Bachelor's Thesis presented to obtain the Bachelor Degree

Implied Trinomial Trees and Their Implementation with XploRe

by

Karel Komorád

(180174)

Submitted to:

Prof. Dr. Wolfgang Härdle

Institute for Statistics and Econometrics

Humboldt University

Spandauer Str. 1

D–10178 *Berlin*

Berlin, April 4, 2002

Declaration of Authorship

I hereby confirm that I have authored this bachelor thesis independently and without use of others than the indicated resources.

All passages, which are literally or in general matter taken out of publications or other resources, are marked as such.

Karel Komorád

Berlin, April 4, 2002

Abstract

This bachelor thesis describes and summarizes the method of implied trinomial trees as introduced in (Derman, Kani and Chriss, 1996) and provides a detailed guide through its use in the statistical software XploRe. This method is an option pricing technique which tries to fit the market volatility smile. It uses an inductive algorithm constructing a possible evolution process of underlying prices from the current market option data. All XploRe functions (the so called **quantlets**) handling the implied trinomial trees, the executing as well as the output quantlets, are presented thoroughly in the last part which can be thus regarded as the manual pages of implied trinomial trees in XploRe.

Contents

1 Introduction **4**

2 Basic Option Pricing Overview **6**

3 Trees and Implied Models **8**

4 ITT's and Their Construction **10**

4.1 Basic insight 10

4.2 State space 12

4.3 Transition probabilities 14

4.4 Possible pitfalls 15

4.5 Illustrative examples 16

5 ITT's in XploRe **21**

5.1 Basic skills 21

5.2 Advanced features 27

5.3 What is hidden 30

Appendix **32**

A List of Quantlets **32**

B Derivation of the transition probabilities **33**

1 Introduction

In recent decades option-like contracts have been through a giant evolution. Options are financial derivatives speculating on the value of an underlying asset. The boom in research on options and the use of them started after Black and Scholes published the formula for options prices. Options are maybe more common than one could think — many issues of corporate securities (e.g. bonds or stocks) also have option features. New financing techniques, for instance, contingent value rights (CVR's), are straightforward applications of options. Liquidly traded options are important financial instruments used for **hedging** — they can be included into the portfolio to reduce risk. So option pricing has become one of the basic techniques in finance.

Unfortunately, option prices computed by the Black-Scholes formula and the market options prices show a discrepancy. This deviation may be due to the assumption

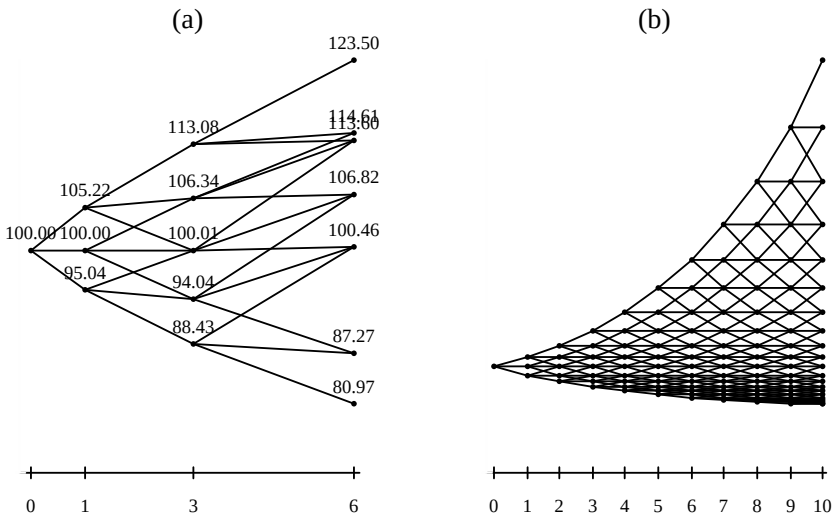

Figure 1: Examples of State Spaces of Two Different Implied Trinomial Trees.

of constant volatility in the Black-Scholes model which is hardly ever fulfilled in the market. Therefore new approaches were examined to find options prices consistent with the market. Binomial trees, discrete version of the Black-Scholes model, are intuitive and probably the most commonly used basic procedure for option pricing. Derman and Kani (1994) presented an extension of binomial trees called implied binomial trees that fit the market data. Implied trinomial trees (ITT) are analogical extension of trinomial trees. Trinomial trees converge in the continuous limit to the same result

as binomial trees, but the use of ITT's give us a freedom in the choice of the possible path of the underlying price, the so called **state space**. This freedom could be in some circumstances desirable. Figure 1(a) shows a possible state space with three periods over six years. An ITT with one-year time steps over ten years is depicted in figure 1(b).

This bachelor thesis summarizes traditional option pricing concepts and recalls basic terms in section 2. Section 3 explains the implied tree theories before section 4 describes the construction of an ITT. For a deeper insight into the problem of ITT's, Derman, Kani and Chriss (1996) offers a more elaborate explanation. The last section is a manual of the use of implied trinomial trees in the statistical computing environment XploRe. There are all new quantlets listed with a short description in appendix A. Appendix B shows in detail derivation of equation for the transition probabilities. As a supplement to this paper a compact disc containing MD*Book with this thesis as well as all quantlets dealing implied trinomial trees is enclosed.

2 Basic Option Pricing Overview

Options are contingent claims on the value of an underlying asset, usually a stock or a traded market index. The simplest type of options are **European options**. Their importance is mainly theoretical. An European **call** (or **put**) gives the owner the right to buy (or sell) an underlying asset at a fixed price (the so called **strike** price) at a given date.

The famous Black-Scholes model of option pricing can be viewed as a continuous generalization of duplication model. For an explanation of the duplication model see (Ross, Westerfield and Jaffe, 2002). Black and Scholes assume that the underlying asset follows a geometric Brownian motion with a constant volatility and is described by the stochastic differential equation:

$$\frac{dS_t}{S_t} = \mu dt + \sigma dZ_t \tag{1}$$

Where S denotes the underlying price, μ is the expected return and Z stands for the standard Wiener process. As a consequence, the underlying distribution is lognormal. This is the crucial point — volatility is the only parameter of the Black-Scholes formula which is not explicitly observable from the market data. When plugging the market option prices into the formula and solving for volatility (the so called **implied volatility**), empirical data shows a variation with both exercise (the **skew** structure) and expiration time (the **term** structure), both together are commonly called the **volatility smile**. Figure 2 illustrates this behavior. Panel 2(a) shows the decrease of implied

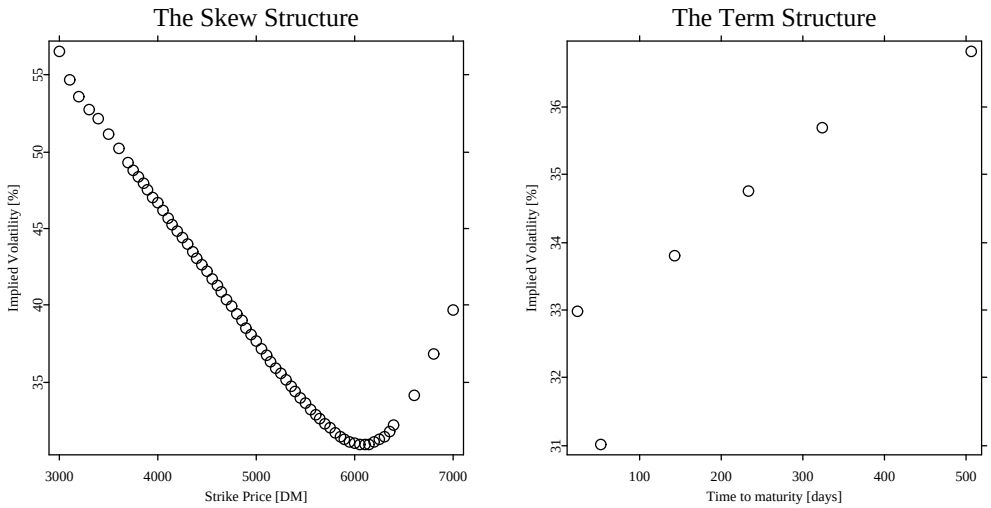

Figure 2: Implied Volatilities of DAX Options on Jan 29, 1999.

volatility σ_{imp} with the strike level of the put options on the DAX index with a fixed expiration of 52 days, as observed on January 29, 1999. Panel 2(b) shows the increase of σ_{imp} with the time to expiration of put options with a fixed strike level 6000 DM.

