Contents

1 Local Volatility and Implied Trees

```
library(fOptions)
library(fExoticOptions)
library(fAsianOptions)
library(fInstrument)
library(empfin)
```

The Breeden-Litzenberger formula provides the density of the underlying asset at expiry as a function of the price of European options priced in the Black-Scholes framework. In other words, it establishes the relationship between the implied Black-Scholes volatility $\Sigma(S, K, T)$ and the distribution of S_T, where S is today's value of the underlying asset, and T is the time from today to expiry. In many instances, for pricing American options for example, it is important to know the density of S_t at any time t between the current time and expiry. In this chapter, we consider the function $\sigma(S_t, t)$, i.e. the local volatility of return, as a function of the underlying asset and time.

The relationship between the Black-Scholes implied volatility $\Sigma(S, K, T)$ and $\sigma(S, t)$ was independently formalized in 1994, first by Derman and Kani in a binomial framework, then by Dupire in continuous time. These papers establish that there exists a unique stochastic process S_t with a variable continuous volatility $\sigma(S, t)$ that can fit all options quotes with their corresponding implied volatilities $\Sigma(S, K, T)$.

1.1 The Dupire local volatility equation

The Dupire equation, with no interest rate nor dividend is:

$$\frac{\sigma^2(K,T)}{2} = \frac{\frac{\partial C(S,K,T)}{\partial T}}{K^2 \frac{\partial^2 C}{\partial K^2}}$$

A few observations may help understand this formula. The numerator

$$\frac{\partial C(S,K,T)}{\partial T} = \frac{C(S,K,T+dT) - C(S,K,T)}{dT}$$

is proportional to a calendar spread for calls with strike K. Similarly, the denominator is proportional to a butterfly centered at K:

$$\frac{\partial^2 C}{\partial K^2} = \frac{C(S, K + dK, T) - 2C(S, K, T) + C(S, K - dK, T)}{dK^2}$$

The calendar spread has a value determined by the probability that $S_T = K$ and the volatility at T of the call maturing at $T + dT$:

$$C(S, K, T + dT) - C(S, K, T) \propto p(S, K, T)\sigma^2(K, T)$$

Recall also that according to the Breeden-Litzenberger formula, the probability $p(S, t, K, T)$ is proportional to the value of a butterfly spread:

$$p(S, K, T) \propto \frac{\partial^2 C}{\partial K^2} \propto \text{butterfly spread}$$

Which tells us, broadly speaking, that the local variance is the ratio of the value of a calendar spread and a butterfly spread. Finally, rewriting Dupire's equation as:

$$\frac{\partial C}{\partial T} - \frac{\sigma^2}{2} K^2 \frac{\partial^2 C}{\partial K^2} = 0$$

we see that if we have a continuous implied volatility surface $\Sigma(S, t, K, T)$, we can (in theory) compute $\frac{\partial C}{\partial T}$ and $\frac{\partial^2 C}{\partial K^2}$ and derive $\sigma(K, T)$.

We next follow E. Derman in what he calls "A poor man's derivation of the Dupire equation", presented in a binomial framework. This derivation builds upon the previous observation relating the Dupire equation to the ratio of a calendar spread and a butterfly spread.

To set up the stage, consider a Jarrow-Rudd binomial tree rooted in S_0; the time increment in the tree is $\Delta T/2$. The relevant part of the tree is presented below.

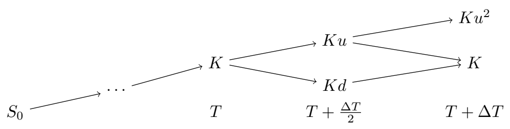

Figure 1.1: Call spread in a binomial model

Let $\phi(K) = C(S, K, T + dT) - C(S, K, T)$ be the value of the calendar call spread, and note that, in this binomial tree,

$$S_T < K => S_{T+\Delta T} <= 0 => \phi(K) = 0 \qquad (1.1)$$

$$S_T > K => E(S_{T+\Delta T} - K | S_T) = S_T - K => \phi(K) = 0 \qquad (1.2)$$

$$(1.3)$$

For the call spread to expire with a positive value, we must have $S_T = K$, and then follow two consecutive up branches of the binomial tree to reach Ku^2. The probability of this event is the probability of reaching node (K, T), $P_{K,T}$, times the probability of moving from (K, T) to $(Ke^{\sigma\sqrt{2\Delta T}}, T + \Delta T)$, which is $\frac{1}{4}$. The price being the probability times the payoff, which is dK.

$$C(S, K, T + \Delta T) - C(S, K, T) = \frac{\partial C}{\partial T}\Delta T = \frac{1}{4}P_{K,T}\, dK$$

For small ΔK, we can write $dK \approx \Delta K \approx K\sigma\sqrt{2\Delta T}$. Note that σ is the volatility at node (K, T) over the time interval $[T, T + \Delta T]$.

