to the observed market prices. Since the function σ determines the diffusion (Equation 1) we are essentially implying a diffusion from observed market prices.

We can formulate this as an optimal control problem (Fleming and Rishel[17]) as follows. Define the functional $F(\sigma)$ by

$$F(\sigma) = \sum_{i=1}^{n} (C(T_i, K_i) - C_i)^2 + \sum_{j=1}^{m} (P(T_j, K_j) - P_j)^2$$

This means we substitute σ (a function of time and index level) into Equation 1, solve this equation with $S_0 = S$, use the solution to compute $C(T_i, K_i), 1 \leq i \leq n$ and $P(T_j, K_j), 1 \leq j \leq m$ from Equations 3 and 4, respectively, and combine this with the observed market prices (C_i, P_j) to compute the sum that appears in this definition. The problem is now to find a control σ (a so-called Markovian or feedback control) in a suitable function space such that the functional $F(\sigma)$ is as small as possible.

A bid/ask spread can be factored into the analysis by assigning weights to the terms in the function to be minimized. In the end one must check that the bid/ask spreads are actually respected by the solution.

The implied tree techniques that we develop later are implicitly solving this optimal control problem (or finding an approximate solution). In the case of the explicit techniques (Derman/Kani and Dupire) where there is no optimization step the optimization is hidden in the smoothing/filtering techniques used to construct arbitrage free input call/put price surfaces.

The problem of choosing an appropriate function space of controls for the optimal control problem corresponds to the problem of reducing the dimensionality of the fitting/smoothing problem in the implied tree techniques.

After solving for σ a finite difference technique or simulation can be used to price contingent claims on this index, or this volatility function can be used to construct an implied binomial (more likely trinomial) tree and options can then be priced in the usual way.

6.5 Solving for State Prices (The Green's Function)

The problem of constructing an implied tree is closely related to the problem of constructing a discrete approximation to the Green's function (or fundamental solution—see John[24]) for the backward partial differential

equation satisfied by all options on the index (a generalized Black-Scholes equation).

The implied tree methodologies effectively back out the Green's function from observed market prices. This can be seen by noting that what we have called Arrow-Debreu state prices are nothing more than the values of a discrete version of the Green's function at the nodes of the tree (for more on this see Jamshidian[23]).

Since the price of any European option can be written as an integral of the Green's function multiplied by a function that defines the option's payout, we can also view the implied tree approach as an attempt to solve a simultaneous system of integral equations for the Green's function. The special nature of the payout for European call and put options (the ramp shape) is what enables us (in principle) to actually solve for the value of the Green's function at each node of the tree (essentially by forming butterfly positions that pick out the value of the Green's function at each node).

The Green's function satisfies the forward Kolmogorov equation (see the next subsection, and Equation 15 and 16 of Section 8 for the discrete analog satisfied by Arrow-Debreu prices), and if we knew the coefficients in Equation 1 (which means we know the forward Kolmogorov equation) we could use a finite difference technique to solve for the Green's function directly. Of course, the primary problem that we address in this paper is that we do not know these coefficients.

6.6 PDE Inverse Problem

It is well known that the transition probabilities $p(S,t|S_0,0)$ that are determined by a solution to Equation 1 satisfy the following (forward Kolmogorov) equation:

$$\frac{\partial p}{\partial t} = -\frac{\partial}{\partial S}[r(t,S)p] + \frac{1}{2}\frac{\partial^2}{\partial S^2}[\sigma(t,S)p] \tag{5}$$

(We are considering the more general form of Equation 1 where the risk-free rate is state-dependent.)

Now, assume that we have call option prices for all strikes at the fixed maturity T. From Equation 6 in the next subsection, this means that we know $p(S,T|S_0,0)$ for all S, that is, we know p along the slice $t = T$.

Typically we observe option prices for a few discrete maturities and a few discrete strikes. For each fixed maturity assume that the price information in the strike direction has been interpolated onto a smooth curve. (This is where the ill-posedness discussed in the next subsection causes problems.)

The inverse problem is now defined as follows. Assuming that we know p along a few discrete time slices (along with the asymptotic behavior of p as S approaches zero and infinity), determine the functions $r(t, S)$ and $\sigma(t, S)$ such that the solution p of Equation 5 is as close as possible to the slice data. More precisely, we require that the L^2 norm of the difference along these time slices be as small as possible. Note that in smoothing the option prices in the strike direction we must ensure that the smoothed prices are positive and $\partial^2 C/\partial K^2 \geq 0$ (so the conditional probability is positive and arbitrage opportunities do not occur).

6.7 Ill-Posedness

It turns out that the problems formulated above (and consequently the implied tree construction problem) are ill-posed. This means that the answer does not depend on the input data in a continuous fashion.

To see this first note that it follows from Equation 3 above (see Dupire[15]) that the (risk-neutral) conditional distribution, $p(S_T, T|S_0, 0)$, can be written in terms of call option prices as follows:

$$\frac{\partial^2 C(T,K)}{\partial K^2} = e^{-rT} p(K, T|S_0, 0) \tag{6}$$

Furthermore, Dupire has shown (with certain regularity assumptions) that knowledge of the conditional distributions $p(S_T, T|S_0, 0)$ (for all T) completely determines the full risk-neutral diffusion (the dynamics of the process S_t). Consequently, it is possible (in theory) to construct the diffusion based on observed European option prices, and this tree should price even path-dependent options correctly. Dupire used this observation to motivate the idea of constructing an implied tree from option prices.

Unfortunately, the equation above also shows why the problem of constructing an implied tree from observed option prices is ill-posed: option prices are observed, not their second derivatives, and the second derivative of a function is not a continuous function of a few discrete observations of the function's value. In particular, small changes in the discrete observations are consistent with arbitrarily large changes in the second derivative appearing in this equation, so these small changes are consistent with arbitrarily large changes in the conditional probability distributions $p(S_T, T|S_0, 0)$, or with wildly oscillating or negative probabilities. Additional constraints are needed.

The ill-posedness described here will typically result in arbitrage opportunities in an implied tree. The techniques that we discuss later in connec-

tion with the Derman/Kani approach for removing arbitrage opportunities do not solve the fundamental problem of ill-posedness because these techniques are local in nature. They are capable of fixing the tree locally, but the fix can have serious side-effects later in the construction process.

The standard (but non-trivial) way to deal with ill-posedness is to reduce the dimensionality of the problem by projecting onto a finite dimensional space of functions that possess desirable properties, and to look for a solution in this subspace. This obviously reduces the number of ways that the solution can move given a small change in the input data. Other approaches include regularization (forcing higher derivatives to be bounded to reduce oscillation) and smoothing techniques.

It is shown in Derman and Kani[12] that the volatility $\sigma(t, S_t)$ can be written in terms of call option prices as follows:

$$\frac{1}{2}\sigma^2(T,K)K^2\frac{\partial^2 C}{\partial K^2} - r(T)K\frac{\partial C}{\partial K} - \frac{\partial C}{\partial T} = 0 \tag{7}$$

This equation shows that the comments just made about ill-posedness also apply to the problem of determining $\sigma(t, S_t)$ from observed call option prices.

7 Other Models

In this section we outline a few possible alternative approaches to the problem of interpolating useful information from observed option prices and modeling volatility. Most of these techniques can be viewed as attempts to find the right price dynamics in the form of Equation 1 (possibly with both r and σ being state- and path-dependent). This material is not used elsewhere and may be skipped if desired.

7.1 Parametric Estimation

The simplest approach is to assume some parametric form for the volatility function (for example, a constant elasticity of variance form—see Cox and Rubinstein[8]) and to solve for parameter values that minimize the difference between computed and observed contingent claim prices.

A disadvantage of this approach is that the parameterization results in a biased solution, one that is determined by the observed data and the form of the parameterization (and this form is often chosen because it is easy to work with, not because it has some economic justification). In other words, we are not permitting the data to "speak for itself."

7.2 Direct Estimations based on Microeconomic Analysis

Alternatively, one can try to derive the coefficients $r(t, S_t)$ and $\sigma(t, S_t)$ from a microeconomic analysis of the market (He and Leland[19]), and then solve Equation 1 using a binomial or trinomial tree, or by simulation.

This approach does not have the self-correcting property of the other techniques, so computed option prices will only be as good as the model that was used to derive the coefficients.

On the other hand, this approach involves actually modeling the market instead of just approximating it by fitting and smoothing techniques. For example, market realities like transaction costs and open interest might be explicitly factored into the analysis.

7.3 Stochastic Volatility Models

In a stochastic volatility approach (see Hull and White[22] for example) we model volatility as a second random factor. That is, we assume that volatility itself satisfies an equation of the same form as Equation 1 where the random component of this second equation is correlated with the random component of the equation satisfied by the index (Equation 1).

The volatility component of the equation satisfied by the volatility will of course have a "speed" component, that is, a function corresponding to the coefficient function σ in Equation 1. This speed component is naturally referred to as the volatility of volatility.

When volatility models like this are solved (by simulation or otherwise) we typically find that the solution has the following qualitative properties.

When the correlation between the index and its volatility is positive, the two processes tend to move together, and the distribution of the index tends to have a fat right tail and a thin left tail. When the correlation is negative the opposite is true (fat left tail, thin right tail), and this appears to be consistent with the volatility smile that we observe today (see Figure 11, for example). We conclude from these observations that the correlation between the index and its volatility has a strong effect on the skewness of the index distribution.

The volatility of volatility tends to have a more symmetrical effect on the index distribution. As it is increased the kurtosis of the distribution is increased (that is, the graph of the distribution becomes taller and more narrow).

7.4 Jump Processes

Other models consider more general processes than those that can be described by Equation 1. For example, we may allow for the possibility that the process may execute random discontinuous moves (or jumps).

Such models may provide better approximations to market reality since random jumps of this kind are often observed. For more information see Cox and Ross[6] and Cox and Rubinstein[8], for example.

7.5 Pricing Volatility by Arbitrage

One disadvantage of the last two approaches (the introduction of stochastic volatility and jump processes) is that they require the specification of risk preferences (the market price of risk), and this makes it difficult to implement a consistent pricing methodology.

An approach to volatility modeling recently suggested by Dupire[14] overcomes this problem by using liquid call and put option prices to actually price volatility by arbitrage. This means that one can lock in a profit if volatility is not priced as the model suggests. This line of development follows the ideas introduced by Heath, Jarrow and Morton[21] for the purpose of modeling the movement of interest rate term structures.

7.6 Correlation

An interesting explanation of the smile based on correlation is given in Kelly[26]. He suggests that the smile can be explained by looking at the relationship between the level of the index and the correlation between the prices of its components. For example, if the correlation increases as the index level declines, then the index volatility might increase even though the volatility of every component remains the same. In this situation the market will expect higher volatility when the index declines, possibly explaining the observed smile.

Of course, this observation is just a special case of the observations that we made in our discussion of stochastic volatility models above.

7.7 Static Replication

Derman, Ergener and Kani[11] describe an interesting approach where a barrier option is priced and hedged on the basis of observed European option prices by constructing a static (that is, not rebalanced) portfolio of Euro-

pean options that has approximately the same boundary values and payout characteristics as the barrier option.

Since the European options in the hedge portfolio and the barrier option all satisfy the same partial differential equation with (approximately) the same boundary values, the theory of partial differential equations implies that the value of the hedge portfolio must be (approximately) equal to the value of the barrier option in the region enclosed by the boundary.[10] If the index hits the boundary before the barrier option matures then the hedge portfolio must be unwound into the equivalent position in cash, stock or option (depending the barrier option contract specifications). Note that the position that must be liquidated at the boundary might be quite large, and this can offset the advantages of the static nature of the hedge in some situations.

The connection between this approach and the problem of constructing an implied tree can be explained as follows. In the implied tree approach we use observed European option prices to construct the implied tree and the tree is used to price and hedge exotic options like barrier options. The fine structure of the tree should help to improve the performance of standard delta-neutral hedging. On the other hand, the static replication approach uses the same observed European option prices to construct a static hedge directly (no tree is required). A disadvantage of the static replication approach when compared with the implied tree approach is that the static replicating portfolio applies to only one exotic option, while the same implied tree can be used to price and hedge any number of exotic options.[11]

7.8 Historical Approaches

Standard time series methods like GARCH (Bollerslev[3]) attempt to forecast (or extrapolate) the future behavior of an index and/or its volatility based on historical observations. This can be viewed as a filtering process since random noise in the observations must be removed in order to reveal trends and other useful information that might be contained in the data.[12]

[10] This is very similar to the so-called method of images often used to solve PDE boundary value problems.

[11] The static replication approach essentially solves one PDE boundary value problem, while the implied tree approach constructs a Green's function that can be used to compute solutions to all boundary value problems that arise in the process of computing the price of European options.

[12] If estimates up to a particular time are permitted to factor in future information (in order to test the predictive power of a model, for example) the process is often called smoothing instead of filtering.

Other techniques like co-integration look for trends in some linear combination of different time series (Engle and Granger[16]).

A related line of research is followed in the area of non-linear filtering where a stochastic process is estimated based on noisy observations of this process or of some non-linear function of this process (Liptser and Shiryayev[27]). In the linear case the technique is known as the Kalman-Bucy filter (Øksendal[28]). This is a relatively new area that has given rise to mathematical problems for which we do not yet have good numerical solution techniques.[13] On the other hand, recent results in this area (Riedel[29]) suggest interesting new directions.

7.9 Neural Networks

There are interesting analogies between the techniques that we have discussed and techniques that are used in the field of neural networks (see Widrow[35]).

The nodes of a binomial or trinomial tree are analogous to the nodes of a neural network. The values at the output nodes of a neural network can be compared with the values computed for European options for a number of strikes and maturities on an implied tree. The process of updating our estimate of the volatility function in Equation 1 based on errors in the computed option prices can be compared with the back propagation algorithm used to train a neural network. An analogous update is done in the process of implementing a nonlinear filter or smoother (see the last subsection).

Training a neural network can be viewed as a multidimensional function approximation problem, so parallels can be drawn between neural network techniques and the problem of constructing arbitrage free input call/put price surfaces.

Connections between neural network techniques, statistical mechanics and some hard optimization problems can be found in Simic[34].

8 Implied Trees from Contingent Claim Prices

In this section we begin our discussion of implied tree construction. Fundamental ideas that are common to all of the methodologies that we discuss

[13] The problems we have in mind include the solution of stochastic partial differential equations (SPDE's). The fundamental equation for the forward rate curve that appears in the Gaussian HJM model of Brace and Musiela[5] is an example; this equation is easier to solve since it is the infinite dimensional analog of the classical Ornstein-Uhlenbeck equation.

are presented. The details of specific implied tree construction algorithms are presented in the remaining sections of the paper.

8.1 Outline

Implied tree construction amounts to the problem of estimating the tree parameters (state prices, transition probabilities, etc.) from observed market prices for a few contingent claims.

We start with the standard formula used to compute the value of a European claim in terms of a risk-neutral expectation. Then we use the law of iterated expectations to write this expectation as an iterated integral over a discrete set of time slices between today and the maturity date.

Next we assume that there are only a finite number of possible states (with non-zero probability of occurrence) in each time slice, and that transitions from one time slice to the next occur in the standard binomial or trinomial fashion. A binomial or trinomial tree results when we discretize the integrals along each time slice, taking into account our assumptions about what transitions are possible.

Along the way this (heuristic) argument shows how the tree parameters can (in principle) be determined from the prices of contingent claims.

8.2 Contingent Claim Pricing

For a fixed function $g(x)$ consider a contingent claim that pays $g(S_T)$ at time T. For example, if $g(x) = \max(x - K, 0)$, then the payout is that of a call option with strike K. If $g(x) = x - K$, then the payout is that of a forward contract with delivery price K that matures at time T. This can also be viewed as a futures contract with the usual qualifications.

Let g_t denote the fair (arbitrage-free) value today of the contract defined by the function g. By the standard risk-neutral valuation technique the value of the claim defined by g can be written (Feynman-Kac formula):

$$g_t = E[e^{-\int_t^T r(s)ds} g(S_T)] = \int e^{-\int_t^T r(s)ds} g(S_T) p(S_T, T|S_t, t) dS_T \quad (8)$$

Now subdivide the interval $[t, T]$ into n subintervals of length $\Delta t = (T - t)/n$ with endpoints $t_0, t_1, \cdots, t_n$, where

$$t_i = t + i\Delta t,\ 0 \le i \le n$$

Since we are assuming that the stock price follows a diffusion process, it has the Markov property, and its transition probabilities satisfy the following

(Chapman-Kolmogorov) formula:

$$p(S_{t_2}, t_2|S_{t_0}, t_0) = \int p(S_{t_2}, t_2|S_{t_1}, t_1)p(S_{t_1}, t_1|S_{t_0}, t_0)dS_{t_1}$$

Applying this formula for intermediate times $t_{n-1}, t_{n-2}, \cdots, t_1$ (equivalent to applying the law of iterated expectations) yields:

$$\begin{aligned} g_t &= \int e^{-\int_t^T r(s)ds} g(S_n)dS_n \int dS_{n-1}p(S_n|S_{n-1})p(S_{n-1}|S_0) \\ &= \iint e^{-\int_t^T r(s)ds} g(S_n)dS_n dS_{n-1}p(S_n|S_{n-1})p(S_{n-1}|S_0) \\ &= \cdots \\ &= \iint \cdots \int g(S_n)dS_n dS_{n-1} \cdots dS_1 \\ &\quad \times d_n p(S_n|S_{n-1})d_{n-1}p(S_{n-1}|S_{n-2}) \cdots d_1 p(S_1|S_0) \end{aligned} \tag{9}$$

Here we have simplified the notation by setting:

$$\begin{aligned} S_i &= S_{t_i} \\ p(S_{t_i}, t_i|S_{t_j}, t_j) &= p(S_i|S_j) \\ d_i &= e^{-\int_{t_{i-1}}^{t_i} r(s)ds} = e^{-r_i(t_i - t_{i-1})} \\ r_i &= \frac{1}{t_i - t_{i-1}} \int_{t_{i-1}}^{t_i} r(s)ds \end{aligned}$$

If we now assume that there are only a finite number of discrete states in each time slice, the integrals become (finite) sums, and for fixed $S_n = S_n^k$ (price in state k at time t_n) the term

$$\lambda_n^k = \sum_{S_i,\ 0<i<n} d_n p(S_n^k|S_{n-1})d_{n-1}p(S_{n-1}|S_{n-2}) \cdots d_1 p(S_1|S_0) \tag{10}$$

gives the fair value of a claim that pays 1 if the state at time t_n is S_n^k, and pays zero otherwise. (This means $g(x) = 1$ if $x = S_n^k$, $g(x) = 0$ otherwise.) This state-contingent price is often called an Arrow-Debreu price (see Arrow[1] and Debreu[10]). It is the product of the risklessly-discounted transition probabilities at each node in each path leading to the state price S_n^k.