Implied volatility of an option is the market's estimate of the average future underlying volatility during the life of that option, **local volatility** is the market's estimate of underlying volatility at a particular future time and market level. Let us recall that as a consequence of complete markets, the Black-Scholes model is risk-neutral.

The most natural simplification of the Black-Scholes model are binomial trees. The binomial tree corresponding to the risk neutral underlying evaluation process is the same for all options on this asset, no matter what the strike price or time to expiration is. The tree cannot know about which option we are valuing on it. There are several alternative ways to construct a binomial tree. Here we will recall the familiar Cox-Ross-Rubinstein binomial tree. It has constant logarithmic spacing between nodes on the same level. This spacing shows the future price volatility. Figure 3 shows a standard CRR tree. Starting at a general node S, the price of the underlying asset can either increase to the higher price S_u with the probability p or decrease to the lower price S_d with the probability $1-p$:

$$S_u = Se^{\sigma\sqrt{\Delta t}} \tag{2}$$

$$S_d = Se^{-\sigma\sqrt{\Delta t}} \tag{3}$$

$$p = \frac{F - S_d}{S_u - S_d} \tag{4}$$

where F denotes the forward price in the node S and σ is the constant volatility. For other types of binomial trees see (Derman, Kani and Chriss, 1996) .

New models try to overcome the problem of varying volatility and to price options consistent with the market prices — to account for the volatility smile. However many of them extend the original Black-Scholes theory. Let us mention, for instance, incorporating a stochastic volatility factor or discontinuous jumps in the underlying price. But these methods bring other problems. In the next section we will pay attention to another approach of Black-Scholes extension developed in (Derman and Kani, 1994) — to the implied trees.

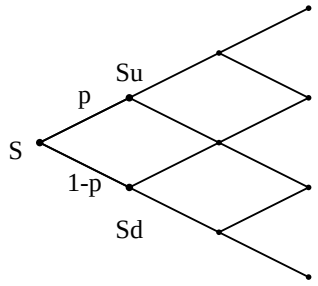

Figure 3: CRR Binomial Tree.

3 Trees and Implied Models

While the Black-Scholes attitude assumes that the underlying asset follows geometric Brownian motion from equation 1 with a constant volatility, implied theories assume that the stock or index price follows a process, whose volatility $\sigma(S,t)$ varies with spot price and time. This process can be expressed by the following stochastic differential equation:

$$\frac{dS_t}{S_t} = \mu dt + \sigma(S,t)dZ_t \tag{5}$$

This approach ensures that the valuation of the option remains preference-free (all uncertainty is in the spot price and thus we can hedge options using the underlying). Derman and Kani (1994) shows that it is possible to determine $\sigma(S,t)$ directly from the market prices of liquidly traded options.

The most natural discrete representation of a non-lognormal evolution process of the underlying prices is an implied binomial tree (IBT). It is similarly like the CRR binomial tree is a discrete version of the Black-Scholes constant-volatility process. For an exhaustive explanation on IBT's see (Derman and Kani, 1994). Generally we can use any (higher) multinomial tree as a discrete development from the Black-Scholes model. However, all of them converge, as the time step tends towards zero, to the same continuous result — the constant-volatility process. IBT's are, from the set of

all implied multinomial trees, minimal, i.e. they have only one degree of freedom — the arbitrary choice of the central node at each level of the tree. In this sense one may feel that they are sufficient and no more complicated trees are necessary. Despite the limiting similarity of multinomial trees, one form of them could be more convenient. Sometimes one may wish more flexibility in the discrete world — to get transition probabilities and probability distributions which vary as smoothly as possible across the tree. This is important especially when the market option prices are not very precise (e.g. because of inefficiency, market frictions, etc.) It is useful to recall the concept of Arrow-Debreu prices before the following derivation in section 4. **Arrow-Debreu** price $\lambda_{n,i}$ at node (n, i) is computed as the sum over all paths starting in the root of the tree leading to node (n, i) of the product of the risklessly discounted transition probabilities. Arrow-Debreu price of the root is equal to one. Arrow-Debreu prices at the final level of a (multinomial) tree are a discrete approximation of the implied distribution. Notice that they are discounted and thus the risk-neutral probability corresponding to each node (at the final level) should be calculated as the product of the Arrow-Debreu price and the factor e^{rT}.

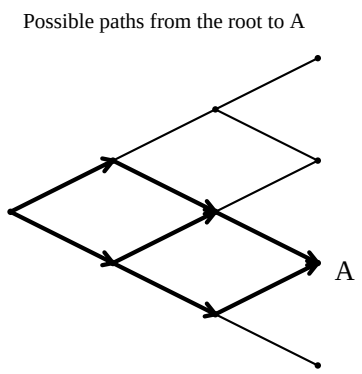

Figure 4: Computing Arrow-Debreu Price in a Binomial Tree.

4 ITT's and Their Construction

4.1 Basic insight

Trinomial tree with N levels is a set of nodes $s_{n,i}$, $n = 1, \ldots, N$, $i = 1, \ldots, 2n-1$. The underlying price can move from each node $s_{n,i}$ to one of three nodes: with probability p_i to the upper node, value $s_{n+1,i}$, with probability q_i to the lower node, value $s_{n+1,i+2}$ and with probability $1 - p_i - q_i$ to the middle node, value $s_{n+1,i+1}$. Let us denote the nodes in the new level with capital letters: $S_i(=s_{n+1,i})$, $S_{i+1}(=s_{n+1,i+1})$ and $S_{i+2}(=s_{n+1,i+2})$ respectively.

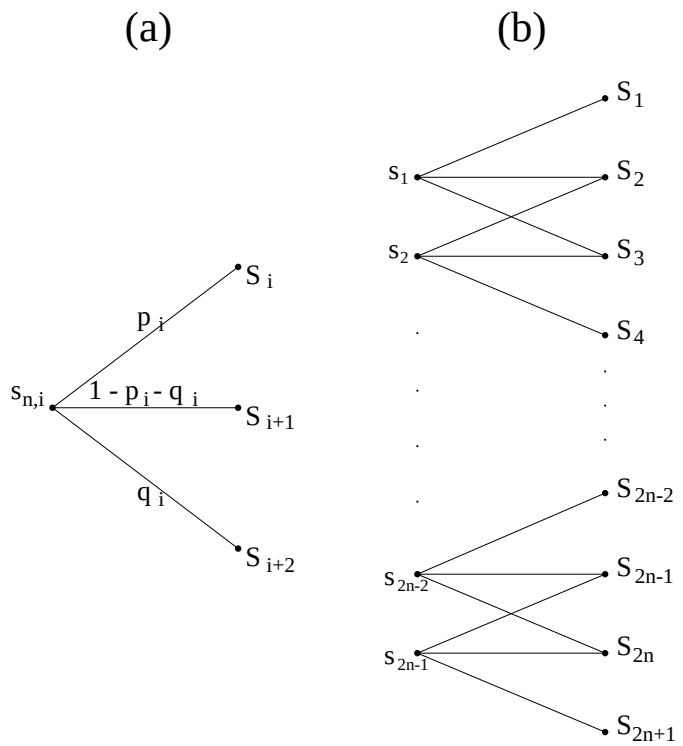

Figure 5: General Nodes in a Trinomial Tree.