With zero interest rate, the probability of reaching node (K, T) is the state price of that node, which can be evaluated as the value of a butterfly that pays \$1 at node (K,T) and zero otherwise.

$$P_{K,T} = \frac{c(S, K + dK, T) - 2C(S, K, T) + C(S, K - dK, T)}{dK}$$

For small dK, we can write:

$$P_{K,T} \approx \frac{\partial^2 C}{\partial K^2} dK$$

Substituting this expression in (1.1), we get Dupire's equation:

$$\frac{\partial C}{\partial T} = \frac{1}{4}\frac{\partial^2 C}{\partial K^2}\frac{(K\sigma\sqrt{2\Delta T})^2}{\Delta T} \tag{1.4}$$

$$\sigma^2(K, T) = \frac{\frac{\partial C}{\partial T}}{\frac{1}{2}K^2\frac{\partial^2 C}{\partial K^2}} \tag{1.5}$$

1.2 *Computing $\sigma(K, T)$ in a trinomial tree*

In a standard trinomial tree, the transition probabilities are independent of the state. By contrast, the Derman-Kani implied tree involves transition probabilities that are state-dependent, with a local volatility $\sigma(K, T)$ associated with each node. With this extension, one obtain a model that matches the observed prices of vanilla options at various maturities and strikes.

We start with a trinomial tree constructed with a constant volatility. This determines the geometry of the tree, according to the formulae derived in Section **??**. We will now modify the transition probabilities in order to match observed option prices.

Let's focus on one node and its three children nodes. The notation is illustrated in Figure 1.2.

States $S_i, i = 1, \ldots, 2n - 1$ are associated with maturity T, and states S'_i with maturity $T + \Delta t$.

We will also use the following notation:

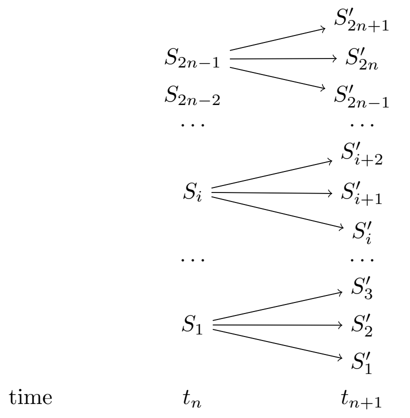

Figure 1.2: Two time slices in a trinomial tree. Time steps start at $t_1 = 0$, time step t_n has $(2n - 1)$ nodes.

λ_i. state price for node i,
$C(S'_i)$. price of a call of strike S'_i and maturity $T + \Delta t$.

The call can be priced in the trinomial tree, giving:

$$C(S'_{i+1}) = \lambda_i e^{-r\Delta t} p_i (S'_{i+2} - S'_{i+1}) \tag{1.6}$$

$$+ \sum_{j=i+1}^{2n} \lambda_j e^{-r\Delta t} \left[p_j (S'_{j+2} - S'_{i+1}) + (1 - p_j - q_j)(S'_{j+1} - S'_{i+1}) + q_j (S'_j - S'_{i+1}) \right] \tag{1.7}$$

The price process in the tree is a martingale, and thus the expected value at time t_{n+1}, given S_j must be the forward price at node (n, j):

$$S_j e^{r\Delta t} = F_j \tag{1.8}$$

$$= p_j S'_{j+2} + (1 - p_j - q_j) S'_{j+1} + q_j S'_j \tag{1.9}$$

Using this identiy, the call price becomes:

$$C(S'_{i+1}) = \lambda_i e^{-r\Delta t} p_i (S'_{i+2} - S'_{i+1}) \tag{1.10}$$

$$+ \sum_{j=i+1}^{2n} \lambda_j e^{-r\Delta t} \left[F_j - S'_{i+1} \right] \tag{1.11}$$

In the equation above, all quantities except p_i are known, yielding:

$$p_i = \frac{e^{r\Delta t} C(S'_{i+1}) - \sum_{j=i+1}^{2n} \lambda_j e^{-r\Delta t} \left[F_j - S'_{i+1} \right]}{\lambda_i (S'_{i+2} - S'_{i+1})}$$

Using again the martingale property, one can compute the probability q_i:

$$q_i = \frac{F_i - p_i (S'_{i+2} - S'_{i+1}) - S'_{i+1}}{S'_i - S'_{i+1}}$$

The corresponding local volatility at node S'_i is finally obtained by:

$$\sigma(S_i, t_n) F_i^2 \Delta t =$$
$$p_i (S'_{i+2} - F_i)^2 + (1 - p_i - q_i)(S'_{i+1} - F_i)^2 + q_i (S'_i - F_i)^2 \tag{1.12}$$

A similar calculation can be performed with put prices.

1.3 *Illustration*

For the sake of this example, we will use a simple expression for the Black-Scholes volatility, function of time to expiry (T) and strike (S):

$$\sigma(S,T) = \sigma_0(1 + a(S - 100)) + bT$$

We now proceed step-by-step in the calculation process. The first step is to construct a classical trinomial tree, with constant probabilities. This determines the geometry of the tree.

In this example, we construct a 3-periods tree, annual volatility is 30%. For simplicity, interest rate and dividend yield are set to 0.

The resulting tree is pictured in Figure 1.3. This is a standard trinomial tree: branching probabilities are identical at each node, and volatility at each node is also constant at 30% annual rate.