The following fundamental contingent claim pricing relation is a consequence of Equations 9 and 10:

$$g_t = \sum_k g(S_n^k)\lambda_n^k \tag{11}$$

Since we are assuming that the riskless interest rate $r(t)$ is state-independent, it follows easily from Equation 10 that

$$\sum_k \lambda_n^k = d_n d_{n-1} \cdots d_1 \equiv D_n \equiv \frac{1}{B_n}, \tag{12}$$

where D_n is the discount factor to time t_n, and B_n is the value of a bank account at time t_n, assuming that it starts with one dollar at time t_0.

Recall that $t_0 = t$, so $B_0 = B_t = 1$, and from Equation 11 we have:

$$g_t = \frac{g_t}{B_t} = \sum_k g_n^k \lambda_n^k = \sum_k \frac{g_n^k}{B_n}(B_n \lambda_n^k) = \sum_k \frac{g_n^k}{B_n} Q_n^k, \tag{13}$$

where $Q_n^k \equiv B_n \lambda_n^k$ defines a probability measure on the n-th time slice (i.e., $\sum_k Q_n^k = 1$). This shows that the contingent claim price in units of the bank account is a martingale with respect to the measure Q_n^k (defined on the underlying probability space giving rise to the filtration represented by the tree). In other words, Q_n^k is the so-called equivalent martingale measure (Harrison and Kreps [18]). Note that Q_n^k is equal to λ_n^k normalized to sum to one along each time slice.

It follows easily from Equation 10 that:

$$\lambda_n^k = \sum_j d_n p(S_n^k | S_{n-1}^j) \lambda_{n-1}^j \tag{14}$$

If we assume that the states evolve in a binomial fashion as illustrated in Figure 3, then this reduces to:

$$\begin{aligned} \lambda_n^k &= d_n [p(S_n^k | S_{n-1}^{k-1}) \lambda_{n-1}^{k-1} + p(S_n^k | S_{n-1}^k) \lambda_{n-1}^k] \\ &= d_n [p_{n-1}^{k-1} \lambda_{n-1}^{k-1} + (1 - p_{n-1}^k) \lambda_{n-1}^k] \end{aligned} \tag{15}$$

Here we are adopting the convention: $\lambda_n^{-1} = \lambda_n^{n+1} = 0$. Note that the states in the tree are numbered starting with zero, and the number of states at level n is $n + 1$.

Similarly, if the states evolve the the trinomial fashion illustrated in Figure 4, we have:

$$\begin{aligned} \lambda_n^k &= d_n [p(S_n^k | S_{n-1}^{k-2}) \lambda_{n-1}^{k-2} + p(S_n^k | S_{n-1}^{k-1}) \lambda_{n-1}^{k-1} + p(S_n^k | S_{n-1}^k) \lambda_{n-1}^k] \\ &= d_n [p_{n-1}^{k-2} \lambda_{n-1}^{k-2} + q_{n-1}^{k-1} \lambda_{n-1}^{k-1} + r_{n-1}^k \lambda_{n-1}^k] \end{aligned} \tag{16}$$

Here $0 \le k \le 2n$, and we set $\lambda_{n-1}^{-1} = \lambda_{n-1}^{-2} = 0$, and $\lambda_{n-1}^{2n} = \lambda_{n-1}^{2n-1} = 0$. Note that the number of states at level n is $2n + 1$, and that at each node we have $p_n^k + q_n^k + r_n^k = 1$.

8.3 Constructing an Implied Tree

Equation 15 permits one solve for the λ_n^k's in the forward direction through a binomial tree, while Equation 11 permits one to solve for the prices of contingent claims in the backward direction. By choosing an appropriate set of contingent claims (call and put options, futures, the underlying, etc.), all of which must satisfy the backward relations, it is possible to impose enough constraints to solve for the necessary tree parameters (the up/down move sizes, transition probabilities, etc.). A similar approach applies to a trinomial tree if we replace Equation 15 with Equation 16.

Which contingent claims to use for the construction, and how to interpolate the market prices of these claims so we can compute prices where we need them (for example, call prices for a particular set of strikes) is the main problem that must be addressed in order to develop a robust algorithm.

This is the basis of the Derman/Kani and Dupire approaches. Both of these approaches construct the implied tree in the forward direction starting at the root.

On the other hand, in Rubinstein's approach we try to estimate the conditional probability distribution along some fixed time slice in the future using an optimization technique, and some (rather strong) assumptions are made about path probabilities in order to develop a backwards recursive algorithm for building the tree starting from this future date.

Note that all of the methodologies amount to the problem reconstructing the conditional probability distributions in each time slice of the tree based on observed European option prices. But the latter prices can be written as integrals of these probability distributions, and since integration is a smoothing operator, this means that we must back out the nuances of the probability distributions based on smoothed observations. Since information is lost in the smoothing process, this program is not feasible unless we have many (reliable) option prices to work with and/or additional constraints are imposed. This is just another way of saying that the problem is ill-posed (see Section 6.7).

A mitigating factor is that the integrals in question all have a hard upper bound (determined by the strike), and this tends to give the observed option price information more resolving power. An example of this is given in Equation 6 (Section 6).

9 The Derman/Kani Approach

Derman and Kani[12] present an implied binomial tree construction algorithm that takes as input the prices of European call and put options with strikes and maturities in the range: $K_1 \leq K \leq K_2, 0 \leq T \leq T1$. Prices must be interpolated onto this range from a few discrete observations using, for example, a bivariate interpolating spline.

The range of maturities needed can be specified a priori, but the range of possible strikes must contain the range of possible index values along each time slice of the tree. Since the index values at the nodes of the tree are determined as the tree is constructed (dynamically to adapt to new information from contingent claim prices), the range of strikes required must be determined by trial and error. This is an important observation since the interpolation procedure must be tuned to accommodate the largest and the smallest possible value of K. This issue may become less irritating if we build the tree based on forward prices, for then there is no drift.

In order to prevent arbitrage opportunities in this model interpolated call values, $C(K,T)$, must satisfy the constraints: $C \geq 0$ (obvious), $\partial^2 C/\partial K^2 \geq 0$ (butterfly) and $\partial C/\partial T \geq 0$ (conversion). Put option prices , $P(K,T)$, must satisfy similar constraints. Unfortunately, it is very difficult to satisfy these constraints using standard spline interpolation software. One possible alternative is to choose a basis of functions that all satisfy these constraints and to find a convex combination of these functions that is (least-squares) close to the observed prices.

Perhaps because of the complexity of this interpolation problem most authors (including Derman and Kani) suggest that the necessary input prices be computed based on an input Black-Scholes implied volatility matrix. Note that this is not cheating; this is just another way of specifying the needed input option prices, and this procedure is probably more natural for traders anyway.

As Derman and Kani have suggested (see the example in their paper) and we have verified, the algorithm tends to be more stable if a CRR tree is used to compute a Black-Scholes price when needed instead of using the Black-Scholes formula. That is, in order to compute a one-step option price at a node of the implied tree under construction (see the next subsection), we build a standard CRR tree using the implied volatility that applies to this node (based on the strike and tenor), and use this CRR tree to price the option.

Since this means that a different CRR tree is needed for each node of the implied tree under construction, this methodology can be very expensive.

For this reason our implementation permits the user to specify how the implied volatility matrix is used: prices can be computed using a CRR tree, or using the Black-Scholes formula.

For the courageous user our implementation also supports input in the form of raw call and put prices, but a fair amount of tuning of the spline fitting/smoothing software is then required.

9.1 The Construction

At the root of the tree (Figure 3) we know today's spot, S_0^0, and the Arrow-Debreu price for a dollar to be received today: $\lambda_0^0 = 1$. Derman and Kani show how to use forward induction to compute the stock prices and Arrow-Debreu prices for all times t_m, $1 \leq m \leq n$. The transition probabilities are also computed along the way. (The detailed computations can be found in Appendix A.)

Assume that we know the stock prices and Arrow-Debreu prices at time t_{m-1}. It is shown that the price S_m^{j+1} can be written in terms of prices at the previous time step, known call option prices, forward prices, and S_m^j as follows:

$$S_m^{j+1} = \frac{S_m^j(C_m^j/d_m - \Sigma_{j+1}) - \lambda_{m-1}^j S_{m-1}^j(F_m^j - S_m^j)}{(C_m^j/d_m - \Sigma_{j+1}) - \lambda_{m-1}^j(F_m^j - S_m^j)} \tag{17}$$

where

$$\Sigma_{j+1} = \sum_{k=j+1}^{m-1}(F_m^k - S_{m-1}^j)\lambda_{m-1}^k \tag{18}$$

Here C_m^j is the price of a call option that is struck at S_{m-1}^j and expires at time t_m, and $F_m^j = S_{m-1}^j e^{(r_m - q_m)\Delta t}$ is the (risk-neutral) forward price to time t_m of the underlying asset conditional on state j at time t_{m-1}.

Similarly, we show that the price S_m^j can be written in terms of prices at the previous time step, put option prices, forward prices, and S_m^{j+1} as follows:

$$S_m^j = \frac{S_m^{j+1}(P_m^j/d_m - \Sigma_{j-1}) + \lambda_{m-1}^j S_{m-1}^j(F_m^j - S_m^{j+1})}{(P_m^j/d_m - \Sigma_{j-1}) + \lambda_{m-1}^j(F_m^j - S_m^{j+1})} \tag{19}$$

where

$$\Sigma_{j-1} = \sum_{k=0}^{j-1}(S_{m-1}^j - F_m^k)\lambda_{m-1}^k \tag{20}$$

Here P_m^j is the price of a put option that is struck at S_{m-1}^j and expires at time t_m. (Note that case is important in the definition of P_m^j since we have already used p_m^j to denote transition probabilities.)

These recurrence relations at level m can be used to solve for all states at this level given a reasonable assumption about the states at the center of the tree. We set the value of the central node in the case when m is even equal to today's spot (when m is even, the number of states at this level is odd). When m is odd, we set the two central nodes so that the average of their logarithms is equal to the logarithm of today's spot. This implies that $S^+S^- = S^2$, where S^+ is the price at the node just above the center of the tree, and S^- is the price at the node just below the center. The node just below the center (when m is odd) has index $j = \frac{m-1}{2}$, and it is shown in Appendix A that the value of the node just above center is:

$$S_m^{j+1} = \frac{S[C_m^j/d_m + \lambda_{m-1}^j S - \Sigma_{j+1}]}{\lambda_{m-1}^j F_m^j - C_m^j/d_m + \Sigma_{j+1}},\ j = \frac{m-1}{2} \tag{21}$$

The transition probabilities as shown in Figure 3 are given by:

$$p_{m-1}^j = \frac{F_m^j - S_m^j}{S_m^{j+1} - S_m^j} \tag{22}$$

so the probabilities for the previous level are determined when we compute the state prices for the current level.

Finally, the Arrow-Debreu prices λ_n^k can be computed using Equation 15.

9.2 Fixing Arbitrage Opportunities

In the process of constructing the tree we must ensure that the transition probabilities are always between zero and one. This is obvious from a mathematical point of view, but it is also required from a financial point of view to prevent arbitrage opportunities. From the expression for the transition probabilities we see that the requirement that $0 \le p_{m-1}^j \le 1$ implies:

$$S_m^j \le F_m^j \le S_m^{j+1}$$

and if we combine this with the same requirement for the next larger value of j we get:

$$F_m^j \le S_m^{j+1} \le F_m^{j+1}$$

When this constraint is violated we override the price S_m^{j+1} with the price implied by the requirement that the logarithmic difference between the new

price and its already computed neighbor (at the same level m) is equal to the logarithmic difference between corresponding prices at the previous level.

For example, if $S_m^{j+1} > F_m^{j+1}$ (so the constraint is violated), and if we are working our way upwards (using calls), then S_m^j must already have been computed and satisfies the no-arbitrage condition:

$$F_m^{j-1} \leq S_m^j \leq F_m^j$$

The equality of logarithmic differences in this case becomes:

$$\log(S_m^{j+1}) - \log(S_m^j) = \log(S_{m-1}^{j+1}) - \log(S_{m-1}^j)$$

or

$$\frac{S_m^{j+1}}{S_m^j} = \frac{S_{m-1}^{j+1}}{S_{m-1}^j}$$

which leads to

$$S_m^{j+1} = \frac{S_{m-1}^{j+1}}{S_{m-1}^j} S_m^j = \frac{F_m^{j+1}}{F_m^j} S_m^j \leq F_m^{j+1}$$

where we have used the constraint known to be satisfied by S_m^j.

This fix, suggested by Derman and Kani, appears to have worked. That is, S_m^{j+1} satisfies one half of the no-arbitrage constraint. Unfortunately, there is no guarantee that we have not over-compensated and forced S_m^{j+1} below the lower no-arbitrage boundary, and in practice this can happen. Furthermore, the Derman and Kani fix does not apply to the situation where one of the central nodes violates the no-arbitrage constraint.

In our implementation we try to use the Derman and Kani fix, but when this fails we simply set the bad index price equal to the average of the upper and lower no-arbitrage limits (that is, the average of the neighboring forward prices from the previous time step). At the extremes of the tree we must manufacture one of these limits when necessary by extrapolation.

For typical input data the fixes discussed in this section tend to be used more often (in the process of constructing the tree) than seems desirable. This is a symptom of the fact that we typically cannot guarantee the complete absence of arbitrage in the input market data (in whatever form we provide it).

9.3 The Algorithm

The algorithm for constructing the implied tree can be summarized as follows:

1. Initialize S_0^0 to today's spot, set $\lambda_0^0 = 1$, and compute the forward for the first step: $F_1^0 = S_0^0 e^{(r_1 - q_1)\Delta t}$.

2. Given the prices and forwards from level $m-1$, compute the value of the central node(s) at level m using the method described above. If m is odd (so there are two central states), fill in the transition probability for the (unique) central node at level $m-1$ using Equation 22.

3. Use the recurrence relations (Equations 17 and 19) to compute prices above and below the central node(s) at level m, and fill in the transition probabilities for level $m-1$ as soon as the necessary prices have been determined. If at any point a computed price violates the no-arbitrage condition, override the price as described above.

4. Compute the Arrow-Debreu prices for level m using Equation 15.

5. If this is level n we are done. Otherwise, compute the forwards needed for the computation at the next level, and go to Step 2.

10 Rubinstein Style Implied Binomial Trees

Rubinstein's approach can be viewed as a conservative one where we try to remain as close to the well-known and well-understood structure of a standard CRR binomial tree as possible. This model requires input option prices for European options that expire at some fixed time T. A disadvantage of this model is the fact that it does not use input option prices for other maturities.

The algorithm consists of two steps, an optimization step and a backwards recursive construction step.

10.1 Optimization Step

In the optimization step nodal probabilities at time T are computed by requiring that these probabilities be as close as possible to standard CRR probabilities computed using some characteristic (constant) volatility, subject to the constraint that the input option prices are computed correctly (to within the bid/ask spread). A quadratic optimization procedure with linear equality and inequality constraints is used to complete this step.

Given today's spot, the maturity T, the number of time steps n, the interest rate and dividend term structures, and the characteristic volatility,

we begin by constructing a standard CRR binomial tree with these parameters. Let $\lambda^i_{CRR}, 0 \leq i \leq n$ denote the Arrow-Debreu prices in the final time slice (recall that these are equal to the nodal probabilities times the discount factor to time T). Then the optimization step involves finding perturbed prices $\lambda^i_n, 0 \leq i \leq n$ such that

$$\sum_{i=0}^{n} (\lambda^i_n - \lambda^i_{CRR})^2$$

is as small as possible, subject to the constraint that the observed market prices of m contingent claims are computed correctly modulo the bid/ask spread (recall Equation 11):

$$B_j \leq \sum_{i=0}^{n} g_j(S^i_n)\lambda^i_n \leq A_j, \ 1 \leq j \leq m$$

Here g_j is the payoff function corresponding to one of m contingent claims whose market prices are observed to lie between the bid/ask values B_j/A_j, $1 \leq j \leq m$. For call options we have $g_j(x) = \max(x - K_j, 0)$, for put options $g_j(x) = \max(K_j - x, 0)$, and for the underlying index we have $g_j(x) = x$ (in this latter case the bid/ask values equal the bid/ask prices for the underlying index).

We assume that the first claim represents a riskless bond maturing at time T and that there is no spread associated with this claim. Since the bond pays 1 in all states at time T, $g_1(x) \equiv 1$, and we have $A_1 = B_1 = D_T$, where D_T is the discount factor to time T (i.e., the spot price of the bond). Therefore, the first constraint reduces to (recall Equation 12):

$$\sum_{i=0}^{n} \lambda^i_n = D_T$$

Of course, this is equivalent to the requirement that the sum of the corresponding state probabilities at time T is equal to 1 (so we have a valid conditional distribution).

The main difficulty in this step is caused by the large number of unknowns that must be solved for (especially when the number of time steps is large), and the fact that observed option prices can easily lead to an optimization problem with no solution (that is, one where the set of constraints is inconsistent). Furthermore, it is not at all clear that the functional to be minimized here is the best choice.

10.2 Backwards Recursive Construction

Starting with the states and probabilities at time T that were computed in the optimization step, the second step involves working backwards from time T using recursive relations to compute intermediate states, transition probabilities, etc., until we reach time 0 (today).

The recursive relations are based on the assumption that the probability of occurrence of all paths leading to the same node at time T is the same. It is shown in Appendix B that this implies the following backwards recursive relationship between Arrow-Debreu prices at one time step and its immediate predecessor:

$$d_m \lambda_{m-1}^j = (1 - \frac{j}{m})\lambda_m^j + \frac{j+1}{m}\lambda_m^{j+1} \tag{23}$$

It is also shown that the transition probabilities p_{m-1}^j can be written in terms of Arrow-Debreu prices as follows:

$$p_{m-1}^j = \frac{\frac{j+1}{m}\lambda_m^{j+1}}{\lambda_{m-1}^j d_m} \tag{24}$$

Finally, the following backwards relationship between state prices is shown:

$$S_{m-1}^j = e^{(q_m - r_m)\Delta t}[p_{m-1}^j S_m^{j+1} + (1 - p_{m-1}^j)S_m^j] \tag{25}$$

The backwards recursive procedure starts with the states and Arrow-Debreu prices at level n (time T) that were computed in the optimization step. Then Equation 23 is used to compute Arrow-Debreu prices at level $n-1$. Transition probabilities from the states at level $n-1$ are then computed using Equation 24, and the index prices for this level are computed using Equation 25. It should now be clear how this procedure can be used to complete the construction by working backwards to time 0.

Equation 24 illustrates the primary problem with this step. What do we do when λ_{m-1}^j is zero or very small? In our implementation we simply set p_{m-1}^j equal to some convenient value (between 0 and 1) when this happens. This ad hoc approach seems reasonable when we argue that λ_{m-1}^j is typically close to zero only in the tails of the distribution where the index will fall with very small probability. Therefore, the value that we compute for the transition probabilities in this region may not be critical.