Figure 5(a) shows a single node of a trinomial tree. Starting in the node $s_{n,i}$ at time t_n there are five unknown parameters: the two transition probabilities $p_{n,i}$, $q_{n,i}$ and the three new node prices S_i, S_{i+1} and S_{i+2}. Let F_i denote the known forward price of the spot $s_{n,i}$ and λ_i the known Arrow-Debreu price at node (n, i). In particular,

Arrow-Debreu prices for a trinomial tree can be obtained by the following iterative formulas:

$$\lambda_{1,1} = 1 \tag{6}$$

$$\lambda_{n+1,1} = e^{-r\Delta t}\lambda_{n,1}p_{n,1} \tag{7}$$

$$\lambda_{n+1,2} = e^{-r\Delta t}\left(\lambda_{n,1}(1-p_{n,1}-q_{n,1})+\lambda_{n,2}p_{n,2}\right) \tag{8}$$

$$\lambda_{n+1,i+1} = e^{-r\Delta t}\left(\lambda_{n,i-1}q_{i-1}+\lambda_{n,i}(1-p_{n,i}-q_{n,i})+\lambda_{n,i+1}p_{n,i+1}\right) \tag{9}$$

$$\lambda_{n+1,2n} = e^{-r\Delta t}\left(\lambda_{n,2n-1}(1-p_{n,2n-1}-q_{n,2n-1})+\lambda_{n,2n-2}q_{n,2n-2}\right) \tag{10}$$

$$\lambda_{n+1,2n+1} = e^{-r\Delta t}\lambda_{n,2n-1}q_{n,2n-1} \tag{11}$$

Figure 5(b) shows two levels of a general trinomial tree [1]. The evolution process of underlying prices can be seen discrete through an implied binomial tree. Implied trinomial tree (ITT) is another discrete representation of the same process. What we desire from an ITT, and in general from any implied tree, is:

1. the tree reproduces correctly the volatility smile
2. the tree is risk-neutral
3. all the transition probabilities are from the interval $(0,1)$

To fulfill the risk-neutrality condition, the expected value of the underlying price $s_{n,i}$ in the next period t_{n+1} must be its know forward price:

$$Es_{n,i} = p_iS_i + (1-p_i-q_i)S_{i+1} + q_iS_{i+2} = F_i = e^{(r-\delta)\Delta t}s_{n,i} \tag{12}$$

where r denotes the continuous interest rate, δ the dividend yield and Δt is the time step from t_n to t_{n+1}. Derman, Kani and Chriss (1996) say that the node prices and transition probabilities also satisfy:

$$p_i(S_i-F_i)^2 + (1-p_i-q_i)(S_{i+1}-F_i)^2 + q_i(S_{i+2}-F_i)^2 = F_i^2\sigma_i^2\Delta t + o(\Delta t) \tag{13}$$

Where σ_i is the stock or index price volatility during this time period and $o(\Delta t)$ denotes a term of higher order than Δt. Therefore we have two constraints in equations 12 and 13 for five unknown parameters. As a consequence, there is no unique implied trinomial tree, we can thus choose one of the large set of equivalent ITT's. Equivalent in the sense, that as the time spacing Δt tends towards zero, all of these trees tend to give the same results. A common method is to choose freely the underlying prices and then to use these two equations and solve them for the transition probabilities p_i, q_i.

[1] Here we slightly modificated the notation of (Derman, Kani and Chriss, 1996) to be consistent with the program code.

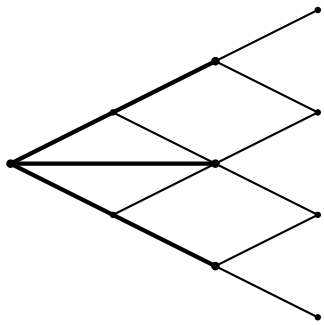

Figure 6: Building a Constant-Volatility Tree by Combining Two Steps of a CRR Binomial Tree.

Afterward we must make sure that these probabilities do not violate the condition 3 above. Since, in general, all multinomial trees converge to the same theoretical result (namely to the continuous process of evolution in the underlying asset), the reason why we use ITT's instead of IBT's is, so we gain these three degrees of freedom. This freedom may allow us to fit the smile better, especially when inconsistent- or arbitrage-violating market options prices make a consistent tree impossible. Even though the constructed tree is consistent, other difficulties can arise, when its local volatility and probability distributions are jagged and "implausible".

4.2 State space

There are several methods we can use to construct an initial state space. As already mentioned, all of them converge to the same theory (constant-volatility Black-Scholes model), in the continuous limit. In other words all these models are equivalent discretizations of the constant volatility diffusion process. We chose a constant-volatility Cox-Ross-Rubinstein (CRR) binomial tree. Then we combined two steps of this tree into a single step of our new trinomial tree. This is illustrated in figure 6. Using equations 2 and 3, we can derive the following formulas for the nodes in the new trinomial tree:

$$S_i = s_{n+1,i} = s_{n,i}\, e^{\sigma\sqrt{2\Delta t}} \tag{14}$$

$$S_i + 1 = s_{n+1,i+1} = s_{n,i} \tag{15}$$

$$S_i + 2 = s_{n+1,i+2} = s_{n,i}\, e^{-\sigma\sqrt{2\Delta t}} \tag{16}$$

Now, summing up the probabilities of reaching one of the three nodes, using formula 4, we get the relationship for the up and down transition probabilities in the trinomial tree (the middle transition probability is equal to $1 - p_i - q_i$):

$$
\begin{aligned}
p_i &= \left(\frac{e^{r\Delta t/2} - e^{-\sigma\sqrt{\Delta t/2}}}{e^{\sigma\sqrt{\Delta t/2}} - e^{-\sigma\sqrt{\Delta t/2}}} \right)^2 \\
q_i &= \left(\frac{e^{\sigma\sqrt{\Delta t/2}} - e^{r\Delta t/2}}{e^{\sigma\sqrt{\Delta t/2}} - e^{-\sigma\sqrt{\Delta t/2}}} \right)^2
\end{aligned}
$$

Derman, Kani and Chriss (1996) describes two other methods for building a constant-volatility trinomial tree.

When the implied volatility varies only slowly with strike and expiration, the regular state space with uniform mesh size is adequate for constructing ITT models. But if the volatility varies significantly with strike or time to maturity, we should choose a state space reflecting this properties. Construction of a trinomial space with proper skew and term structure proceeds in four steps.

- First, we build a regular trinomial lattice with constant time spacing Δt and constant level spacing, assuming all interest rates and dividends equal to zero. Such a lattice can be constructed in the way described above.
- Secondly, we modify Δt at different times. Let us denote the known equally spaced time points $t_0 = 0, t_1, \ldots, t_n = T$, then we can find the unknown scaled times $\tilde{t}_0 = 0, \tilde{t}_1, \ldots, \tilde{t}_n = T$ by solving the following set of non-linear equations:

$$\tilde{t}_k \sum_{i=1}^{n-1} \frac{1}{\sigma^2(\tilde{t}_i)} + \tilde{t}_k \frac{1}{\sigma^2(T)} = T \sum_{i=1}^{k} \frac{1}{\sigma^2(\tilde{t})} \quad ; k = 1, \ldots, n-1 \tag{17}$$

- Subsequently, we change ΔS at different levels. Denoting $S_1, \ldots, S_{2n+1}$ the original known underlying prices and $\tilde{S}_1, \ldots, \tilde{S}_{2n+1}$ the new scaled unknown underlying prices respectively, we solve the following equation:

$$\frac{\tilde{S}_k}{\tilde{S}_{k-1}} = exp\left\{ \frac{c}{\sigma(S_k)} ln \frac{S_k}{S_{k-1}} \right\} \quad k = 2, \ldots, 2n+1 \tag{18}$$

  for some constant c (recommended to be some typical value of the local volatility). There are $2n$ equations for $2n+1$ unknown parameters. Here we always suppose that the new central node is the same as the original central node: $\tilde{S}_{n+1} = S_{n+1}$. For a more elaborate explanation of the theory behind equations 17 and 18, set in the continuous frame, see (Derman, Kani and Chriss, 1996).
- Lastly, you can grow all the node prices by a sufficiently large growth factor which removes forward prices violations (explained in subsection 4.4). Multiplying all zero-rate node prices at time t_n by $e^{(r-\delta)(t_n - t_{n-1})}$ should be always sufficient.

4.3 Transition probabilities

Once the state space of the ITT is fixed, we can compute the transition probabilities for all tree nodes (n, i) at each tree level n. Let $C(K, t_{n+1})$ and $P(K, t_{n+1})$ denote today's price for a standard European call and put option, respectively, struck at K and expiring at t_{n+1}. These values can be obtained by interpolating the smile surface at various strike and time points. Then the trinomial tree value of an option is the sum over all nodes $(n+1, j)$ of the discounted probability of reaching that node multiplied by the pay-off function, i.e. $max(S_j - K, 0)$ for the call and $max(K - S_j, 0)$ for the put option respectively, where S_j denotes the spot price at the node $(n+1, j)$:

$$C\left(K, t_{n+1}\right) = e^{-r\Delta t} \sum_j \left\{p_j \lambda_j + (1 - p_{j-1} - q_{j-1})\lambda_{j-1} + q_{j-2}\lambda_{j-2}\right\} \left(S_j - K\right)^+ \quad (19)$$

$$P\left(K, t_{n+1}\right) = e^{-r\Delta t} \sum_j \left\{p_j \lambda_j + (1 - p_{j-1} - q_{j-1})\lambda_{j-1} + q_{j-2}\lambda_{j-2}\right\} \left(K - S_j\right)^+ \quad (20)$$

If we set the strike price K to the value S_{i+1}, the stock price at the node $(n+1, i+1)$, rearrange the terms in the sum and use equation 12 we can compute from equation 19 the transition probabilities p_i, q_i for all the nodes above the central node:

$$p_i = \frac{e^{r\Delta t} C(S_{i+1}, t_{n+1}) - \sum_{j=1}^{i-1} \lambda_j (F_j - S_{i+1})}{\lambda_i (S_i - S_{i+1})} \quad (21)$$

$$q_i = \frac{F_i - p_i (S_i - S_{i+1}) - S_{i+1}}{S_{i+2} - S_{i+1}} \quad (22)$$

Similarly, we compute the transition probabilities from equation 20 for all the nodes below (and including) the center node $(n+1, n)$ at time t_n:

$$q_i = \frac{e^{r\Delta t} P(S_{i+1}, t_{n+1}) - \sum_{j=i+1}^{2n-1} \lambda_j (S_{i+1} - F_j)}{\lambda_i (S_{i+1} - S_{i+2})} \quad (23)$$

$$p_i = \frac{F_i - q_i (S_{i+2} - S_{i+1}) - S_{i+1}}{S_i - S_{i+1}} \quad (24)$$

Derivation of these equations is described in detail in appendix B. Finally the implied local volatilities are approximated from equation 13:

$$\sigma_i^2 \doteq \frac{p_i (S_i - F_i)^2 + (1 - p_i - q_i)(S_{i+1} - F_i)^2 + q_i (S_{i+2} - F_i)^2}{F_i^2 \Delta t} \quad (25)$$

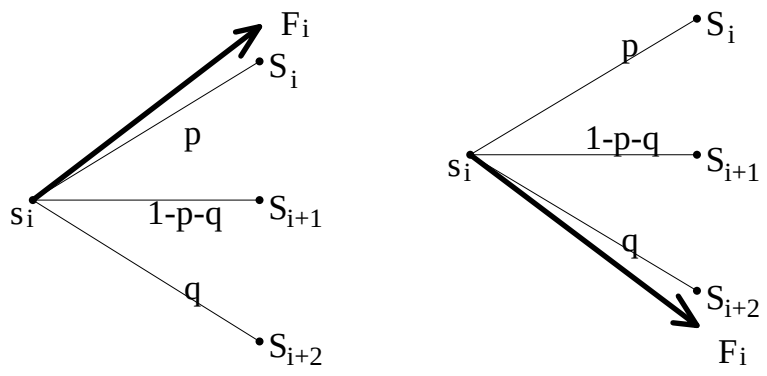

Figure 7: Two Kinds of the Forward Price Violation.

4.4 Possible pitfalls

Equations 21-24 can unfortunately result in transition probabilities which are negative or greater than 1. This is inconsistent with rational option prices and allows arbitrage. We must face two forms of this problem. First, we must watch that no forward price $F_{n,i}$ of a node (n, i) falls outside the range of daughter nodes at the level $n + 1$: $F_{n,i} \in (s_{n+1,i+2}, s_{n+1,i})$. Figure 7 shows this type of problem. This is not very difficult to overcome, since the state space is our free choice — we can overwrite the nodes causing this problem. The second problem refers to extreme values of options prices, which would imply an extreme value of local volatility. This may not be possible to obtain with probabilities between $(0, 1)$. In such case we have to overwrite the option price which produces the unacceptable probabilities. In practice it means that we overwrite even these probabilities with numbers from the interval $(0, 1)$ This is always possible if our state space does not violate the forward condition. From the great number of possible ways to overwrite wrong probabilities with numbers between 0 and 1, we chose to set:

$$p_i = \frac{1}{2}\left(\frac{F_i - S_{i+1}}{S_i - S_{i+1}} + \frac{F_i - S_{i+2}}{S_i - S_{i+2}}\right), \quad q_i = \frac{1}{2}\left(\frac{S_i - F_i}{S_i - S_{i+2}}\right) \tag{26}$$

if $F_i \in (S_{i+1}, S_i)$ and

$$p_i = \frac{1}{2}\left(\frac{F_i - S_{i+2}}{S_i - S_{i+2}}\right), \quad q_i = \frac{1}{2}\left(\frac{S_{i+1} - F_i}{S_{i+1} - S_{i+2}} + \frac{S_i - F_i}{S_i - S_{i+2}}\right) \tag{27}$$

if $F_i \in (S_{i+2}, S_{i+1})$. In either case the middle transition probability is equal to $1 - p_i - q_i$.

4.5 Illustrative examples

To illustrate the construction of an implied trinomial tree, let us now consider that the volatility varies only slowly with the strike price and time to expiration. Assume that the current index level is 100 EUR, the annual riskless interest rate $r = 12\%$, dividend yield is $\delta = 4\%$. The annual Black-Scholes implied volatility is assumed to be $\sigma = 11\%$, moreover let us assume that the Black-Scholes implied volatility increases (decreases) linearly by 10 basis points (=0.1%) with every 10 unit drop (rise) in the strike price, i.e. $\sigma_{impl} = 0.11 - \Delta Strike * 0.001$. To keep the example easy, we consider one-year steps over three years. First we construct the state space — a constant-volatility trinomial tree as described in section 4.2. The first node at time $t_0 = 0$,

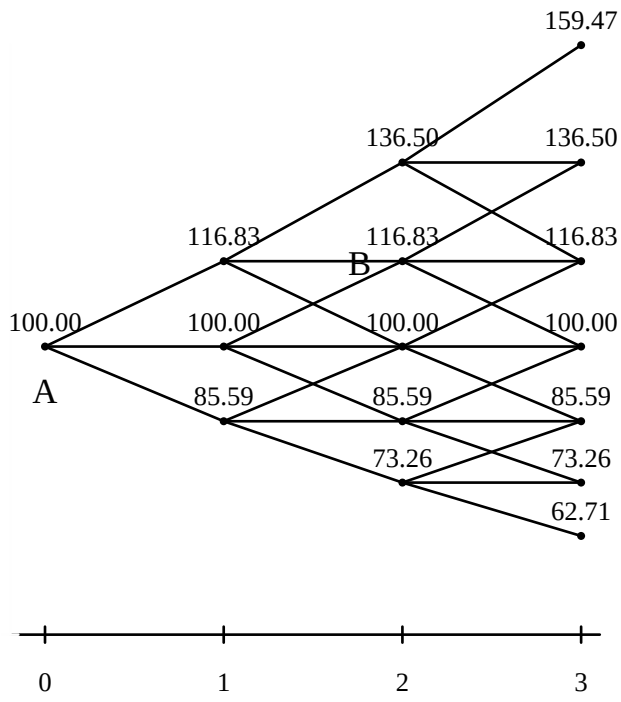

Figure 8: State Space of a Trinomial Tree with Constant Volatility of 11%.

labeled A in figure 8, has the value of $s_A = 100$ — today's spot price. The next three nodes, at time t_1, are computed from equations 14: $S_1 = 116.83$, $S_2 = 100.00$ and $S_3 = 85.59$, respectively. In order to determine the transition probabilities, we need to know the put price struck at $S_2 = 100$ and expiring one year from now $P(S_2, t_1)$.