This is, of course, not a rigorous analysis. This problem is just one symptom of the ill-posedness that we mentioned earlier, and requires further study.

11 Dupire Style Implied Trinomial Trees

The methodology sketched by Dupire[15] can be used to build an implied trinomial tree (see Figure 4) using the market prices for various contingent claims. It is similar in spirit to the Derman/Kani approach, but the current implementation tends to be less stable. One possible explanation for this is the fact that at each level m the states are updated starting with the extreme nodes (either the 0-th or the $2m$-th node), and the calculation proceeds towards the central nodes (unlike the Derman/Kani algorithm where we start at the central nodes and work towards the extremes). This probably has the effect of propagating the noise that will inevitably be contained in the tails of the distribution (due to machine roundoff, for example) into the central portion of the tree (where the important information is).

On the other hand, I do not think modifying the algorithm to start from the center will do much good (this doesn't seem to help much in the Derman/Kani approach, for example). The main cause of the instabilities in inverse problems like these is the unrealistic oscillation that tends to characterize the solution. In the case of implied trees this oscillation is most evident in the implied local volatility. A regularization technique is needed, one that results in a global smoothing of the computed volatility function.

11.1 Some Relationships

Getting back to Dupire's approach (only sketched in his paper), and using the same notation that we used for the Derman/Kani analysis, it is shown in Appendix C that:

$$\lambda_m^j = \frac{C_m^{j-2} - \sum_{k=j+1}^{2m}(S_m^k - S_{m-1}^{j-2})\lambda_m^k}{S_m^j - S_{m-1}^{j-2}},\ 2 \le j \le 2m \tag{26}$$

When $j = 2m$ the summation term is zero and we can solve for λ_m^{2m} (the Arrow-Debreu price at the upper-most node—see Figure 4). This formula can then be used recursively to solve for λ_m^j in the downward direction (all the way down to λ_m^2 if desired).

It is also shown in Appendix C that

$$\lambda_m^j = \frac{P_m^j - \sum_{k=0}^{j-1}(S_{m-1}^j - S_m^k)\lambda_m^k}{S_{m-1}^j - S_m^j},\ 0 \le j \le 2m-2 \tag{27}$$

Here we can solve for λ_m^0 directly, and this formula can be used recursively to solve for λ_m^j in the upward direction (all the way up to λ_m^{2m-2} if desired).

Note that the range of j values covered by these two formulas does not overlap when $m = 1$. Therefore this is a special case that is handled as follows:

$$\lambda_1^0 = \frac{P_1^0}{S_0^0 - S_1^0} \tag{28}$$

$$\lambda_1^2 = \frac{C_1^0}{S_1^2 - S_0^0} \tag{29}$$

$$\lambda_1^1 = d_1 - \lambda_1^0 - \lambda_1^2 \tag{30}$$

where the last equality follows from Equation 12.

Another result derived in Appendix C is the following, valid for $0 \leq j \leq 2m - 2$:

$$r_{m-1}^j = [\frac{\lambda_m^j}{d_m} - q_{m-1}^{j-1}\lambda_{m-1}^{j-1} - p_{m-1}^{j-2}\lambda_{m-1}^{j-2}]/\lambda_{m-1}^j \tag{31}$$

$$p_{m-1}^j = \frac{F_m^j - S_m^j - (1 - r_{m-1}^j)(S_m^{j+1} - S_m^j)}{S_m^{j+2} - S_m^{j+2}} \tag{32}$$

$$q_{m-1}^j = 1 - r_{m-1}^j - p_{m-1}^j \tag{33}$$

Recall that we are using the convention that $\lambda_{m-1}^{-1} = \lambda_{m-1}^{-2} = 0$ and $\lambda_{m-1}^{2m} = \lambda_{m-1}^{2m-1} = 0$.

11.2 The Construction

The algorithm can now be described. We begin by constructing a standard Boyle-style trinomial tree using an opening parameter (Boyle[4] calls it λ) that will accommodate the local volatilities that are needed to support the observed market prices. This parameter determines the opening of the tree (similar to the Courant-Friedrichs-Lewy ratio for PDE's—see John[24]) and must be determined by trial and error.

At this point we have a trinomial tree with index prices already filled in, and the problem reduces to computing Arrow-Debreu prices and transition probabilities that result in a tree that is consistent with observed market prices.

Next we compute the Arrow-Debreu prices λ_1^j, $0 \leq j \leq 2$ by using the special case formulas above, and for each $m > 1$ in turn we first use Equation 26 to compute λ_m^j for $m \leq j \leq 2m$ (working downwards), and then we use Equation 27 to compute λ_m^j for $0 \leq j < m$ (working upwards).

After computing the Arrow-Debreu prices for all nodes we sweep through the tree again to fill in the transition probabilities using the formulas for $r^j_{m-1}, p^j_{m-1}, q^j_{m-1}$ given above.

The main difficulty in the construction is dealing with near-zeros in the denominator of various expressions and keeping the transition probabilities valid (that is, between 0 and 1).

A Derman/Kani Details

Assume that the states evolve in the usual binomial fashion as shown in Figure 3. At the root of the tree we know today's spot, S^0_0, and the Arrow-Debreu price for a dollar to be received today: $\lambda^0_0 = 1$. We will show below how to use forward induction to compute the stock prices and Arrow-Debreu prices for all times t_m, $1 \le m \le n$. The transition probabilities are also computed along the way.

Assume that we know the stock prices and Arrow-Debreu prices at time t_{m-1}. Corresponding to the price S^j_{m-1}, $0 \le j \le m-1$, let C^j_m denote the time-t price of a call option struck at this price that expires at time t_m. In terms of the function $g(x)$ introduced in Section 8.2, the g corresponding to C^j_m is $g(x) = \max(x - S^j_{m-1}, 0)$, and the option pays $g(S_m)$ at time t_m. Similarly, let P^j_m denote the time-t price of a put option struck at S^j_{m-1} that matures at time t_m.

Since we will be working exclusively with transition probabilities at level $m-1$, we will abbreviate p^j_{m-1} to p_j to simplify the notation.

Since we are assuming a risk-neutral process, we have:

$$F^j_m \equiv S^j_{m-1} e^{(r_m - q_m)\Delta t} = p_j S^{j+1}_m + (1 - p_j) S^j_m \tag{34}$$

Using Equations 11 and 15, we can write:

$$\begin{aligned} C^j_m &= d_m \sum_{k=0}^{m} \max(S^k_m - S^j_{m-1}, 0)[p_{k-1}\lambda^{k-1}_{m-1} + (1-p_k)\lambda^k_{m-1}] \\ &= d_m \sum_{k=j+1}^{m} (S^k_m - S^j_{m-1})[p_{k-1}\lambda^{k-1}_{m-1} + (1-p_k)\lambda^k_{m-1}] \end{aligned} \tag{35}$$

Replacing k with $k+1$ in the term involving p_{k-1} yields:

$$d_m \sum_{k=j}^{m-1} (S^{k+1}_m - S^j_{m-1}) p_k \lambda^k_{m-1} = d_m (S^{j+1}_m - S^j_{m-1}) p_j \lambda^j_{m-1}$$

$$+ \quad \sum_{k=j+1}^{m-1} (S_m^{k+1} - S_{m-1}^j) p_k \lambda_{m-1}^k$$

Combining this with the rest of the expression (and using $\lambda_{m-1}^m = 0$) gives:

$$\begin{aligned} C_m^j &= d_m (S_m^{j+1} - S_{m-1}^j) p_j \lambda_{m-1}^j \\ &+ d_m \sum_{k=j+1}^{m-1} (p_k S_m^{k+1} + (1-p_k) S_m^k - S_{m-1}^j) \lambda_{m-1}^k \\ &= d_m (S_m^{j+1} - S_{m-1}^j) p_j \lambda_{m-1}^j \\ &+ d_m \sum_{k=j+1}^{m-1} (F_m^k - S_{m-1}^j) \lambda_{m-1}^k \end{aligned} \tag{36}$$

Here we have used Equation 34 in the last step.

From Equation 34 we have:

$$p_j = \frac{F_m^j - S_m^j}{S_m^{j+1} - S_m^j}$$

Substituting into the previous expression yields:

$$(S_m^{j+1} - S_{m-1}^j) \lambda_{m-1}^j \frac{F_m^j - S_m^j}{S_m^{j+1} - S_m^j} = \frac{C_m^j}{d_m} - \Sigma_{j+1} \tag{37}$$

where Σ_{j+1} is the (known) sum in Equation 36.

Solving for S_m^{j+1} in terms of S_m^j, we find:

$$S_m^{j+1} = \frac{S_m^j (C_m^j / d_m - \Sigma_{j+1}) - \lambda_{m-1}^j S_{m-1}^j (F_m^j - S_m^j)}{(C_m^j / d_m - \Sigma_{j+1}) - \lambda_{m-1}^j (F_m^j - S_m^j)} \tag{38}$$

The put options P_m^j can be used in the same way. The equation corresponding to Equation 35 above is:

$$P_m^j = d_m \sum_{k=0}^{j} (S_{m-1}^j - S_m^k)[p_{k-1} \lambda_{m-1}^{k-1} + (1-p_k) \lambda_{m-1}^k]$$

Replacing k with $k+1$ in the term involving p_{k-1} (and using $\lambda_{m-1}^{-1} = 0$) yields:

$$d_m \sum_{k=-1}^{j-1} (S_{m-1}^j - S_m^{k+1}) p_k \lambda_{m-1}^k = d_m \sum_{k=0}^{j-1} (S_{m-1}^j - S_m^{k+1}) p_k \lambda_{m-1}^k$$

Combining this with the rest of the expression yields:

$$\begin{aligned} P_m^j &= d_m(S_{m-1}^j - S_m^j)(1-p_j)\lambda_{m-1}^j \\ &+ d_m \sum_{k=0}^{j-1}(S_{m-1}^j - [p_k S_m^{k+1} + (1-p_k)S_m^k])\lambda_{m-1}^k \\ &= d_m(S_{m-1}^j - S_m^j)(1-p_j)\lambda_{m-1}^j \\ &+ d_m \sum_{k=0}^{j-1}(S_{m-1}^j - F_m^k)\lambda_{m-1}^k \end{aligned} \tag{39}$$

From Equation 34 we have:

$$(1-p_j) = \frac{S_m^{j+1} - F_m^j}{S_m^{j+1} - S_m^j}$$

Substituting into the previous expression gives:

$$(S_{m-1}^j - S_m^j)\lambda_{m-1}^j \frac{S_m^{j+1} - F_m^j}{S_m^{j+1} - S_m^j} = \frac{P_m^j}{d_m} - \Sigma_{j-1} \tag{40}$$

where Σ_{j-1} is the (known) sum in Equation 39.

Finally, solving for S_m^j in terms of S_m^{j+1} yields:

$$S_m^j = \frac{S_m^{j+1}(P_m^j/d_m - \Sigma_{j-1}) + \lambda_{m-1}^j S_{m-1}^j (F_m^j - S_m^{j+1})}{(P_m^j/d_m - \Sigma_{j-1}) + \lambda_{m-1}^j (F_m^j - S_m^{j+1})} \tag{41}$$

The recurrence relations given by Equations 38 and 41 at level m can be used to solve for all states at this level after we have fixed the central node(s).

Following Derman and Kani[12] we fix the central node(s) by setting the central price equal to today's spot when m is even (so the number of nodes at this level is odd), and when m is odd we set the prices that straddle the center of the tree so that the average of their logarithms is equal to the logarithm of today's spot:

$$\log(S) = \frac{1}{2}[\log(S_m^+) + \log(S_m^-)]$$

Here $S_m^+ = S_m^{\frac{m+1}{2}}$ and $S_m^- = S_m^{\frac{m-1}{2}}$, and we have:

$$S_m^+ S_m^- = S^2$$

Taking $j = \frac{m-1}{2}$, note that $S^j_{m-1} = S$, and use Equation 34 to conclude that:

$$p_j = \frac{F^j_m S^{j+1}_m - S^2}{(S^{j+1}_m)^2 - S^2}$$

Substituting into Equation 36 yields:

$$\frac{C^j_m}{d_m} = (S^{j+1}_m - S)\lambda^j_{m-1}\frac{F^j_m S^{j+1}_m - S^2}{(S^{j+1}_m)^2 - S^2} + \Sigma_{j+1}$$

Canceling the common factor and simplifying yields:

$$(S^{j+1}_m + S)(\frac{C^j_m}{d_m} - \Sigma_{j+1}) = \lambda^j_{m-1}F^j_m S^{j+1}_m - \lambda^j_{m-1}S^2 \tag{42}$$

Finally, solving for S^{j+1}_m (where $j = \frac{m-1}{2}$), we find:

$$S^{j+1}_m = \frac{S[C^j_m/d_m + \lambda^j_{m-1}S - \Sigma_{j+1}]}{\lambda^j_{m-1}F^j_m - C^j_m/d_m + \Sigma_{j+1}},\ j = \frac{m-1}{2} \tag{43}$$

The transition probabilities as shown in Figure 3 are given by:

$$p^j_{m-1} = \frac{F^j_m - S^j_m}{S^{j+1}_m - S^j_m} \tag{44}$$

so the probabilities for the previous level are determined when we compute the state prices for the current level.

B Rubinstein Details

Fix a level m (time t_m) and state i at this level, where $m \leq n$ and $0 \leq i \leq m$. Using Rubinstein's assumption that all paths leading to the same state at level n (time t_n) have the same risk-neutral probability of occurrence, we will derive an expression for $P(S^j_n \cap S^i_m)$ (the probability that a path starting at S^0_0 passes through the states S^i_m and S^j_n), and using this we will derive backward relationships for Arrow-Debreu prices and transition probabilities.

Note that Rubinstein's assumption for level n (that all paths leading to the same state at this level have the same probability of occurrence) immediately implies the same for levels $m < n$. This can be proved by induction, starting with level $n-1$, and working backwards. It is easy to see that if the assumption is not true at level $n-1$, then it cannot be true

at level n, a contradiction. Therefore, the assumption must be true at level $n-1$, completing the induction step.

The set of paths from S_0^0 to S_n^j that pass through S_m^i can be partitioned into $n+1$ disjoint subsets, $U_j, 0 \leq j \leq n$, by defining U_j to be the set of paths through S_m^i (m and i are fixed here) that ultimately reach S_n^j. The probability of the state S_m^i can therefore be written:

$$P(S_m^i) = \sum_{j=0}^{n} P(U_j) = \sum_{j=0}^{n} P(S_n^j \cap S_m^i) \tag{45}$$

By definition we have

$$P(S_n^j | S_m^i) = \frac{P(S_n^j \cap S_m^i)}{P(S_m^i)} \tag{46}$$

Since all paths to S_n^j (from S_0^0) have the same risk-neutral probability, $P(S_n^j \cap S_m^i)$ is equal to $P(S_n^j | S_0^0)$ times the fraction of these paths that pass through the state S_m^i. The total number of paths that reach S_n^j is $n!/(j!(n-j)!)$. It is not hard to see that the number of distinct paths between the states S_m^i and S_n^j is $(n-m)!/((j-i)!(n-m-j+i)!)$ if $i \leq j \leq i+n-m$, and the number is zero when $j < i$ or $j > i+n-m$. Therefore, the number of paths to S_n^j that pass through S_m^i is equal to this number times the number of paths between S_0^0 and S_m^i (which is $m!/(i!(m-i)!)$). Let $A_j(m,i)$ denote the fraction of paths to S_n^j that pass through S_m^i. We have shown that:

$$A_j(m,i) = \frac{m!}{i!(m-i)!} \frac{(n-m)!}{(j-i)!(n-m-j+i)!} \frac{j!(n-j)!}{n!} \tag{47}$$

when $i \leq j \leq i+n-m$, $A_j(m,i) = 0$ otherwise.

It then follows that

$$P(S_n^j \cap S_m^i) = A_j(m,i) P(S_n^j | S_0^0)$$

To simplify the notation let $Q_n^j = P(S_n^j | S_0^0)$ denote the terminal node probabilities. It then follows from Equation 45 that

$$Q_m^i \equiv P(S_m^i) = \sum_{j=0}^{n} A_j(m,i) Q_n^j \tag{48}$$

and from Equation 46 we have

$$P(S_n^k | S_m^i) = \frac{A_k(m,i) Q_n^k}{\sum_{j=0}^{n} A_j(m,i) Q_n^j} \tag{49}$$

Note that $d_1 d_2 \cdots d_m Q_m^i = \lambda_m^i$, so Equation 48 can be written in terms of Arrow-Debreu prices as follows:

$$d_{m+1} d_{m+2} \cdots d_n \lambda_m^i = \sum_{j=0}^{n} A_j(m,i) \lambda_n^j \tag{50}$$

When $m = 1$ it is easy to check that:

$$A_j(1,i) = \begin{cases} 1 - \frac{j}{n} & \text{when } i = 0,\ 0 \le j \le n-1 \\ \frac{j}{n} & \text{when } i = 1,\ 1 \le j \le n \\ 0 & \text{otherwise} \end{cases}$$

Similarly, when $m = n - 1$ (so we are one step away from the terminal date), $A_j(m,i)$ is zero unless $i \le j \le i + n - m = i + 1$, and it is easy to check that

$$A_j(n-1,i) = \begin{cases} 1 - \frac{i}{n} & \text{if } j = i \\ \frac{i+1}{n} & \text{if } j = i+1 \\ 0 & \text{otherwise} \end{cases}$$

It then follows from Equation 50 that

$$d_n \lambda_{n-1}^i = (1 - \frac{i}{n}) \lambda_n^i + \frac{i+1}{n} \lambda_n^{i+1} \tag{51}$$

Recall (see Figure 3) that $p_{n-1}^i = P(S_n^{i+1} | S_{n-1}^i)$, so if we set $m = n - 1$ and $j = i + 1$ in Equation 49 above and use the expression for $A_j(n-1,i)$, we find:

$$p_{n-1}^i = \frac{\frac{i+1}{n} \lambda_n^{i+1}}{(1 - \frac{i}{n}) \lambda_n^i + \frac{i+1}{n} \lambda_n^{i+1}} = \frac{\frac{i+1}{n} \lambda_n^{i+1}}{d_n \lambda_{n-1}^i} \tag{52}$$

where the second equality follows from Equation 51.

As we showed above Rubinstein's assumption about path probabilities at level n implies the same assumption for earlier levels. This permits us to conclude that the last two equations remain valid if n is replaced with any m satisfying $1 \le m \le n$.