From the smile, the implied volatility of this option is 11%. We calculate its price using a constant-volatility trinomial tree with the same state space, and find it to be 0.987 EUR. The forward price corresponding to node A is $F_A = Se^{(r^*-\delta^*)\Delta t} = 107.69$. Here r^* and δ^* denote the continuous interest rate $r^* = log(1+r)$ and $\delta^* = log(1+\delta)$, respectively. The down transition probability is then computed from equation 23:

$$q_A = \frac{e^{log(1+0.12)\times 1}0.987 - \Sigma}{1 \times (100.00 - 85.59)} = 0.077$$

The summation term Σ in the numerator is zero in this case, because there are no nodes with price lower than S_3 at time t_1. The up transition probability p_A is computed from equation 24:

$$p_A = \frac{107.69 + 0.077 \times (100.00 - 85.59) - 100}{116.83 - 100.00} = 0.523$$

The middle transition probability is then equal to $1 - p_A - q_A = 0.6$. As one can see from equations 6 — 11 the Arrow-Debreu prices turn out to be just discounted transition probabilities: $\lambda_{1,1} = 0.467$, $\lambda_{1,2} = 0.358$ and $\lambda_{1,3} = 0.069$. Equation 25 then gives the value of the implied local volatility at node A: $\sigma_A = 9.5\%$. Let us show the computation of one node more. Have a look at the node B in year 2 of figure 8. The index level at this node is $s_B = 116.83$ and its forward price one year later is $F_B = 125.82$. From this node, the index can move to one of three future nodes at time $t_3 = 3$, with prices $s_{3,2} = 136.50$, $s_{3,3} = 116.83$ and $s_{3,4} = 100.00$, respectively. The value of a call option struck at 116.83 and expiring at year 3 is $C(s_{3,3}, t_3) = 8.87$, corresponding to the implied volatility of 10.83% interpolated from the smile. The Arrow-Debreu price $\lambda_{2,2}$ is computed from equation 8:

$$\lambda_{2,2} = e^{-log(1+0.12)\times 1}(0.467 \times (1 - 0.517 - 0.070) + 0.358 \times 0.523) = 0.339$$

The numerical values here are the already known values from the last level t_1. The complete trees of Arrow-Debreu prices, transition probabilities and local volatilities for our example are shown in figures 9 — 11. Now, using equation 21 and 22 we can find the transition probabilities:

$$\begin{aligned} p_{2,2} &= \frac{e^{log(1+0.12)\times 1} \times 8.87 - \Sigma}{0.339 \times (136.50 - 116.83)} = 0.515 \\ q_{2,2} &= \frac{125.82 - 0.515 \times (136.50 - 116.83) - 116.83}{100 - 116.83} = 0.068 \end{aligned}$$

Where Σ contributes with the only term $0.215 \times (147 - 116.83)$ (there is one single node above S_B whose forward price is equal to 147). Last from equation 25, we find the implied local volatility at this node $\sigma_B = 9.3\%$.

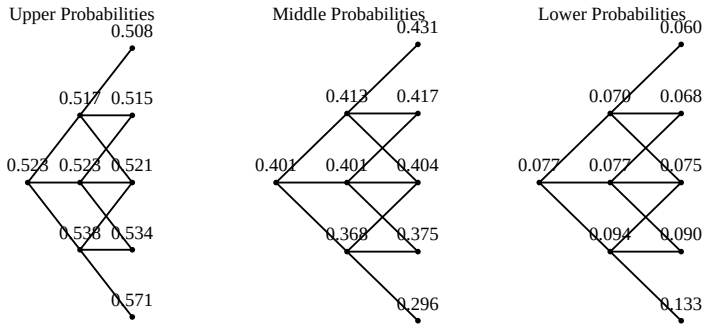

Figure 9: Transition Probabilities when $\sigma_{impl} = 0.11 - \Delta Strike * 0.001$.

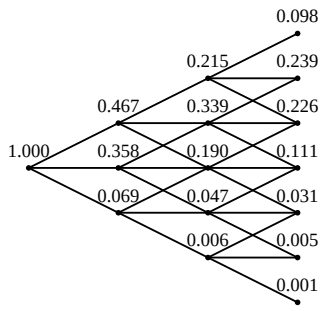

Figure 10: Tree of Arrow-Debreu prices when $\sigma_{impl} = 0.11 - \Delta Strike * 0.001$.

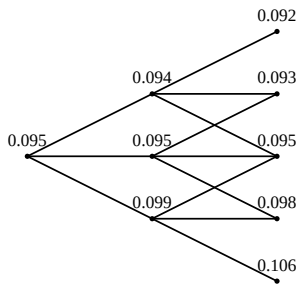

Figure 11: Implied Local Volatilities when $\sigma_{impl} = 0.11 - \Delta Strike * 0.001$.

As already mentioned in section 4.4, the transition probabilities may fall out of the interval $(0, 1)$. If we slightly modificate our previous example, we can show such a case. Let us now assume that the Black-Scholes volatility from our example increases (decreases) linearly 0.5 percentage points with every 10 unit drop (rise) in the strike price, i.e. $\sigma_{impl} = 0.11 - \Delta Strike * 0.05$ (that is, the volatility is five times as steep as before — illustrated in figure 12). Using the same state space, we find unadmissable transition probabilities at nodes C and D of figures 13 — 15. To overwrite them with plausible values, we used the set up described in equations 26 and 27.

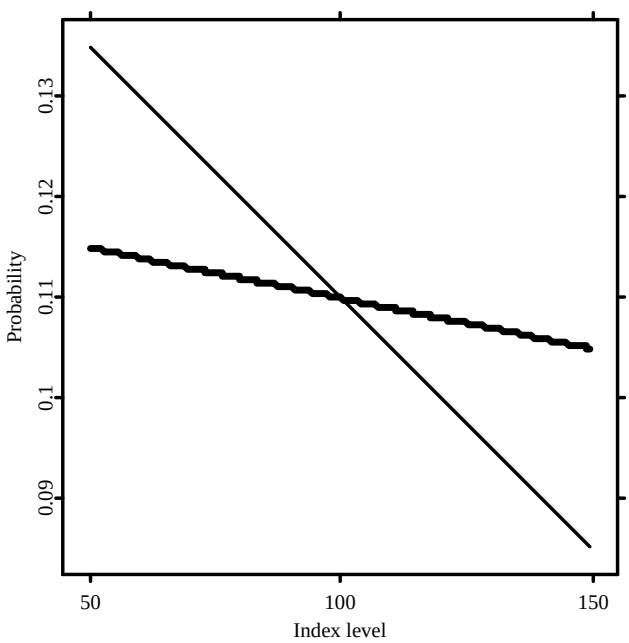

Figure 12: Skew structure from the examples. Thick line for $\sigma_{impl} = 0.11 - \Delta Strike * 0.001$ and thin line when $\sigma_{impl} = 0.11 - \Delta Strike * 0.005$.

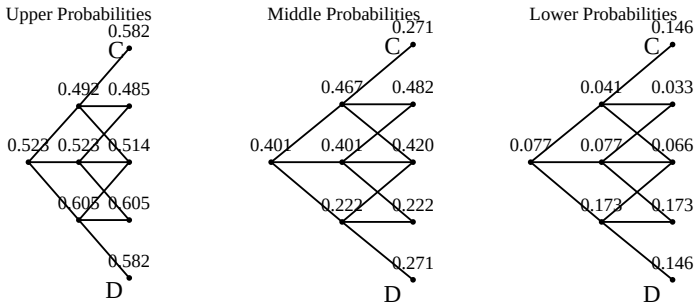

Figure 13: Transition Probabilities when $\sigma_{impl} = 0.11 - \Delta Strike * 0.005$.

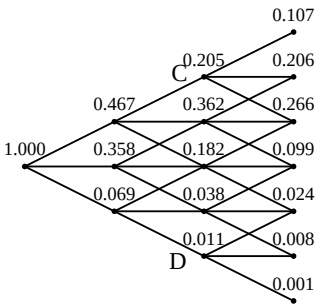

Figure 14: Tree of Arrow-Debreu prices when $\sigma_{impl} = 0.11 - \Delta Strike * 0.005$.

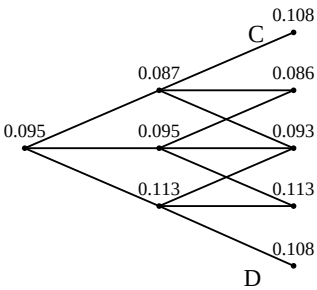

Figure 15: Implied Local Volatilities when $\sigma_{impl} = 0.11 - \Delta Strike * 0.005$.

5 ITT's in XploRe

Now we want to show how easy it is to compute implied trinomial trees in the statistical program XploRe. The quantlets which compute and show ITT's belong to the quantlib `finance` and will be explained in this section. The first subsection gives the necessary basic knowledge, which will be extended in the second subsection. Lastly, the third subsection describes quantlets which are not launched by the user, but are used for internal computations.