Equation 34 can now be used to solve for the states at level $m - 1$, $S_{m-1}^j,\ 0 \le j \le m - 1$:

$$S_{m-1}^j = e^{(q_m - r_m)\Delta t} [p_{m-1}^j S_m^{j+1} + (1 - p_{m-1}^j) S_m^j] \tag{53}$$

Let us connect the results of this section with those in Rubinstein[30, 31]. Using Rubinstein's notation we have $P_j(u^k d^l) = P(S_n^j | S_m^i)$, where $i = k$, and $m = k + l$. Rubinstein's $X_j(m,i)$ is related to our $A_j(m,i)$ by

$$A_j(m,i) = \frac{m!}{i!(m-i)!} X_j(m,i),$$

and our Equation 49 above agrees with the equation on Page 809 of Rubinstein[30] (a common factor in front of the A_j's can be canceled, yielding an expression in terms of the X_j's). Rubinstein shows how to use these conditional probabilities to estimate the greeks (Delta, Gamma, Theta) from the implied tree.

Rubinstein works with path probabilities instead of nodal probabilities. Let T_m^i denote the probability of a path from S_0^0 to S_m^i (recall that every path to this node has the same probability of occurrence). We have:

$$T_m^i = \frac{Q_m^i}{\frac{m!}{i!(m-i)!}} = \frac{i!(m-i)!}{m!} Q_m^i$$

It then follows from Equations 47 and 48 that:

$$T_m^i = \sum_{j=0}^{n} B_j(m,i) T_n^j$$

where

$$B_j(m,i) = \frac{(n-m)!}{(j-i)!(n-m-j+i)!}$$

when $i \leq j \leq i+n-m$, $B_j(m,i) = 0$ otherwise.

Taking $m = n-1$, we have:

$$B_j(n-1,i) = \begin{cases} 1 & \text{if } j = i \text{ or } j = i+1 \\ 0 & \text{otherwise} \end{cases}$$

Therefore,

$$T_{n-1}^i = T_n^i + T_n^{i+1} \tag{54}$$

It is easy to see that $T_n^{i+1} = T_{n-1}^i p_{n-1}^i$, so we have:

$$p_{n-1}^i = \frac{T_n^{i+1}}{T_{n-1}^i} = \frac{T_n^{i+1}}{T_n^i + T_n^{i+1}} \tag{55}$$

Rubinstein suggests that the last two equations (together with Equation 53) be used to perform the backwards recursive step. Since the path probabilities are very small when the number of time steps is large this can lead to numerical problems due to computer roundoff. For example, if the number of time steps is $n = 50$, then the number of paths to the central node ($j = 25$) is

$$\frac{50!}{(25!)^2} \approx 1.26 \times 10^{14}$$

so the probability of an individual path is small indeed.

Accordingly, we prefer to work with Arrow-Debreu prices directly by using Equations 51 and 52 instead. Note that this does not completely resolve the numerical problems due to roundoff since the Arrow-Debreu prices are very small in the tails of the distribution.

C Dupire Details

In this section we show how to construct an implied trinomial tree from observed market prices for European options and the spot interest rate and dividend term structures. The methodology is similar to that used by Derman/Kani in the binomial case.

Although a trinomial tree has more degrees of freedom than a binomial tree, the methodology described here for constructing a trinomial tree is not as flexible as the Derman/Kani approach. For example, the index values are fixed at the beginning of the algorithm based on a (constant) volatility assumption, and the problem reduces to computing the corresponding Arrow-Debreu prices and transition probabilities so that the input option prices are reproduced. On the other hand, in the Derman/Kani approach the index values are computed dynamically as the tree is constructed. Unfortunately, as we have seen, neither approach really address the fundamental problem of ill-posedness.

Using the same notation that was used in the Derman/Kani analysis, it follows from Equation 11 that the value of a European call option with strike S^j_{m-1} that expires at time m can be written (see Figure 4):

$$C^j_m = \sum_{k=0}^{2m} \max(S^k_m - S^j_{m-1}, 0)\lambda^k_m = \sum_{k=j+2}^{2m} (S^k_m - S^j_{m-1})\lambda^k_m \tag{56}$$

Similarly, the value of the corresponding put options can be written:

$$P^j_m = \sum_{k=0}^{2m} \max(S^j_{m-1} - S^k_m, 0)\lambda^k_m = \sum_{k=0}^{j} (S^j_{m-1} - S^k_m)\lambda^k_m \tag{57}$$

These relations are valid for $0 \le j \le 2m-2$ (the valid range of j at level $m-1$).

By replacing j with $j-2$ in Equation 56 and solving for λ^j_m, we find:

$$\lambda^j_m = \frac{C^{j-2}_m - \sum_{k=j+1}^{2m}(S^k_m - S^{j-2}_{m-1})\lambda^k_m}{S^j_m - S^{j-2}_{m-1}},\ 2 \le j \le 2m \tag{58}$$

Similarly, solving for λ_m^j in Equation 57 gives:

$$\lambda_m^j = \frac{P_m^j - \sum_{k=0}^{j-1}(S_{m-1}^j - S_m^k)\lambda_m^k}{S_{m-1}^j - S_m^j},\ 0 \le j \le 2m-2 \tag{59}$$

Note that the range of j values covered by the last two formulas do not overlap when $m = 1$. Nevertheless, this level can be handled by using Equation 58 to solve for λ_1^2 and Equation 59 to solve for λ_1^0. Then Equation 12 can be used to conclude that $\lambda_1^1 = d_1 - \lambda_1^0 - \lambda_1^2$.

Arrow-Debreu prices λ_m^j can then be computed for all levels $m > 1$ by using Equation 58 to work downwards from the upper-most state, and Equation 59 to work upwards from the lower-most state. There is some flexibility in deciding where these two (vertical) recursive steps meet; in our implementation we have chosen to use Equation 58 for states $j \ge m$, and Equation 59 is used for states $j < m$.

To complete the analysis we must show how to compute the transition probabilities p_m^j, q_m^j, r_m^j from the already computed trinomial tree of index prices (S_m^j) and Arrow-Debreu prices (λ_m^j). To this end we use the forward relations (Equation 16):

$$\lambda_m^k = d_m[(1 - p_{m-1}^k - q_{m-1}^k)\lambda_{m-1}^k + q_{m-1}^{k-1}\lambda_{m-1}^{k-1} + p_{m-1}^{k-2}\lambda_{m-1}^{k-2}], \tag{60}$$

valid for $0 \le k \le 2m$, where $\lambda_{m-1}^{-1} = \lambda_{m-1}^{-2} = 0$ and $\lambda_{m-1}^{2m} = \lambda_{m-1}^{2m-1} = 0$, and risk-neutral forward relation

$$F_m^k \equiv S_{m-1}^k \frac{q_m}{d_m} = p_{m-1}^k S_m^{k+2} + q_{m-1}^k S_m^{k+1} + (1 - p_{m-1}^k - q_{m-1}^k)S_m^k \tag{61}$$

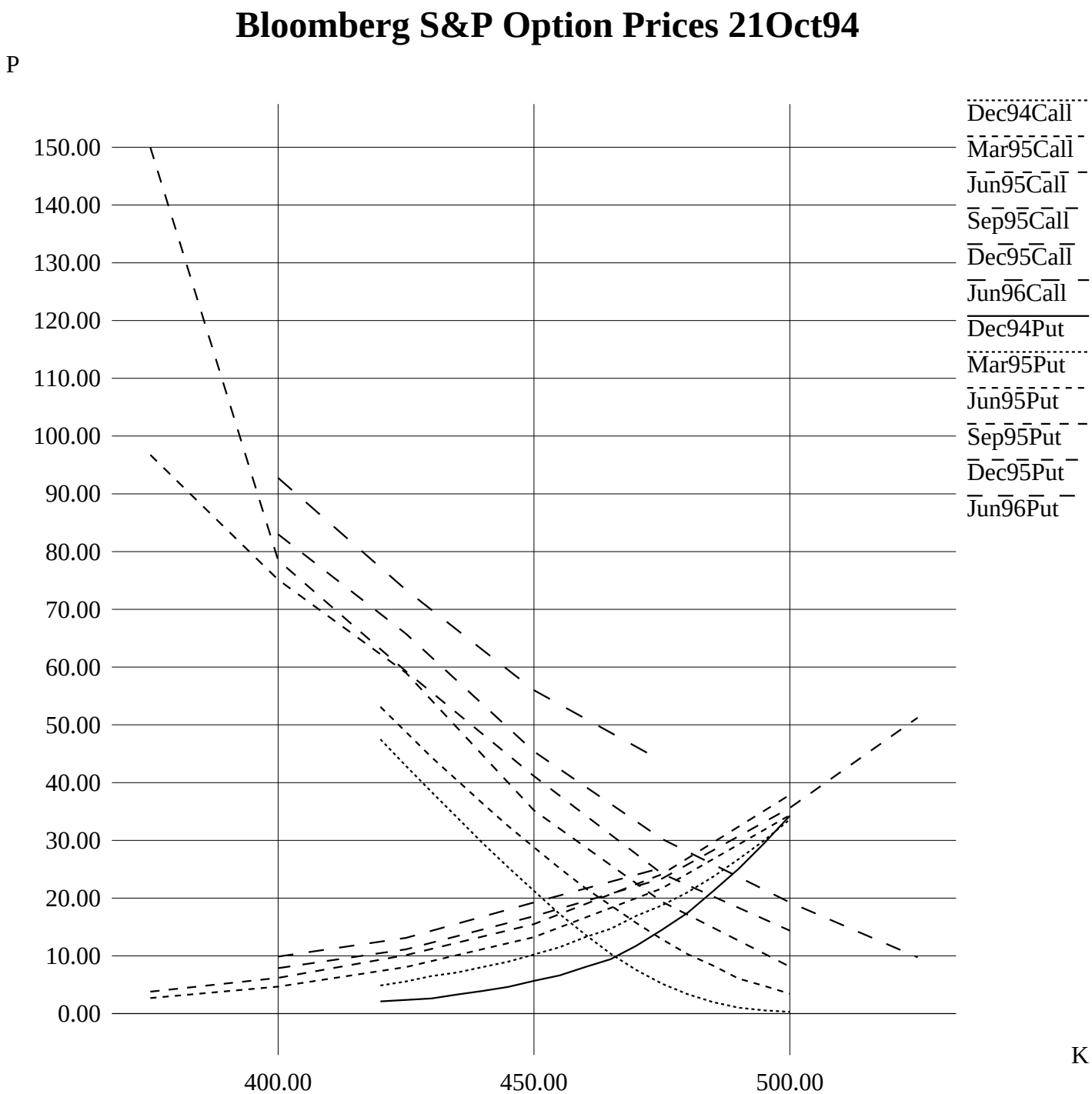
The last two equations constitute a system of two equations in the two unknowns p_{m-1}^j and q_{m-1}^j (recall that $r_{m-1}^j = 1 - p_{m-1}^j - q_{m-1}^j$). The solution is easily found to be:

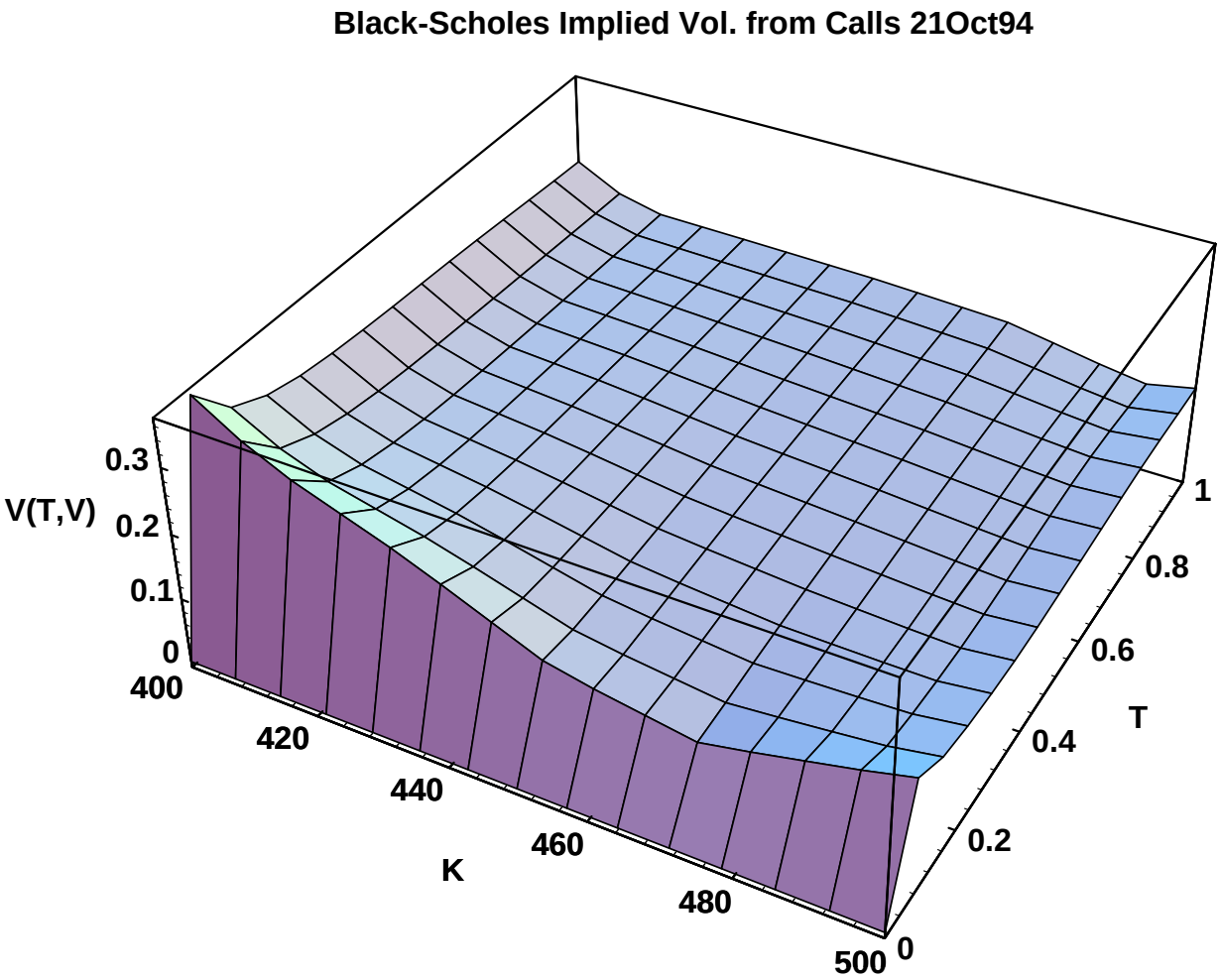
$$r_{m-1}^j = [\frac{\lambda_m^j}{d_m} - q_{m-1}^{j-1}\lambda_{m-1}^{j-1} - p_{m-1}^{j-2}\lambda_{m-1}^{j-2}]/\lambda_{m-1}^j \tag{62}$$

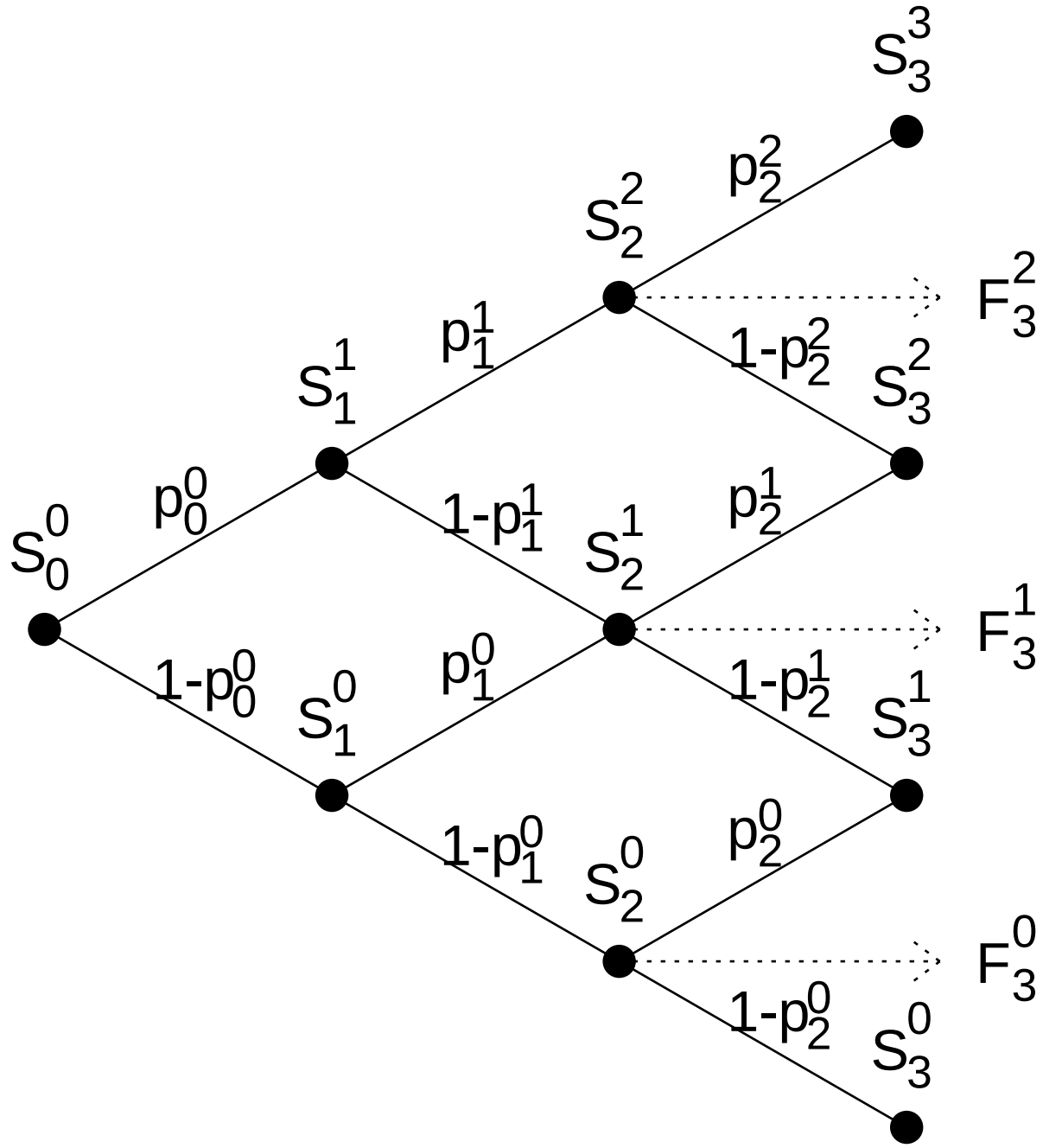
$$p_{m-1}^j = \frac{F_m^j - S_m^j - (1 - r_{m-1}^j)(S_m^{j+1} - S_m^j)}{S_m^{j+2} - S_m^{j+2}} \tag{63}$$

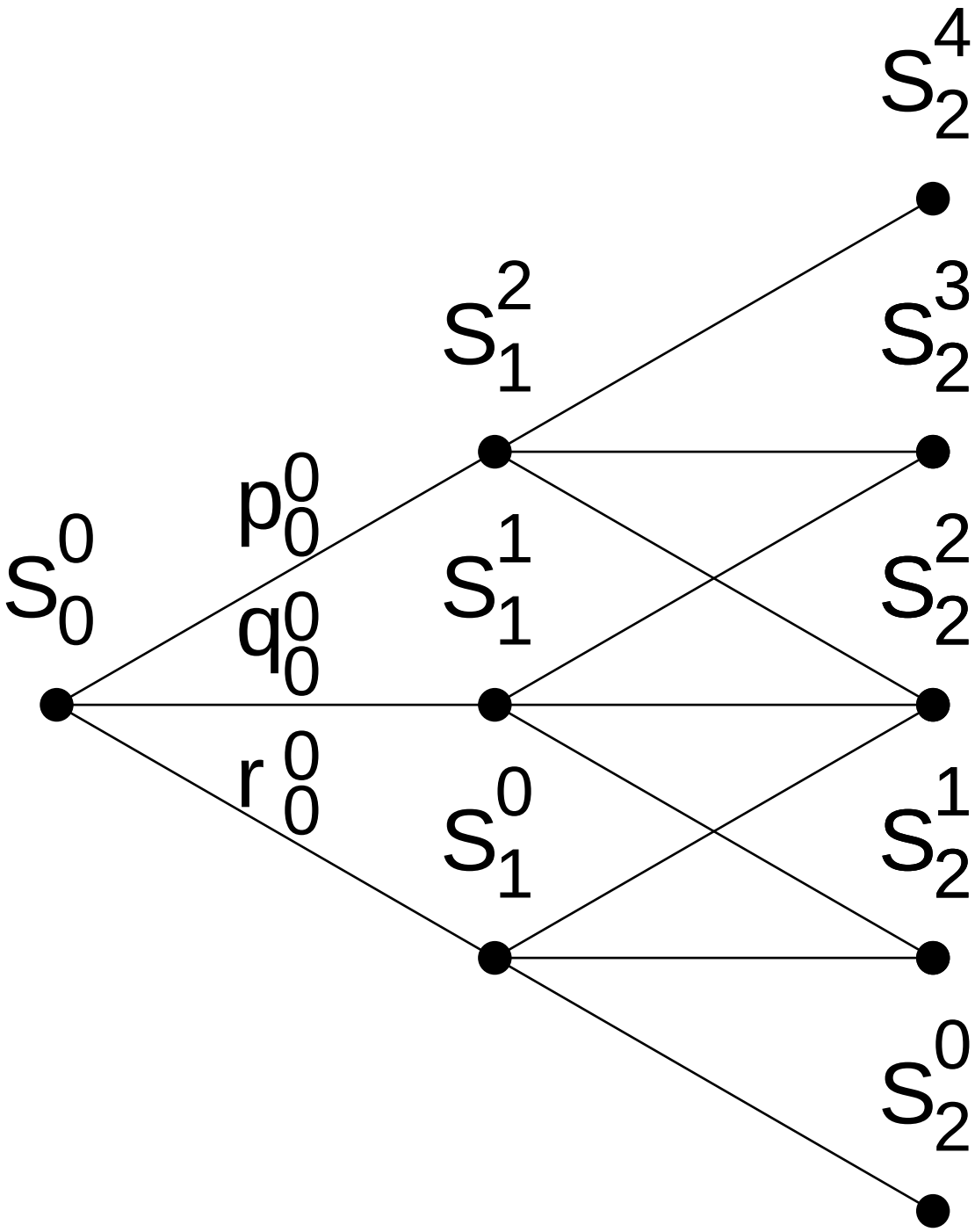
$$q_{m-1}^j = 1 - r_{m-1}^j - p_{m-1}^j \tag{64}$$

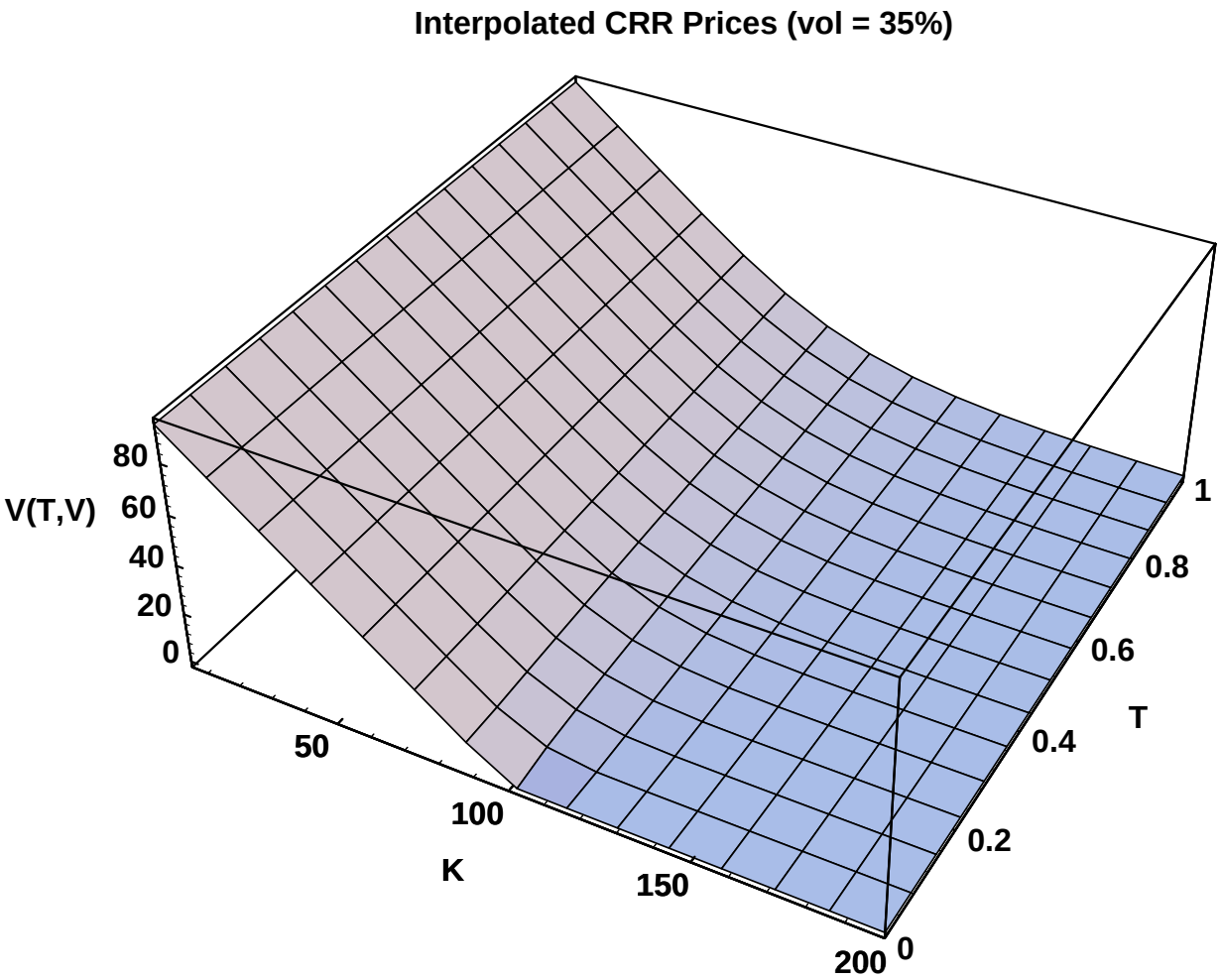
This is valid for $0 \le j \le 2m - 2$.

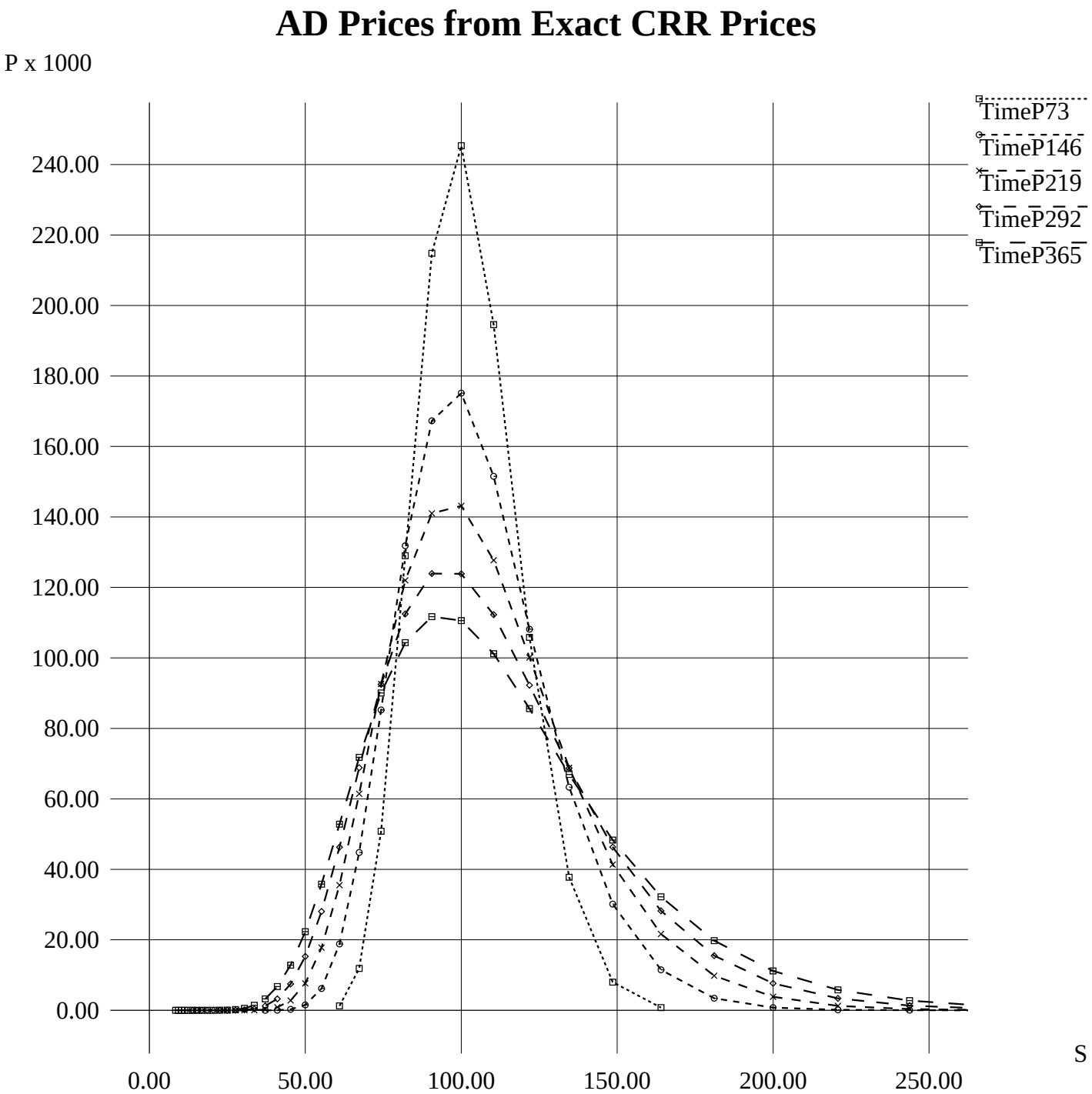

Bloomberg S&P Option Prices 21Oct94
P
150.00
140.00
130.00
120.00
110.00
100.00
90.00
80.00
70.00
60.00
50.00
40.00
30.00
20.00
10.00
0.00
400.00
450.00
500.00
K
Dec94Call
Mar95Call
Jun95Call
Sep95Call
Dec95Call
Jun96Call
Dec94Put
Mar95Put
Jun95Put
Sep95Put
Dec95Put
Jun96Put

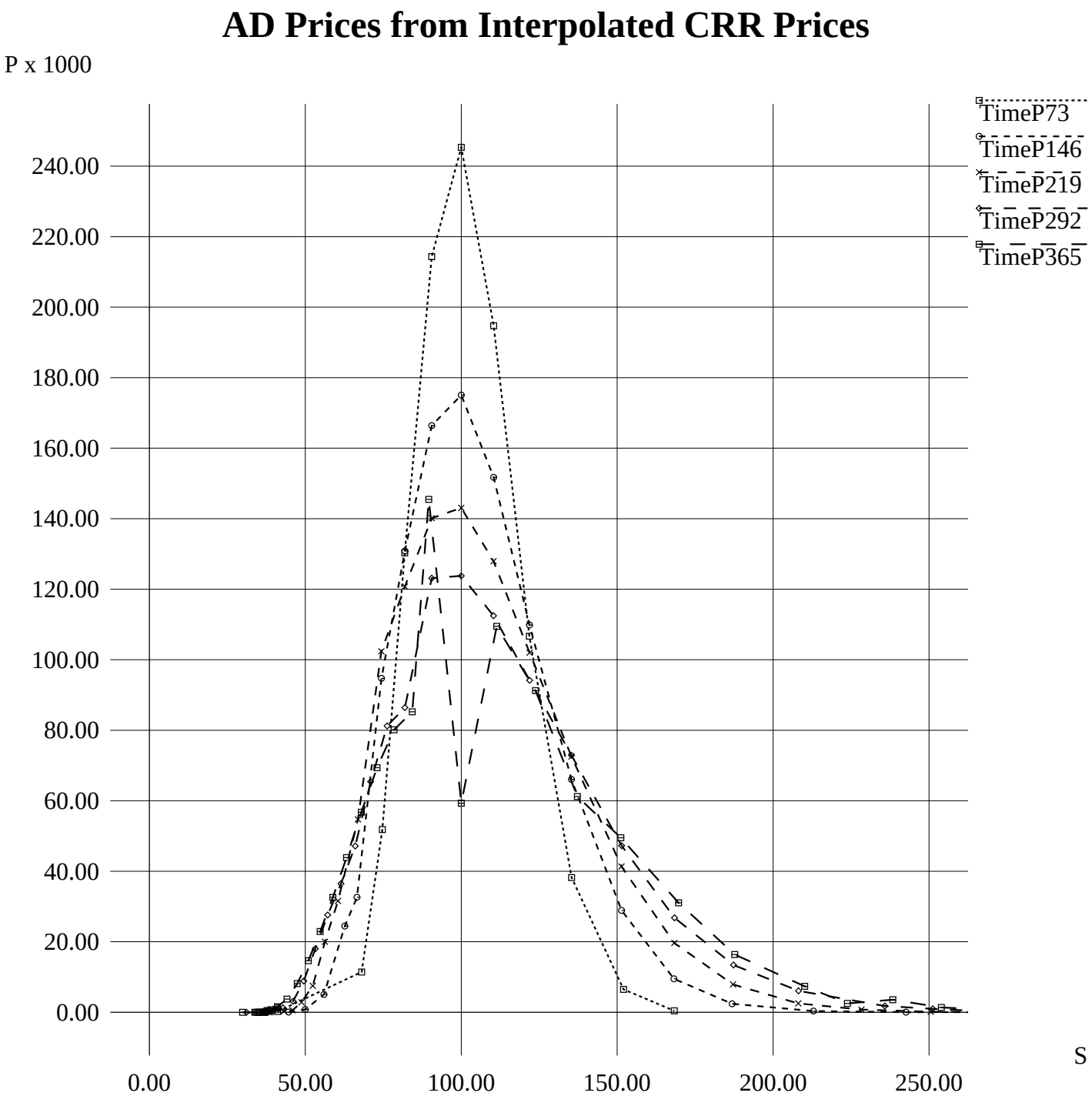

Black-Scholes Implied Vol. from Calls 21Oct94
V(T,V)
0.3
0.2
0.1
0
400
420
440
460
480
500
K
0
0.2
0.4
0.6
0.8
1
T

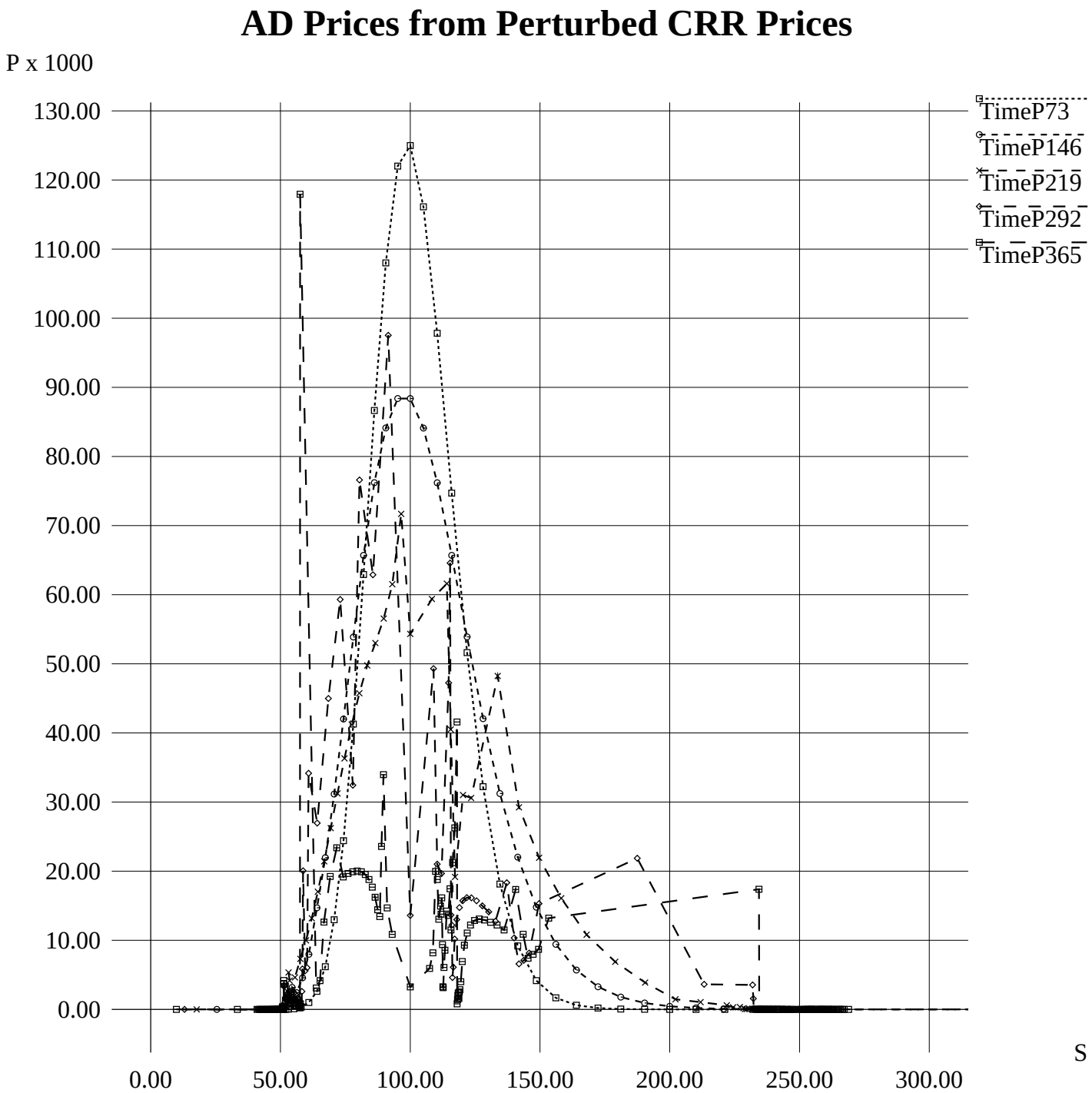

S_3^3
p_2^2
S_2^2
F_3^2
p_1^1
$1-p_2^2$
S_1^1
S_3^2
p_0^0
$1-p_1^1$
p_2^1
S_0^0
S_2^1
F_3^1
$1-p_0^0$
p_1^0
$1-p_2^1$
S_1^0
S_3^1
$1-p_1^0$
p_2^0
S_2^0
F_3^0
$1-p_2^0$
S_3^0