5.1 Basic skills

The simplest way to invoke implied trinomial trees in XploRe is to use the interactive menus, which guide you through the computation procedure:

```
{Ttree, P, Q, AD, LocVol, Onodes, Oprobs, Time} = ITT (volafunc)
```

There is only one parameter that has to be inserted — the name of the volatility function(s) `volafunc`. The output parameters will be discussed later on. First, we consider only slow volatility variation. Then `volafunc` is one string containing the name of the volatility function defined by the user. This function has the following form:

```
sigma = volafunc (S, K, tau)
```

where `S` denotes the spot price, `K` the exercise price and `tau` is the time to maturity on each level. For our first example we will assume that the implied volatility is 15% for all expirations and that it increases (decreases) 0.5% with every 10 points drop (rise) in the strike price:

```
library("finance")                ; load the library
proc(sigma)=volafunc(S, K, tau)   ; define the volatility function
  sigma=0.015 + (S-K)/10 * 0.05
endp
ITT("volafunc")                   ; ITT's with interactive menus
```

ITT01.xpl

XploRe asks you to insert the strike price, the annual interest rate and the divided yield. Subsequently, you must insert all the time points, when the steps of the ITT

occurred. At this stage let us recall that the tree is supposed to start at time $t_0 = 0$. To end the insertion of the time steps, type any non-positive value. For instance, in our example from subsection 4.5, we typed consecutively: $0, 1, 2, 3$ and 0. Once the ITT is computed, you can display the results — a menu asks you to choose between the state space, the tree of the transition probabilities, the tree of Arrow-Debreu prices, the tree of local volatilities and the state price density. If you choose the state space of the tree you can specify whether you want to describe the nodes with underlying prices or the arrows with transition probabilities respectively. While in this plot the scale of the y-axe corresponds to the underlying price, the tree in any other graph is only auxiliary and the vertical spacing does not mean anything.

Next, let us consider significant term and skew structure. In such cases the input parameter `volafunc` is a string vector with three rows containing names of functions with the following syntax (the names are arbitrary):

```
sigma = term (t)

sigmader = termder (t)

sigma = skew (S)
```

First row contains the name of the volatility function for the term structure defined by the user. Second row is the name of the quantlet returning Jacobian of this term structure function. Both of these two quantlets have only one parameter time `t` and it is necessary that they can use vectors. Third row is the name of the function for the skew structure. It has one input parameter `S` which is the spot price. If there is no term structure, but significant skew structure, insert in the first row only an empty string: `volafunc[1]=""`. The following example computes the state space of an ITT with significant term and skew structure:

```
library("finance")          ; load the libraries
library("nummath")
proc(fx)=t1(t)              ; the term structure function
  T=3                       ; T = time[rows(time)]
  q=0.3
  k=-q/(T+2)
  fx = sqrt(k*t+q)
endp
proc(fder)=t1der(t)         ; 1st derivative of the term structure
```

```
  T=3                          ; is the same as in t1(.)
  q=0.3
  k=-q/(T+2)
  fder = k / 2 / sqrt(k*t+q)
endp
proc(sigma)=struct(S)     ; the skew structure function
  K=100                   ; exercise price (= spot at t_0)
  sigma = (S/(3*K+S))^2
endp

S = 100                       ; current index level
r = 0.1                       ; compounded riskless interest rate
div = 0.05                    ; dividend yield
time = 0|1|2|3                ; starting time vector
volaf = "t1"|"t1der"|"struct"

impltree=ITT(S, r, div, time, volaf)  ; compute the ITT
```

ITT02.xpl

The state space is plotted in figure 16. Now we hid the part which manages about the plotting routines, this will be explained later.

The general form of `ITT` without the interactivity is:

```
{Ttree, P, Q, AD, LocVol, Onodes, Oprobs, Time} = ITT(S, r, div,
              time,volafunc{,skewconst})
```

where `S` is the spot price of the underlying at time $t_0 = 0$, `r` and `div` are compounded riskless interest rate and dividend yield (both of them $\in (0,1)$). `time` is vector of time points corresponding to the tree. They are to be given in years and they are supposed to increase. The parameter `volafunc` has been already described above and `skewconst` is an optional parameter which will be discussed in subsection 5.2. The output `Ttree` is a matrix containing the Implied Trinomial Tree. `Ttree[1,1]` is the root and all elements which do not belong to the tree are NaN's. `P` and `Q` are matrices with the up and down transition probabilities respectively, `AD` is a matrix containing the Arrow-Debreu prices for the computed tree and `LocVol` is a matrix with implied local volatilities. If there were some violations of the forward condition (described on the page 15) then `Onodes` is a matrix of `NaN`'s with the same dimensions like `Ttree`, but

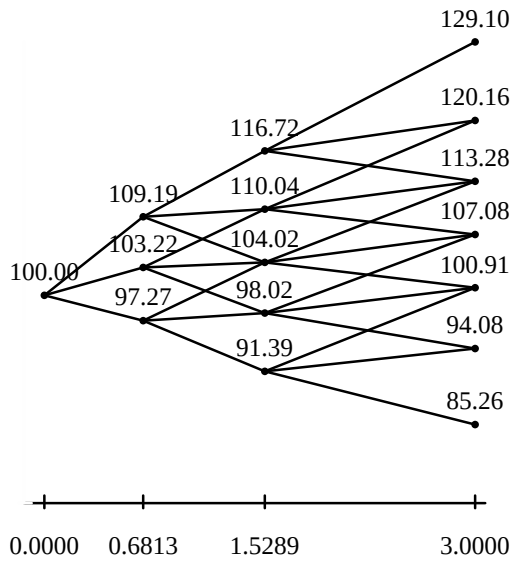

Figure 16: State Space with Significant Term and Volatility Structure.

on the places of the overwritten nodes is their value. If there were no overwritten nodes `Onodes` is only scalar (=NaN). `Oprobs` is a matrix with two columns containing x- and y-coordinates of the nodes, where a probability violated the second condition from subsection 4.4. If there were none, `Oprobs` is equal to `NaN`. But for each probability causing an overwrite, four rows are appended to the first auxiliary row `0~0`. First row of this quartet is the parental node where overwrite occurred, the three other rows are corresponding daughter nodes. `Time` is the same as the input parameter `time`, if there is no significant term structure. In other case it contains the solution of the equation system 17. So if we change the parameter `time` from our last example: `time=0|1|2`, then the variable `impltree` will contain these following components:

```
Contents of impltree.Ttree

[1,]      100    110.5   119.94
[2,]     +NAN   103.97   112.18
[3,]     +NAN   97.461   105.56
[4,]     +NAN     +NAN   98.944
[5,]     +NAN     +NAN   91.247
```

```
Contents of impltree.P

[1,]  0.45353  0.67267
[2,]     +NAN  0.72456
[3,]     +NAN  0.70371

Contents of impltree.Q

[1,]  0.00053582  0.11468
[2,]     +NAN  0.091887
[3,]     +NAN  0.093641

Contents of impltree.AD

[1,]        1   0.4123  0.24826
[2,]     +NAN   0.4963  0.40038
[3,]     +NAN  0.00048711  0.12417
[4,]     +NAN     +NAN  0.04091
[5,]     +NAN     +NAN  4.0831e-05

Contents of impltree.LocVol

[1,]  0.049924  0.040269
[2,]     +NAN  0.036105
[3,]     +NAN  0.04117

Contents of impltree.Onodes

[1,]     +NAN

Contents of impltree.Oprobs

[ 1,]        0        0
[ 2,]  0.83772    110.5
[ 3,]        2   119.94
[ 4,]        2   112.18
```

```
[ 5,]        2   105.56
[ 6,]  0.83772   103.97
[ 7,]        2   112.18
[ 8,]        2   105.56
[ 9,]        2   98.944
[10,]  0.83772   97.461
[11,]        2   105.56
[12,]        2   98.944
[13,]        2   91.247

Contents of impltree.Time

[1,]        0
[2,]  0.83772
[3,]        2
```

ITT03.xpl

The simplest way to display results is to use quantlet `plotITT`. It specifies explicitly which components of the computed ITT should be plotted.

```
plotITT (itree, what{, r{, text{, prtext}}})
```

The input parameter `itree` is just the output parameter of the quantlet `ITT`, i.e. a list containing `Ttree`, `P`, `Q`, `AD`, `LocVol`, `Onodes`, `Oprobs` and `Time`. `what` is a vector determining, which plots will be displayed. It has up to 5 rows and each non-zero tells XploRe that the corresponding plot should be shown. The possible plots are in sequence: 1) state space of the given ITT, 2) trees with transition probabilities, 3) the tree of local volatilities, 4) the tree of Arrow-Debreu prices and 5) the state price density. `r` is the interest rate ($\in (0,1)$), and it must be included whenever you want to show the state price density. `text` is an optional parameter which can be used only when the state space of the ITT should be plotted. It says whether the tree nodes should be described (`text` is a nonzero) or not (`text = 0`). And similarly `prtext` determines whether the arrows connecting the state space should be described with corresponding transition probabilities. In default, the state space denotes any node which violated the forward condition with a red rhombus and overwritten probabilities are shown with magenta lines. For instance, the following commands produce plot of

the state price density of the implied trinomial tree from figure 1(b). The state price density is depicted in figure 17.

```
library("finance")
library("smoother")

proc(sigma)=volafunc(S,K,time)
  sigma=0.15 + (S-K)/10 * 0.005
endp

S = 100          ; current index level
r = 0.0625             ; compounded riskless interest rate
div = 0.04005          ; dividend yield
time = 0:10 ; ten years annually

t=ITT(S, r, div, time, "volafunc")  ; compute the ITT
what=0|0|0|0|1
plotITT(t,what,r)                   ; and plot SPD
```

ITT04.xpl

5.2 Advanced features

We have mentioned a constant c in section 4.2. Derman, Kani and Chriss (1996) suggest to set this constant, used for solving the non-linear equation system 18, to represent the at-the-money or some other typical value of local volatility. In default it is set to 0.1 in the quantlet `ITT`, but the parameter `skewconst` allows to set it for each time step on a different value.