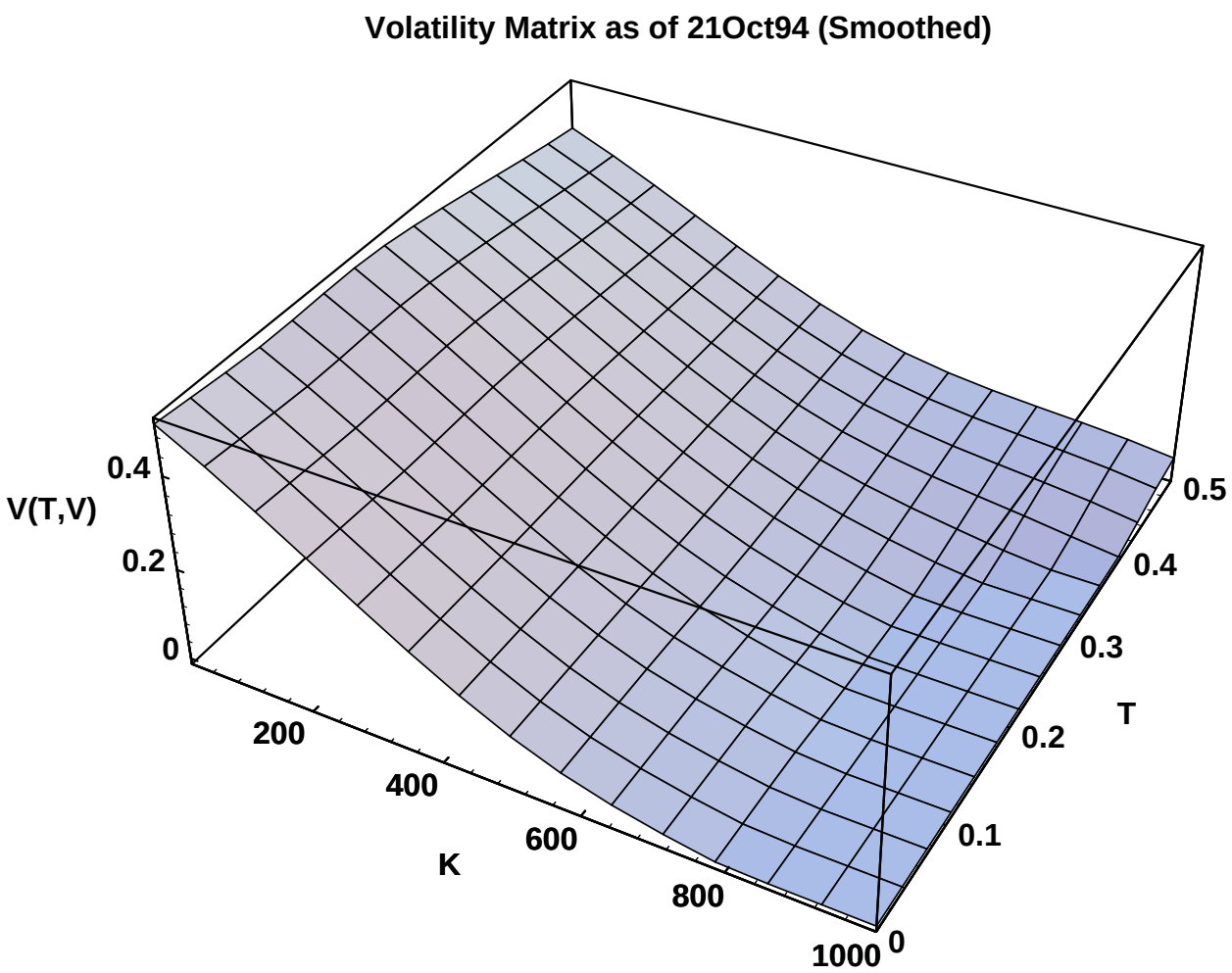

S_0^0
p_0^0
q_0^0
r_0^0
S_1^2
S_1^1
S_1^0
S_2^4
S_2^3
S_2^2
S_2^1
S_2^0

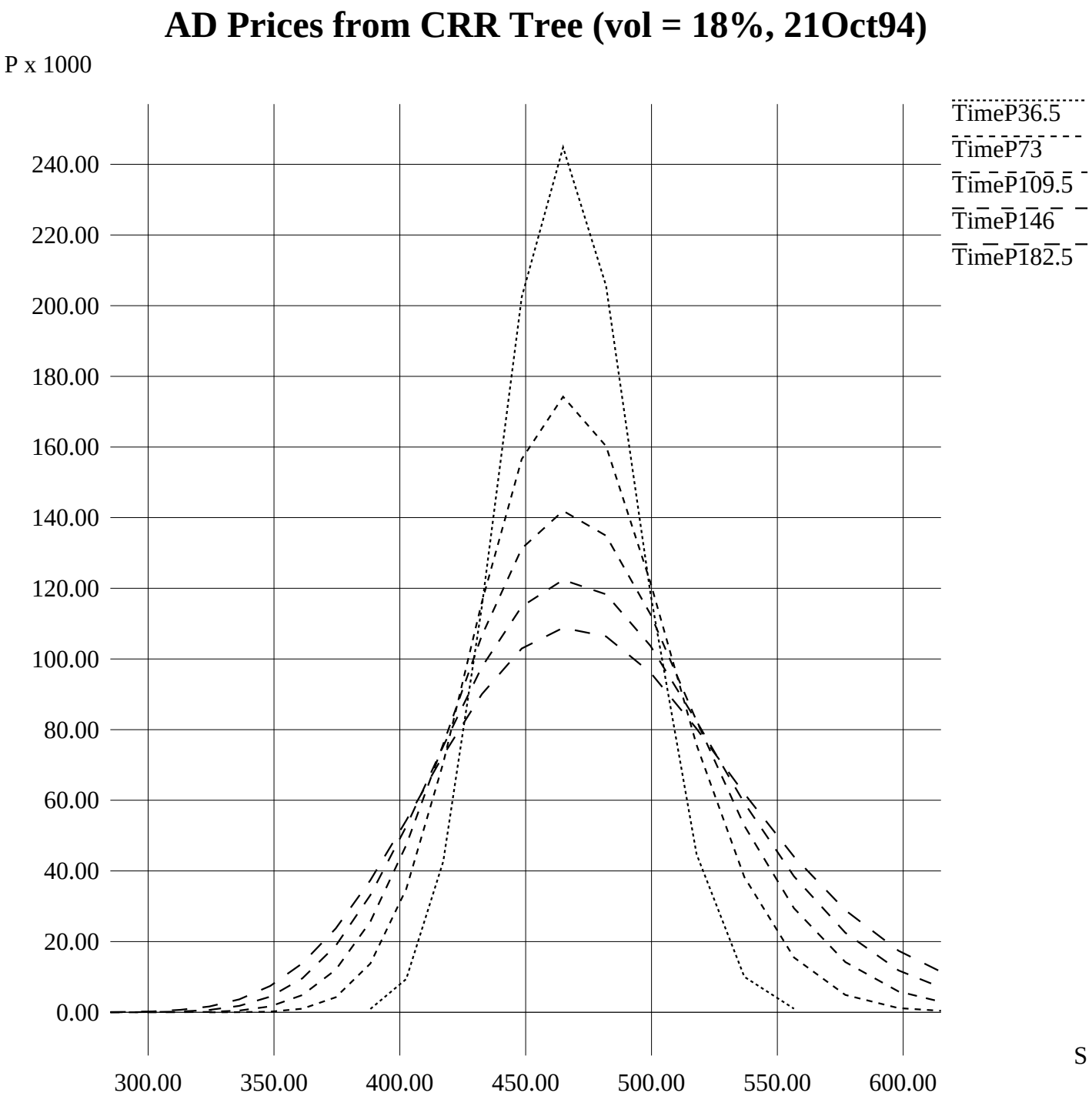

Interpolated CRR Prices (vol = 35%)
V(T,V)
80
60
40
20
0
50
100
150
200
K
0
0.2
0.4
0.6
0.8
1
T

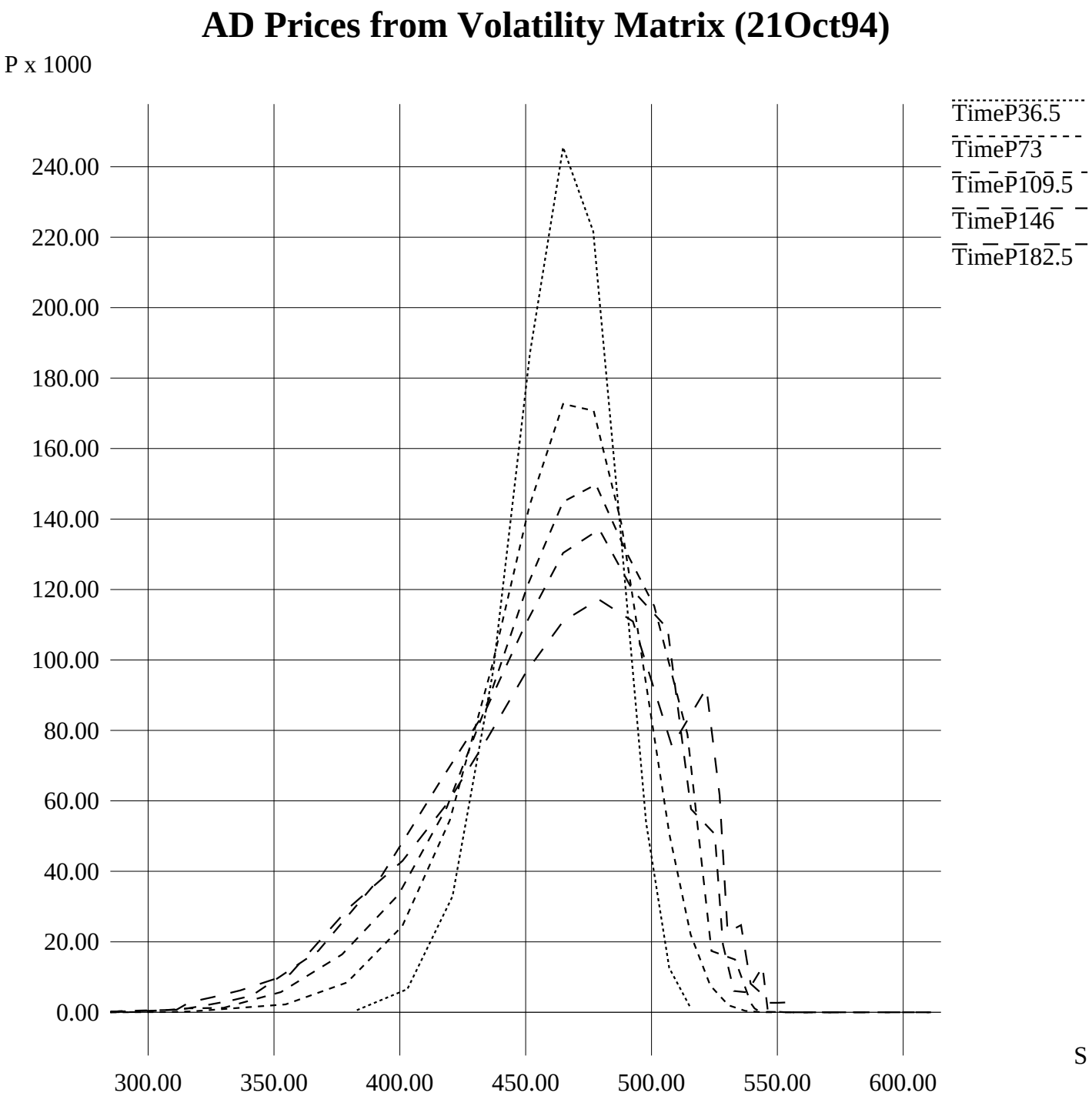

AD Prices from Exact CRR Prices
P x 1000
TimeP73
TimeP146
TimeP219
TimeP292
TimeP365
240.00
220.00
200.00
180.00
160.00
140.00
120.00
100.00
80.00
60.00
40.00
20.00
0.00
0.00
50.00
100.00
150.00
200.00
250.00
S

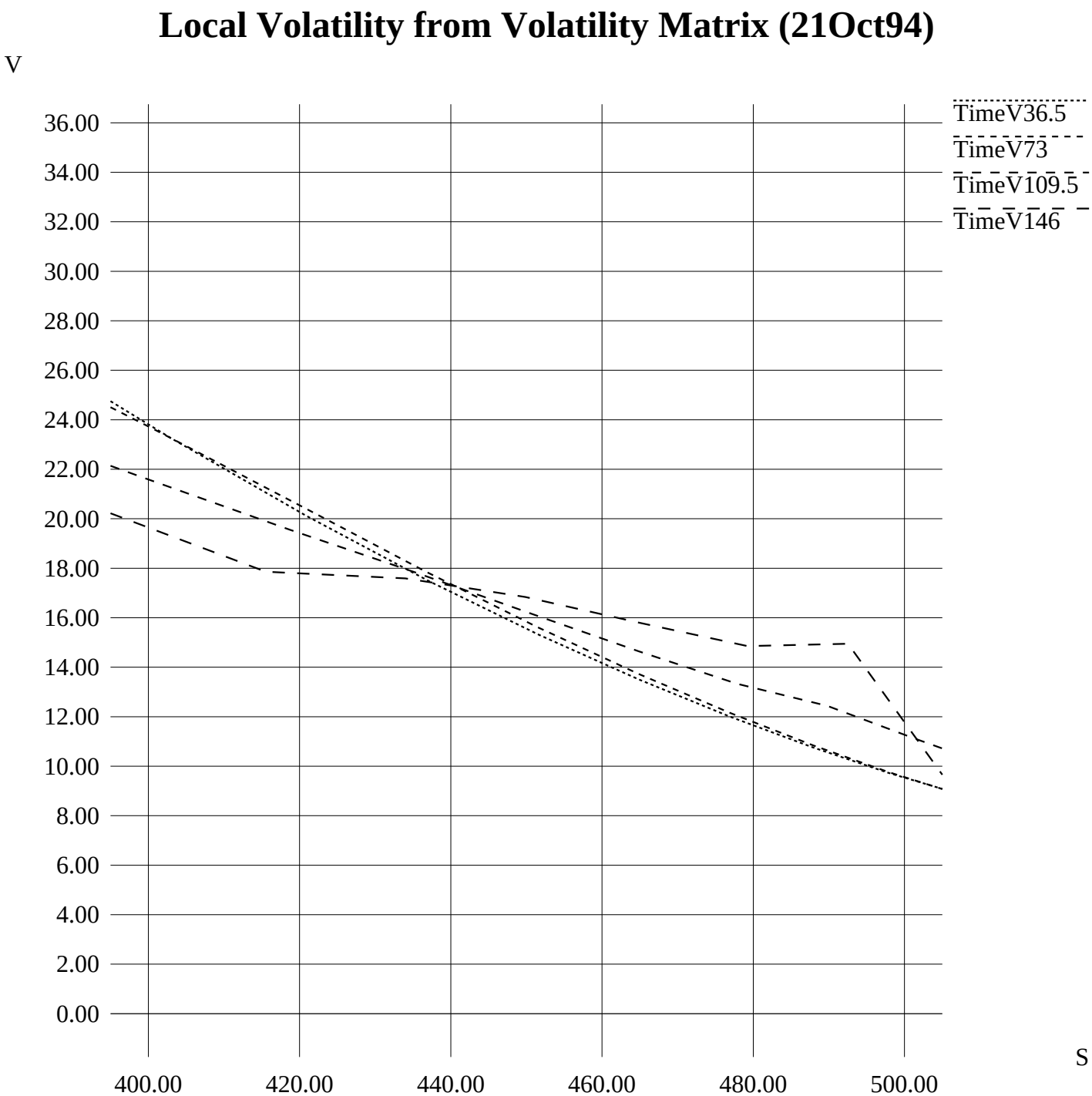

AD Prices from Interpolated CRR Prices
P x 1000
TimeP73
TimeP146
TimeP219
TimeP292
TimeP365
240.00
220.00
200.00
180.00
160.00
140.00
120.00
100.00
80.00
60.00
40.00
20.00
0.00
0.00
50.00
100.00
150.00
200.00
250.00
S

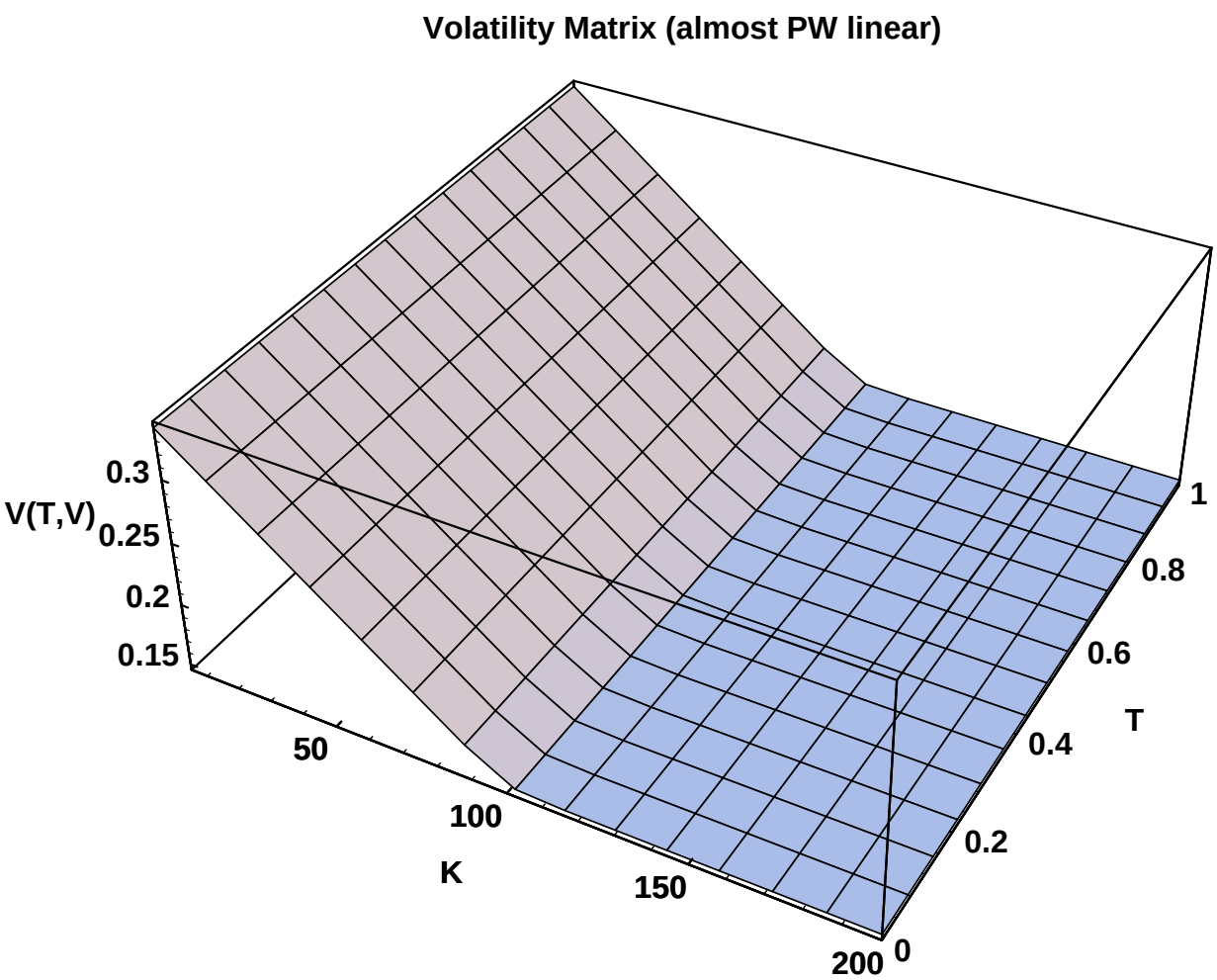

AD Prices from Perturbed CRR Prices
P x 1000
130.00
120.00
110.00
100.00
90.00
80.00
70.00
60.00
50.00
40.00
30.00
20.00
10.00
0.00
0.00
50.00
100.00
150.00
200.00
250.00
300.00
S
TimeP73
TimeP146
TimeP219
TimeP292
TimeP365


Volatility Matrix as of 21Oct94 (Smoothed)
V(T,V)
0.4
0.2
0
200
400
600
800
1000
K
0
0.1
0.2
0.3
0.4
0.5
T


AD Prices from CRR Tree (vol = 18%, 21Oct94)
P x 1000
240.00
220.00
200.00
180.00
160.00
140.00
120.00
100.00
80.00
60.00
40.00
20.00
0.00
300.00
350.00
400.00
450.00
500.00
550.00
600.00
TimeP36.5
TimeP73
TimeP109.5
TimeP146
TimeP182.5
S

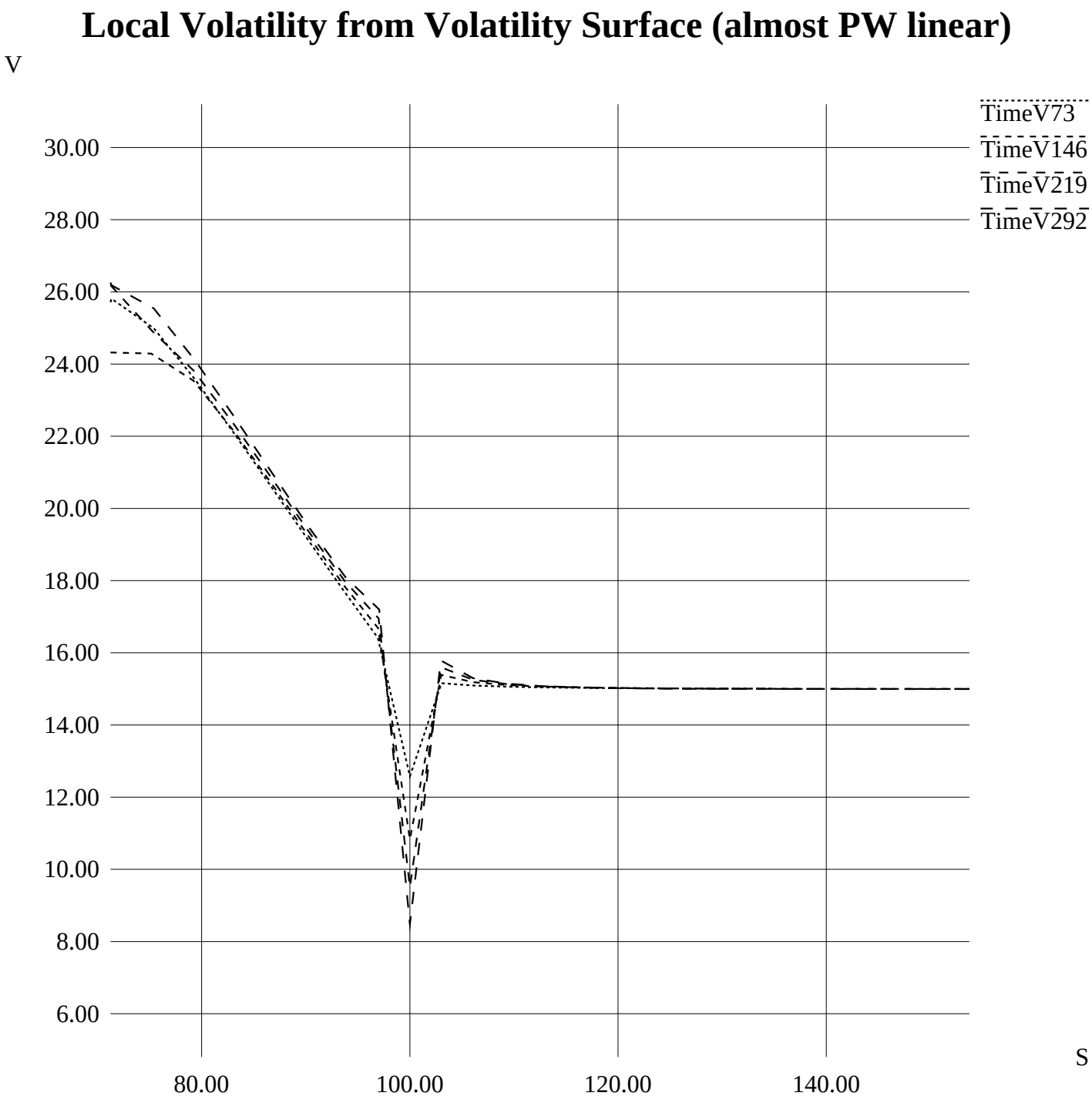

AD Prices from Volatility Matrix (21Oct94)
P x 1000
240.00
220.00
200.00
180.00
160.00
140.00
120.00
100.00
80.00
60.00
40.00
20.00
0.00
300.00
350.00
400.00
450.00
500.00
550.00
600.00
S
TimeP36.5
TimeP73
TimeP109.5
TimeP146
TimeP182.5

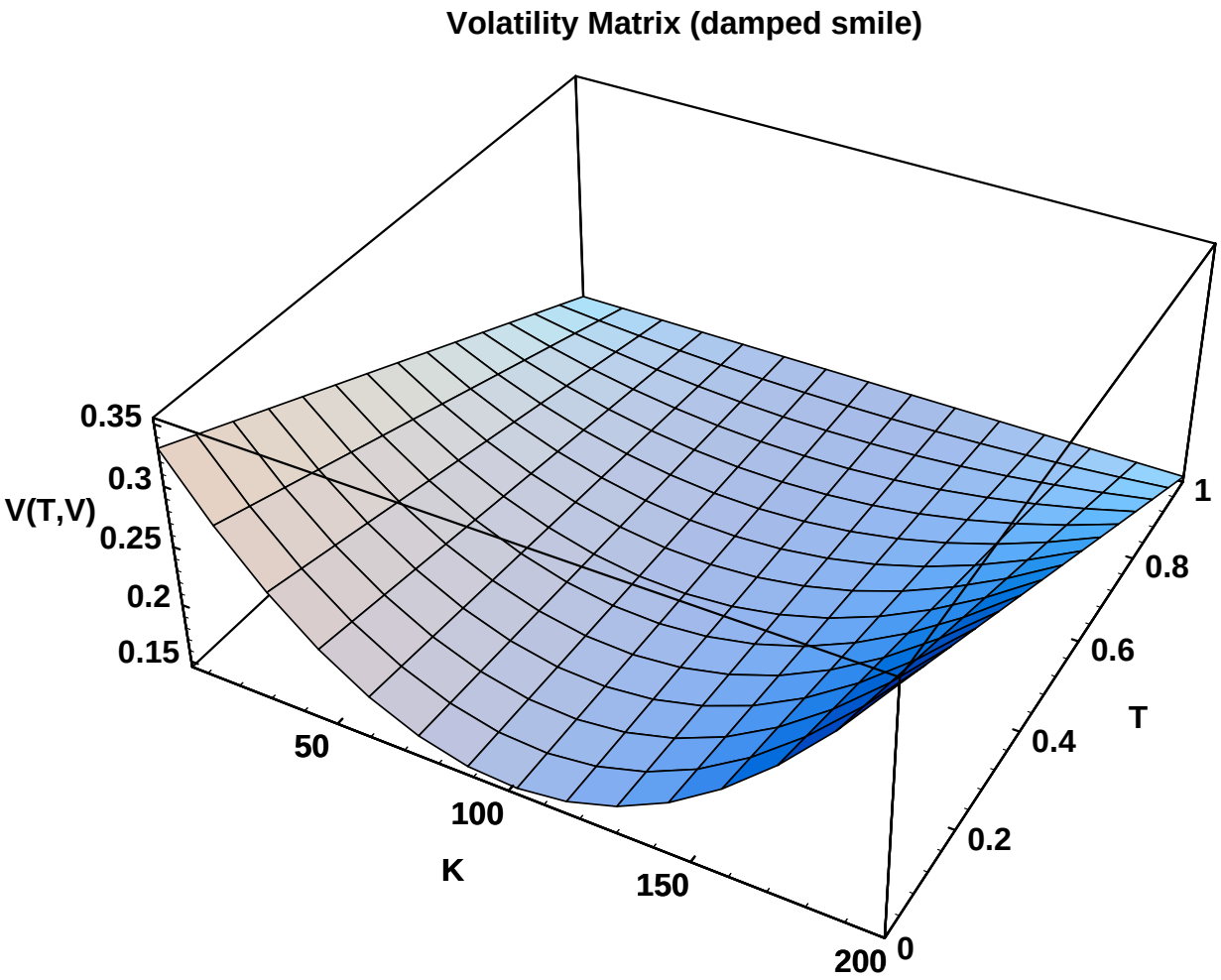

Local Volatility from Volatility Matrix (21Oct94)
V
36.00
34.00
32.00
30.00
28.00
26.00
24.00
22.00
20.00
18.00
16.00
14.00
12.00
10.00
8.00
6.00
4.00
2.00
0.00
400.00
420.00
440.00
460.00
480.00
500.00
S
TimeV36.5
TimeV73
TimeV109.5
TimeV146

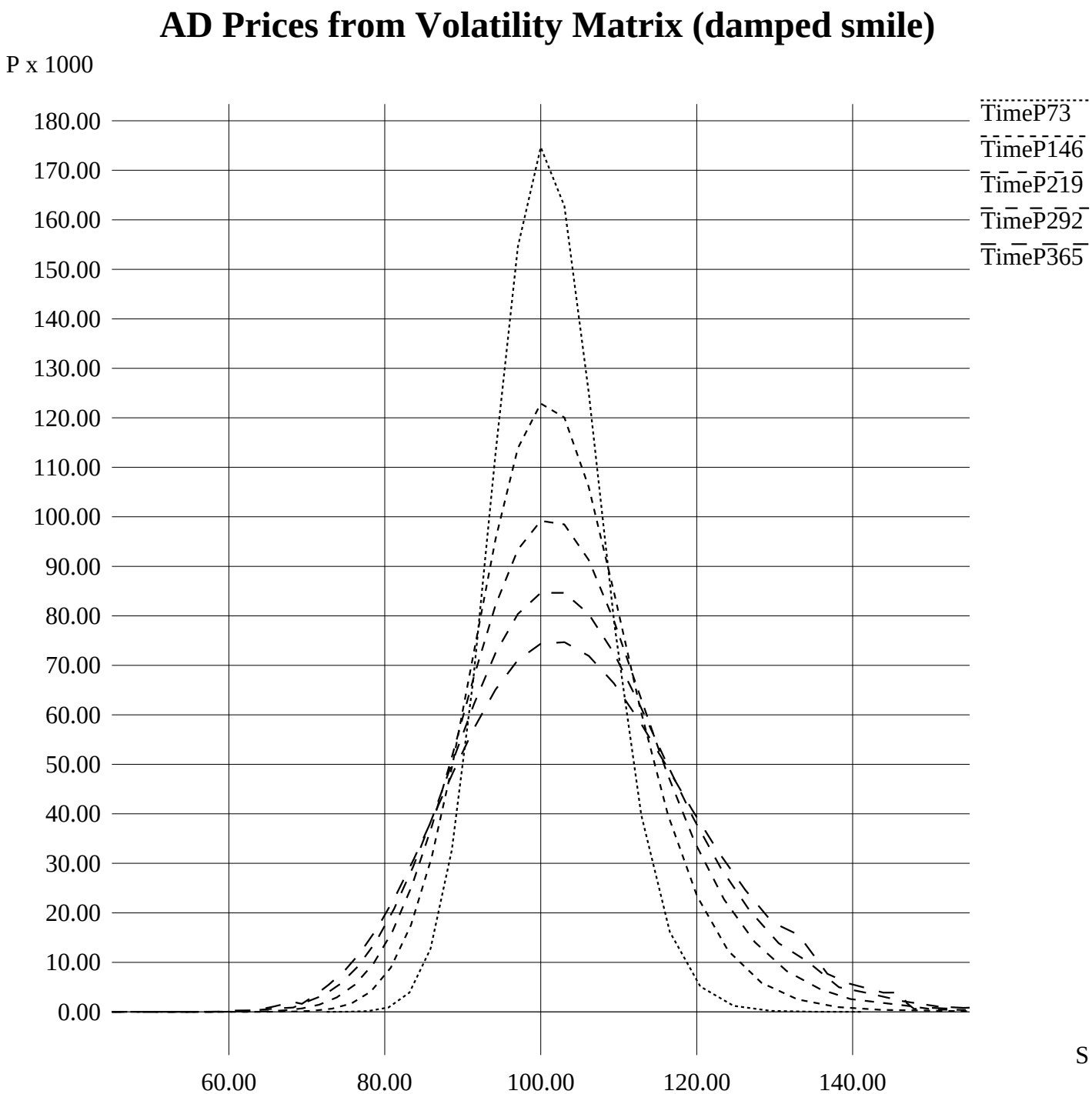

Volatility Matrix (almost PW linear)
V(T,V)
0.3
0.25
0.2
0.15
50
100
150
200
K
0
0.2
0.4
0.6
0.8
1
T