In order to allow the user to compose the plot of any trinomial tree (it does not matter if it is a tree of state space, transition probabilities or something else) with other graphical objects into one graph, there are quantlets returning the tree as a graphical object. Moreover these quantlets allow the user to enjoy other features by plotting the trees. Let us start with the quantlet `grITTspd`:

```
dat = grITTspd (ttree,ad,r,time)
```

Here `ttree`, `ad` and `time` are the output parameters `Ttree`, `AD` and `Time` respectively,

which were returned by the quantlet `ITT`. `r` is then the corresponding compound interest rate. The output parameter `dat` is the estimated state price density as a graphical object. The density curve will be estimated using quantlet `regest` with quartic kernel and bandwidth of 20% of the range of the last level in `ttree`.

Next, we introduce a quantlet used in general for any component of a trinomial tree:

```
nodes = grITTcrr (onetree{,time{,col{,scale{,prec}}}})
```

Here `onetree` is any matrix output of the quantlet `ITT`: `Ttree`, `P`, `Q`, `AD` or `LocVol`. `time` is an optional parameter, in default it is equidistant time axe, but it can be also equal to the output parameter `Time` of `ITT`. Parameter `col` is (2x1) vector, which specify the colours in the tree. First row is the colour of the mesh and second row the colour of its description. `scale` is a boolean determining if the tree should be scaled by the values given in `onetree` or if a particular constant-volatility trinomial tree should be plotted and the values of `onetree` will be used only for description.

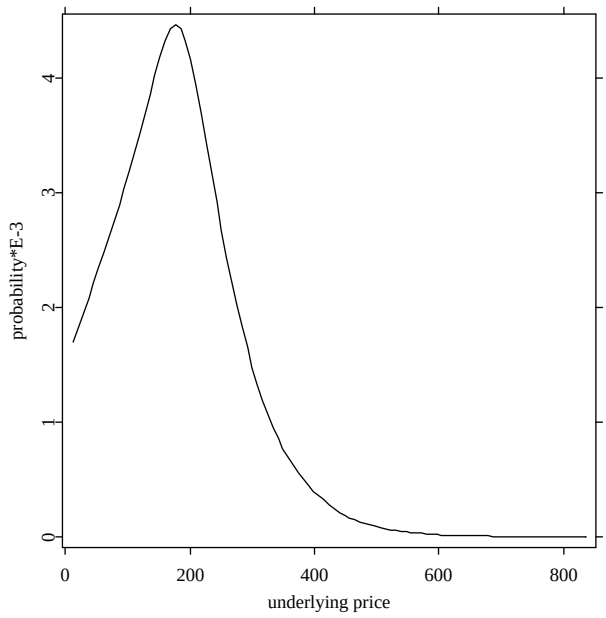

Figure 17: Estimation of the State Price Density from an ITT.

This is, for instance, used in the quantlet `plotITT` for plotting the transition probabilities whose values have nothing common with the vertical spacing in the tree. `prec` is a scalar specifying the precision of the description. The output parameter `nodes` is a graphical object again.

The last graphical tool is the quantlet `grITTstsp`:

```
{nodes,probs,axe,on,op} = grITTstsp (itree{,text{,col{,prec}}})
```

`itree` is the output of `ITT` again, `text` and `prec` have the same meaning as before, but the parameter `col` has now five rows. They correspond in the following order to the colour of the tree mesh, of the nodes description, of the probabilities description, of the overwritten nodes, and lastly, of the colour of overwritten probabilities. Here we have five output parameters — all of them are composed graphical objects corresponding in sequence to the tree nodes, to the description of the probabilities, to the time axe, to the overwritten nodes and to the overwritten probabilities. For instance, figure 16 was produced by the following commands (the ITT `impltree` has been already computed on the page 23):

```
axesoff()
grtree=grITTstsp(impltree,1,0*matrix(5),2|2)  ; black state space
d=createdisplay(1,1)
show(d,1,1,grtree.nodes,grtree.axe)
setgopt(d,1,1,"border",0)
axeson()
```

`ITT02.xpl`

Note that if there were no overwrites in the nodes or probabilities, `on` or `op` are equal to NaN and it is not possible to show them.

If one wants to get a text output (the graphical output may be too "wild" and its description unreadable), it is possible to write the result in the XploRe output window as a string:

```
ITTnicemat (title,onetree,prec)
```

Here `title` is a string with desired title of the matrix, `onetree` is any output (tree) matrix from the quantlet `ITT` and `prec` specifies the precision. You can compare figure 1(a) with the result of `nicemat`: `ITTnicemat`

```
State Space Matrix
                          123.50
                  113.08  114.61
          105.22  106.34  113.60
 100.00   100.00  100.01  106.82
           95.04   94.04  100.46
                   88.43   87.27
                           80.97
```

ITT05.xpl

Last in this section we will introduce a quantlet, which actually belongs into the next section, but it is also possible to use it separately. `ITTcrr` is called by `ITT` to compute the option price. It builds up a constant-volatility trinomial tree (combining two steps of a standard CRR binomial tree), its formal definition is:

```
{ttree,optprice} = ITTcrr (S, K, r, sigma, time, opt, div)
```

where `S` is the the spot price of the underlying, `K` is the exercise price, `r` is the CONTINUOUS interest rate ($\in (0,1)$), `sigma` is the constant volatility, `time` is the vector of time points, `opt = 1` specifies a call option and `opt = 0` a put option. `div` is the dividend rate ($\in (0,1)$). The output parameters are `ttree` and `optprice`, the constant-volatility trinomial tree and the price of given option respectively.

5.3 What is hidden

This subsection comments on auxiliary quantlets which are not intended for a direct usage. First we present the quantlet `ITTnewnodes`, which computes a new level of an implied trinomial tree. It also checks that the forward condition from section 4.4 is fulfilled.

```
{newnodes,overwritten} = ITTnewnodes (old, r, sigma, deltat, div)
```

`old` is a vector containing the underlying prices on last known level of the tree. `r` and `div` are the continuous interest rate and dividend yield, `sigma` is the volatility and `deltat` is the time step between the last and the new level $\Delta t = t_{n+1} - t_n$ The

outputs are two vectors: the vector with the prices in the new level `newnodes` and the vector `overwritten`. This corresponds to `newnodes`, it is firstly filled with `NaN`'s and afterward, if a node is overwritten, the corresponding node in `overwritten` contains its new value. It is worth to mention how the implausible price will be overwritten: If the highest new underlying price S_1 is lower than the forward price F_1 at the node $s_{n,1}$, it will be set up one percent higher than the forward, and similarly, any lowest new underlying price S_{2n+1} which is higher than the forward price F_{2n-1} of the node $s_{n,2n-1}$ will be set 1% lower than this forward price.

The Arrow-Debreu prices will be computed stepwise with the same philosophy, like the state space, with the difference that no overwrites are necessary.

```
newad = ITTad (up, down, ad, r, deltat)
```

`ITTad` computes the Arrow-Debreu prices of an Implied Trinomial Tree on one particular level. `up` and `down` are vectors of up and down transition probabilities going from the last level of known Arrow-Debreu prices `ad`. The new computed prices are then returned in the vector `newad`.

```
fx = ITTterm (t)
```

`ITTterm` represents the non-linear equation system 17 — it computes its value in time points given by the vector `t`. The equation system is solved with the quantlet `nmnewton`. The Jacobian matrix necessary for the numerical solution is returned by quantlet `ITTtermder`:

```
jac = ITTtermder (t)
```

Appendix

A List of Quantlets

`grITTcrr` generates a trinomial tree and describes it with given values

`grITTspd` generates state price density of given implied trinomial tree

`grITTstsp` generates the state space of given implied trinomial tree

`ITT` is the main function for the Derman/Kani/Chriss method of implied trinomial trees (ITT). It computes the tree with the evolution tree of underlying prices, probability trees with up and down transition probabilities respectively, the tree of Arrow-Debreu prices and the local volatility tree.

`ITTad` is an auxiliary macro for `ITT`. It computes the Arrow-Debreu prices of an Implied Trinomial Tree on one particular level.

`ITTcrr` builds up a constant-volatility trinomial tree by combining two steps of a CRR binomial tree and computes option price of given option with given strike price.

`ITTnewnodes` is an auxiliary macro for `ITT`. It computes new level for an implied trinomial tree, so that the forward condition is not violated.

`ITTnicemat` writes a given matrix as a string with centered trinomial tree in the output window.

`ITTterm` is an auxiliary macro for `ITT`. It computes the implied volatility if there is significant term structure. It is used for the numerical solution of a non-linear equation system with the quantlet `nmnewton`.

`ITTtermder` is an auxiliary macro for `ITT`. It computes the Jacobian matrix of the non-linear equation system specified in ITTterm. It is necessary for its numerical solution with the quantlet `nmnewton`.

`plotITT` plots desired components of a given implied trinomial tree. They are namely: plot of the state space, plot with all three transition probabilities (up, middle and down), plot of the tree with Arrow-Debreu prices, plot of local volatilities tree and plot of the state price density estimated from the implied trinomial tree.

B Derivation of the transition probabilities

Let us recall that since the trinomial tree is risk-free, the forward price satisfies the condition:

$$F_i = p_i S_i + (1 - p_i - q_i) S_{i+1} + q_i S_{i+2} \tag{28}$$

and that the call price is computed from the following equation:

$$C\left(K, t_{n+1}\right) = e^{-r\Delta t} \sum_j \left[p_j \lambda_j + (1 - p_{j-1} - q_{j-1}) \lambda_{j-1} + q_{j-2} \lambda_{j-2}\right] \left(S_j - K\right)^+$$

Now, we set the strike price K to the value S_{i+1}, the stock price at the node $(n+1, i+1)$ and expand the sum:

$$\begin{aligned}
&e^{r\Delta t} C\left(S_{i+1}, t_{n+1}\right) = \sum_{j=1}^{i} [\ldots](S_j - K) = \\
&\underline{\lambda_1 p_1 S_1} \quad + \\
&\lambda_2 p_2 S_2 \quad + \underline{\lambda_1 (1 - p_1 - q_1) S_2} \quad + \\
&\lambda_3 p_3 S_3 \quad + \lambda_2 (1 - p_2 - q_2) S_3 \quad + \underline{\lambda_1 q_1 S_3} \quad + \\
&\qquad \vdots \\
&\underline{\lambda_{i-2} p_{i-2} S_{i-2}} + \lambda_{i-3} (1 - p_{i-3} - q_{i-3}) S_{i-2} + \lambda_{i-4} q_{i-4} S_{i-2} + \\
&\lambda_{i-1} p_{i-1} S_{i-1} + \underline{\lambda_{i-2} (1 - p_{i-2} - q_{i-2}) S_{i-1}} + \lambda_{i-3} q_{i-3} S_{i-1} + \\
&\lambda_i p_i S_i \quad + \lambda_{i-1} (1 - p_{i-1} - q_{i-1}) S_i \quad + \underline{\lambda_{i-2} q_{i-2} S_i} \quad + \\
&-\overline{\lambda_1 p_1} S_{i+1} + \\
&-(\lambda_2 p_2 \quad + \overline{\lambda_1 (1 - p_1 - q_1)}) S_{i+1} + \\
&-(\lambda_3 p_3 \quad + \lambda_2 (1 - p_2 - q_2) \quad + \overline{\lambda_1 q_1}) S_{i+1} \quad + \\
&\qquad \vdots \\
&-(\overline{\lambda_{i-2} p_{i-2}} + \lambda_{i-3} (1 - p_{i-3} - q_{i-3}) + \lambda_{i-4} q_{i-4}) S_{i+1} + \\
&-(\lambda_{i-1} p_{i-1} + \overline{\lambda_{i-2} (1 - p_{i-2} - q_{i-2})} + \lambda_{i-3} q_{i-3}) S_{i+1} + \\
&-(\lambda_i p_i \quad + \lambda_{i-1} (1 - p_{i-1} - q_{i-1}) + \overline{\lambda_{i-2} q_{i-2}}) S_{i+1}
\end{aligned}$$

As one can see, each set of three over-lined terms is equal to λ_j. And similarly, using equation 28, each set of three underlined terms is equal to $\lambda_j F_j$. If we now add and at the same time subtract from the right hand side of this equation the term $\lambda_{i-1} q_{i-1} S_{i+1}$, we get:

$$e^{r\Delta t} C(S_{i+1}, t_{n+1}) = \lambda_i p_i (S_i - S_{i+1}) + \sum_{j=1}^{i-1} \lambda_j (F_j - S_{i+1}) \tag{29}$$

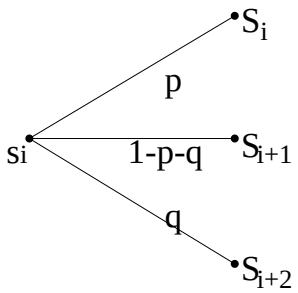

Figure 18: A General Node of the Trinomial Tree.

We can solve equation 29 for the transition probability p_i which is the only unknown parameter:

$$p_i = \frac{e^{r\Delta t}C(S_{i+1}, t_{n+1}) - \sum_{j=1}^{i-1} \lambda_j (F_j - S_{i+1})}{\lambda_i (S_i - S_{i+1})}$$

Then we use equation 28 to determine the down transition probability q_i:

$$q_i = \frac{F_i - p_i(S_i - S_{i+1}) - S_{i+1}}{S_{i+2} - S_{i+1}}$$

Similarly, for the nodes below (and including) the central node $(n+1)$, when going from the relationship for the put price:

$$P\left(K, t_{n+1}\right) = e^{-r\Delta t} \sum_j \left\{p_j \lambda_j + (1 - p_{j-1} - q_{j-1})\lambda_{j-1} + q_{j-2}\lambda_{j-2}\right\} \left(K - S_j\right)^+$$

we set K equal to S_{i+1} again, rearrange the terms in the sum and use equation 28:

$$e^{r\Delta t} P(S_{i+1}, t_{n+1}) = \lambda_i q_i (S_{i+1} - S_{i+2}) + \sum_{j=i+1}^{2n-1} \lambda_j (S_{i+1} - F_j)$$

Then we compute the corresponding transition probabilities:

$$q_i = \frac{e^{r\Delta t} P(S_{i+1}, t_{n+1}) - \sum_{j=i+1}^{2n-1} \lambda_j (S_{i+1} - F_j)}{\lambda_i (S_{i+1} - S_{i-2})}$$

$$p_i = \frac{F_i - q_i(S_{i+2} - S_{i+1}) - S_{i+1}}{S_i - S_{i+1}}$$

References

Derman, E. and Kani, I. (1994). The Volatility Smile and Its Implied Tree. *Quantitative Strategies Research*, http://gs.com/qs, Goldman Sachs.

Derman, E., Kani, I. and Chriss, N. (1996). Implied Trinomial Trees of the Volatility Smile. *Quantitative Strategies Research*, http://gs.com/qs, Goldman Sachs.

Härdle, W., Hlávka, Z. and Klinke, S. (2000). *XploRe Application Guide.* Springer Verlag, Heidelberg.

Härdle, W., Kleinow, T. and Stahl, G. (2002). *Applied Quantitative Finance.* Springer-Verlag, Berlin.

Hull, J (1989). *Options, Futures and Other Derivatives.* Prentice-Hall, Englewood Cliffs, New Jersey.

Ross, S., Westerfield, R. and Jaffe, J. (2002). *Corporate Finance.* Mc Graw-Hill.