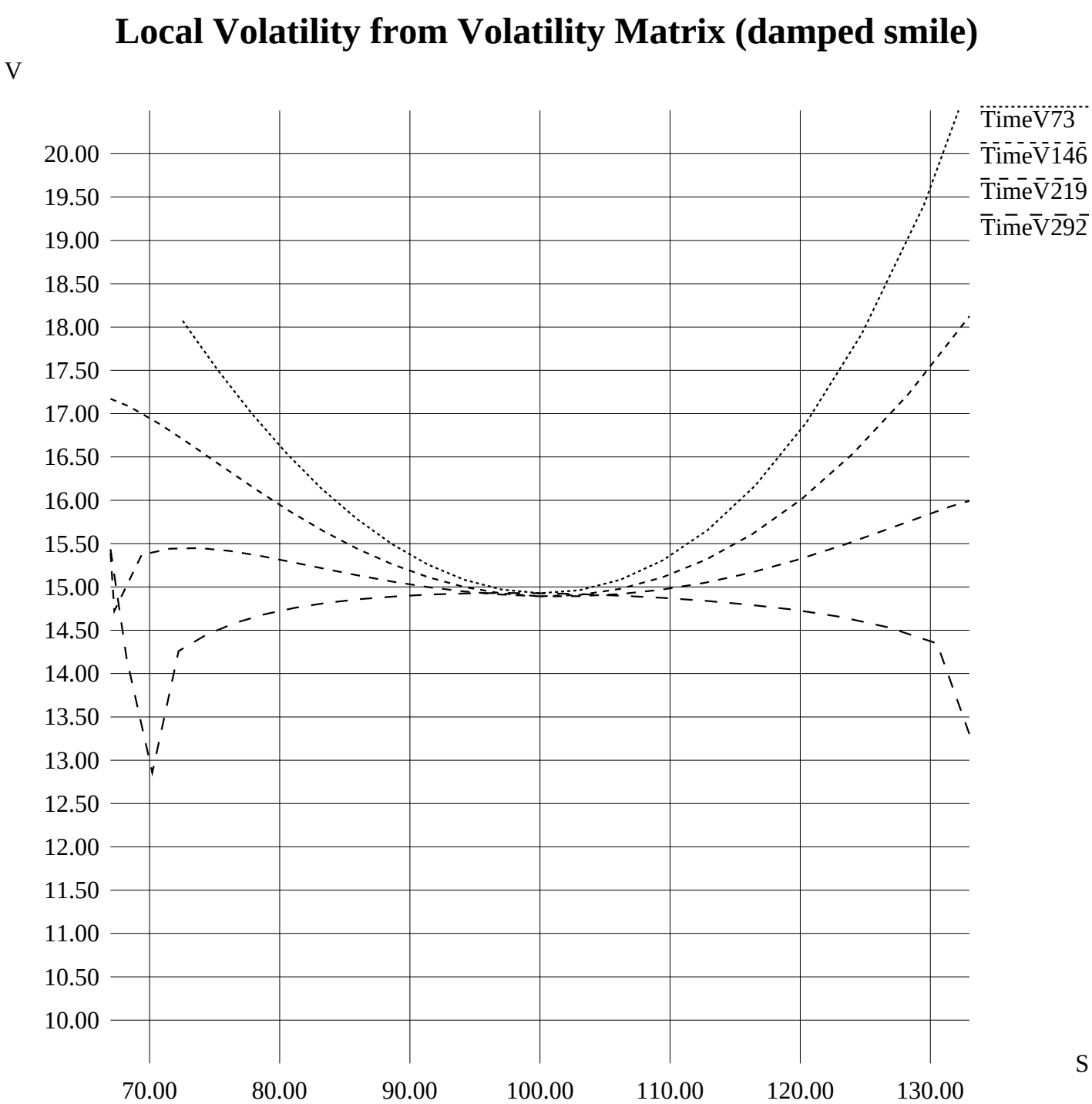
includegraphicsv10ad.pdf

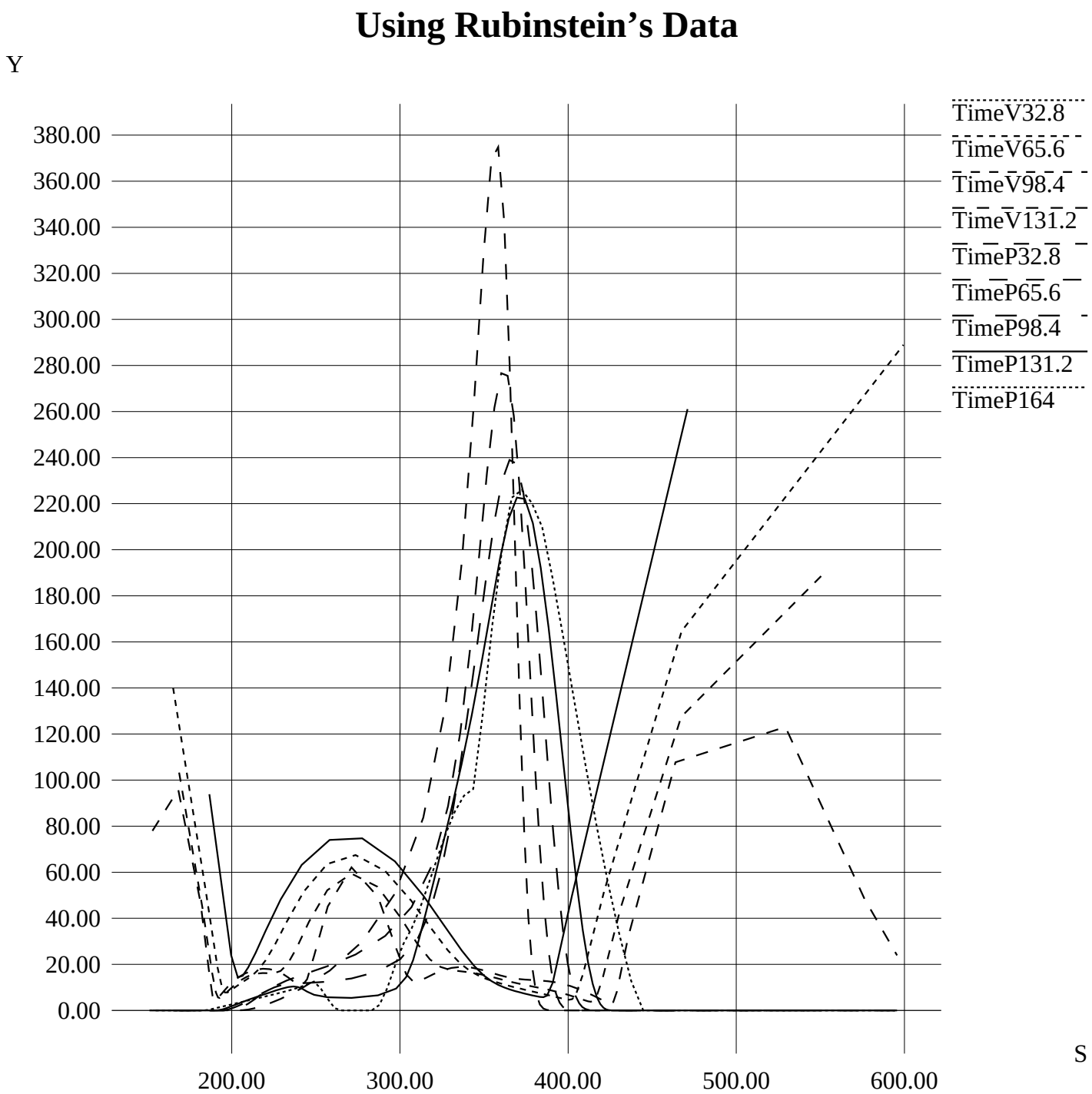
Figure 14: AD prices from volatility matrix (almost PW linear).

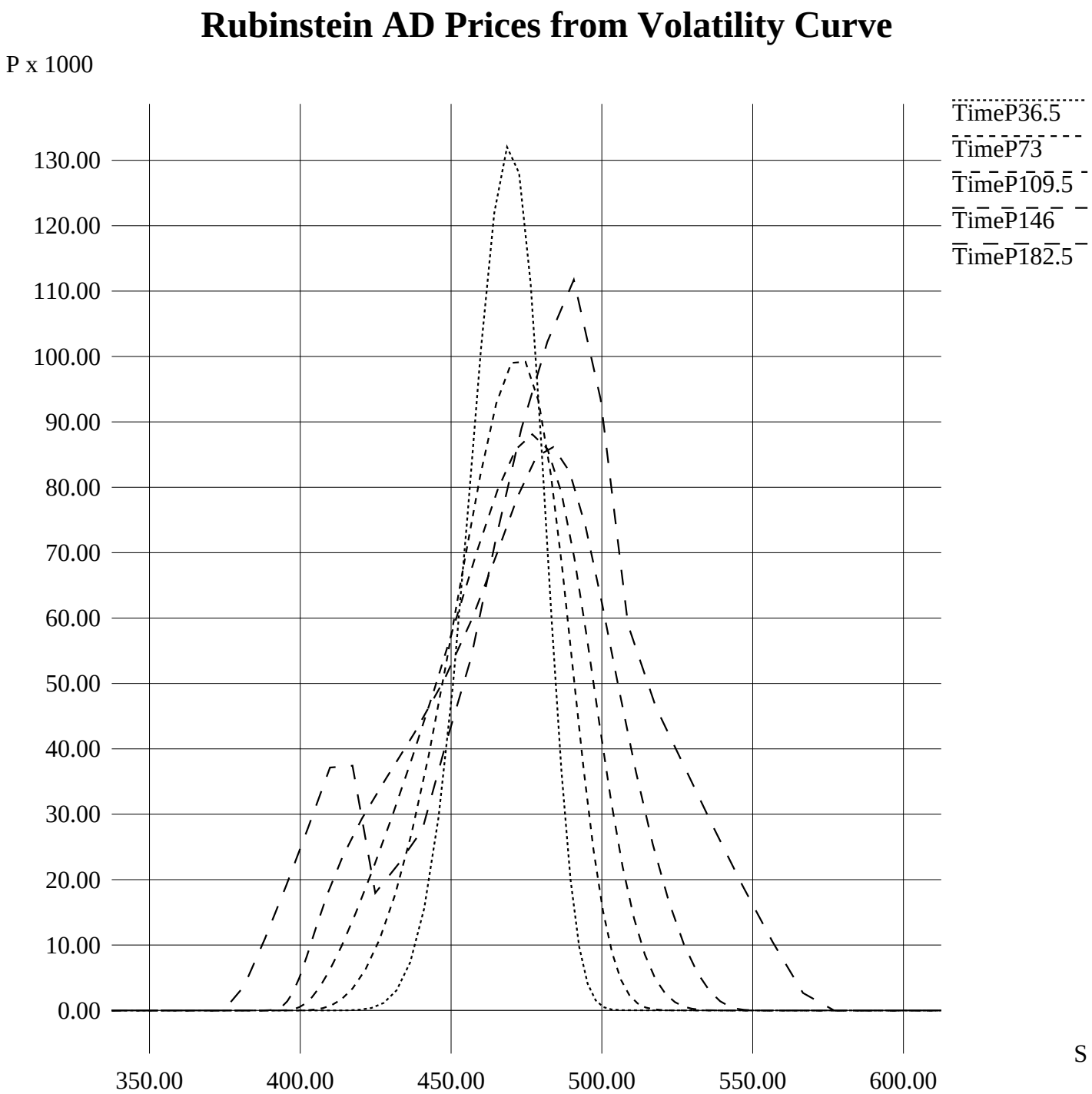

Local Volatility from Volatility Surface (almost PW linear)
V
TimeV73
TimeV146
TimeV219
TimeV292
30.00
28.00
26.00
24.00
22.00
20.00
18.00
16.00
14.00
12.00
10.00
8.00
6.00
80.00
100.00
120.00
140.00
S

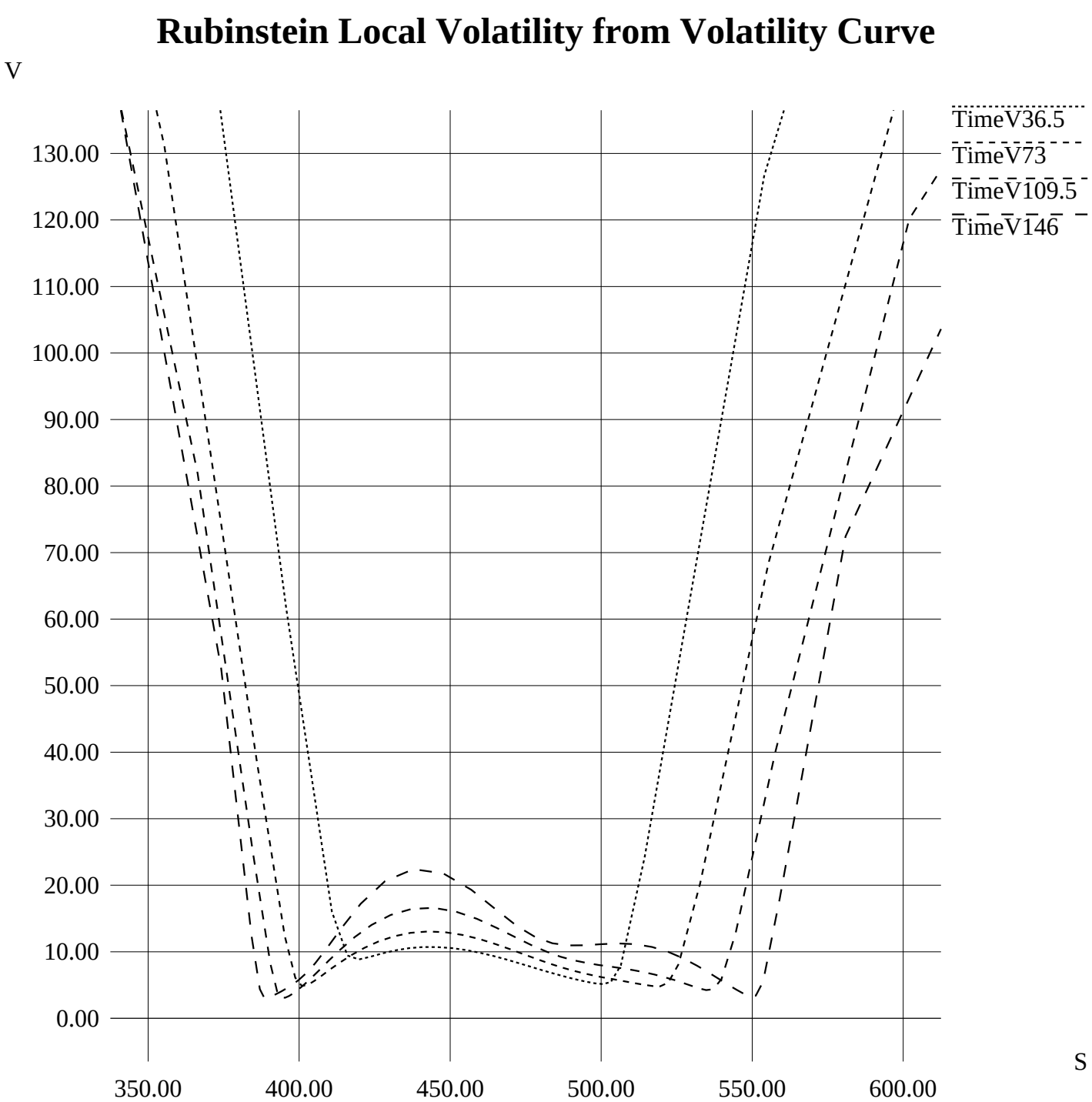

Volatility Matrix (damped smile)
V(T,V)
0.35
0.3
0.25
0.2
0.15
50
100
150
200
K
0
0.2
0.4
0.6
0.8
1
T


AD Prices from Volatility Matrix (damped smile)
P x 1000
180.00
170.00
160.00
150.00
140.00
130.00
120.00
110.00
100.00
90.00
80.00
70.00
60.00
50.00
40.00
30.00
20.00
10.00
0.00
60.00
80.00
100.00
120.00
140.00
S
TimeP73
TimeP146
TimeP219
TimeP292
TimeP365


Local Volatility from Volatility Matrix (damped smile)
V
TimeV73
TimeV146
TimeV219
TimeV292
20.00
19.50
19.00
18.50
18.00
17.50
17.00
16.50
16.00
15.50
15.00
14.50
14.00
13.50
13.00
12.50
12.00
11.50
11.00
10.50
10.00
70.00
80.00
90.00
100.00
110.00
120.00
130.00
S


Using Rubinstein's Data
Y
380.00
360.00
340.00
320.00
300.00
280.00
260.00
240.00
220.00
200.00
180.00
160.00
140.00
120.00
100.00
80.00
60.00
40.00
20.00
0.00
200.00
300.00
400.00
500.00
600.00
S
TimeV32.8
TimeV65.6
TimeV98.4
TimeV131.2
TimeP32.8
TimeP65.6
TimeP98.4
TimeP131.2
TimeP164


Rubinstein AD Prices from Volatility Curve
P x 1000
130.00
120.00
110.00
100.00
90.00
80.00
70.00
60.00
50.00
40.00
30.00
20.00
10.00
0.00
350.00
400.00
450.00
500.00
550.00
600.00
S
TimeP36.5
TimeP73
TimeP109.5
TimeP146
TimeP182.5


Rubinstein Local Volatility from Volatility Curve
V
130.00
120.00
110.00
100.00
90.00
80.00
70.00
60.00
50.00
40.00
30.00
20.00
10.00
0.00
350.00
400.00
450.00
500.00
550.00
600.00
S
TimeV36.5
TimeV73
TimeV109.5
TimeV146

References

[1] Kenneth J. Arrow. The role of securities in the optimal allocation of risk-bearing. *Review of Economic Studies*, 31:91–96, 1964.

[2] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81:637–654, May/June 1973.

[3] Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31:307–327, 1986.

[4] Phelim P. Boyle. A lattice framework for options pricing with two state variables. *Journal of Financial and Quantitative Analysis*, 23(1), March 1988.

[5] Alan Brace and Marek Musiela. A multifactor Gauss Markov implementation of Heath, Jarrow, and Morton. *Mathematical Finance*, 4(3):259–283, 1994.

[6] John C. Cox and Stephen A. Ross. The valuation of options for alternative stochastic processes. *Journal of Financial Economics*, 3:145–166, 1976.

[7] John C. Cox, Stephen A. Ross, and Mark Rubinstein. Option pricing: A simplified approach. *Journal of Financial Economics*, 7:229–263, 1979.

[8] John C. Cox and Mark Rubinstein. *Options Markets*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1985.

[9] Michel Crouhy and Dan Galai. Hedging with a volatility term structure. *The Journal of Derivatives*, pages 45–52, Spring 1995.

[10] Gerard Debreu. *Theory of Value*. Yale University Press, Ltd., London, 1959.

[11] Emanuel Derman, Deniz Ergener, and Iraj Kani. Forever hedged. *Risk*, 7(9), September 1994.

[12] Emanuel Derman and Iraj Kani. Riding on a smile. *Risk*, 7(2), February 1994.

[13] Darrel Duffie. *Dynamic Asset Pricing Theory*. Princeton University Press, 1992.

[14] Bruno Dupire. Model art. *Risk*, 6(9), September 1993.

[15] Bruno Dupire. Pricing with a smile. *Risk*, 7(1), January 1994.

[16] Robert F. Engle and C. W. J. Granger. Co-integration and error correction: Representation, Estimation, and Testing. *Econometrica*, 55(2):251–276, March 1987.

[17] Wendell H. Fleming and Raymond W. Rishel. *Deterministic and Stochastic Optimal Control*. Springer-Verlag, New York, 1975.

[18] J. Michael Harrison and David M. Kreps. Martingales and arbitrage in multiperiod securities markets. *Journal of Economic Theory*, 20:381–408, 1979.

[19] Hua He and Hayne Leland. On equilibrium asset price processes. *The Review of Financial Studies*, 6(3):593–617, 1993.

[20] David Heath, Robert Jarrow, and Andrew Morton. Bond pricing and the term structure of interest rates: A discrete time approximation. *Journal of Financial and Quantitative Analysis*, 25(4):419–440, December 1990.

[21] David Heath, Robert Jarrow, and Andrew Morton. Bond pricing and the term structure of interest rates: A new methodology for contingent claims valuation. *Econometrica*, 60(1):77–105, January 1992.

[22] John Hull and Alan White. The pricing of options on assets with stochastic volatilities. *Journal of Finance*, 42(2):281–300, June 1987.

[23] Farshid Jamshidian. Forward induction and construction of yield curve diffusion models. *Journal of Fixed Income*, 1(1):62–74, June 1991.

[24] Fritz John. *Partial Differential Equations*. Springer Verlag, New York, third edition, 1978.

[25] Ioannis Karatzas and Steven E. Shreve. *Brownian Motion and Stochastic Calculus*. Springer-Verlag, New York, 1988.

[26] Michael Kelly. Stock answer. *Risk*, 7(8), August 1994.

[27] R. S. Liptser and A. N. Shiryayev. *Statistics of Random Processes I: General Theory*. Springer Verlag, New York, 1977.

[28] Bernt Øksendal. *Stochastic Differential Equations*. Springer-Verlag, New York, third edition, 1992.

[29] Kurt S. Riedel. Optimal estimation of dynamically evolving diffusivities. *Journal of Computational Physics*, 115:1–11, 1994.

[30] Mark Rubinstein. Implied binomial trees. *Journal of Finance*, 69(3):771–818, July 1994.

[31] Mark Rubinstein. As simple as one, two, three. *Risk*, 8(1), January 1995.

[32] Dominick Samperi. On barrier option pricing. Citibank Global Derivatives Research Report, March 1995.

[33] David Shimko. A tail of two distributions. *Risk*, 7(9), September 1994.

[34] Petar D. Simic. Statistical mechanics as the underlying theory of 'elastic' and 'neural' optimisations. *Network*, 1:89–103, 1990.

[35] Bernard Widrow. 30 years of adaptive neural networks: Perceptron, Madaline, and Backpropagation. *Procedings of the IEEE*, 78(9):1415–1442, September 1990.

Index