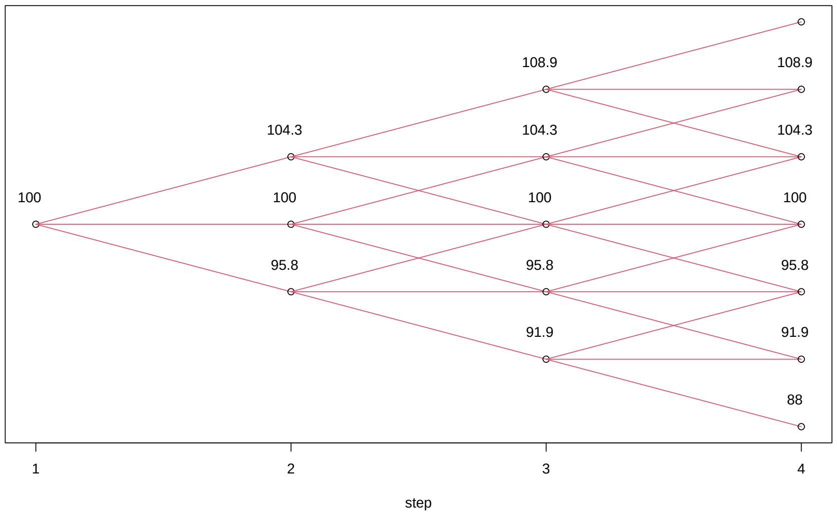

Figure 1.3: Constant volatility tree.

The next step is to price calls and puts expiring at each time step, and struck at each node value. The volatility used to price each option is time and strike dependent, per equation ((1.3)).

The volatility grid is computed by evaluating the volatility function for each node:

```
  Vol <- matrix(data=NA, nrow=nr, ncol=nc)
  for(i in seq(1,nr)) {
    for(j in seq(1, nc))
      if(!is.na(S[i,j])) Vol[i,j] <- bsvol(S[i,j], T[j])
  }
```

and this volatility is used to compute the call and put prices expiring at each time step and strike level:

```
Call <- matrix(data = NA, nrow = nr, ncol = nc)
Put <- matrix(data = NA, nrow = nr, ncol = nc)
for (i in seq(1, nr)) {
    for (j in seq(2, nc)) if (!is.na(S[i, j])) {
        Call[i, j] <- CRRTrinomial("ce", S[1, 1], S[i, j], T[j],
            r, b, Vol[i, j], j - 1)$price
        Put[i, j] <- CRRTrinomial("pe", S[1, 1], S[i, j], T[j],
            r, b, Vol[i, j], j - 1)$price
    }
}
```

Next, we use equations (1.2) and (1.2) to compute the transition probabilities at each node. The probability p_i is computed by:

```
p <- function(i, j) {
  # S'_{i+1} and S'_{i+2}
  SP1 <- S[i+1, j+1]
  SP2 <- S[i, j+1]
  # vector of lambdas
  tmp = 0
  if(i>1) {
    l <- Lambda[1:(i-1),j]
    F <- S[1:(i-1),j]*exp(r*dt)
    #print(paste('F: ', F, ' SP1: ', SP1, ' SP2: ', SP2))
    tmp = t(l) %*% (F-SP1)
  }
  # probability
  (exp(r*dt)*Call[i+1,j+1] - tmp)/(Lambda[i,j]*(SP2-SP1))
}
```

and the probability q_i is determined by:

```
q <- function(i, j) {
  # S'_{i+2}, S'_{i+1} and S'_{i}
  SP2 <- S[i, j+1]
  SP1 <- S[i+1, j+1]
  SP <- S[i+2, j+1]
  F <- S[i,j]*exp(r*dt)
  (F-p(i,j)*(SP2-SP1)-SP1)/(SP-SP1)
}
```

With these two functions, we can proceed recursively, computing the state prices and the transition probabilities one time step at a time. Note that functions $p(i,j)$ and $q(i,j)$ above use as input the state prices up to time step $(i-1)$ in order to compute the transition probabilities at time step i. The transition probabilities for the root node are computed immediately:

```
Lambda <- matrix(data=NA, nrow=7, ncol=4)
Lambda[1,1] <- 1
```

```
Pu <- p(1,1)
Pd <- q(1,1)
Pm <- 1-Pu-Pd
```

and this provides the data for computing the state prices $\lambda_{i,2}, i = 1, \ldots, 3$ for time step Δt.

```
Lambda[1,2] <- Pu * exp(-r*dt)
Lambda[2,2] <- Pm * exp(-r*dt)
Lambda[3,2] <- Pd * exp(-r*dt)
```

The state prices for the other time steps are computed similarly.

```
Lambda[1,3] <- p(1,2)*Lambda[1,2]
Lambda[2,3] <- (1-p(1,2)-q(1,2))*Lambda[1,2] + p(2,2)*Lambda[2,2]
Lambda[3,3] <- q(1,2)*Lambda[1,2] + (1-p(2,2)-q(2,2))*Lambda[2,2] + p(3,2)*Lambda[3,2]
Lambda[4,3] <- (1-p(3,2)-q(3,2))*Lambda[3,2] + q(2,2)*Lambda[2,2]
Lambda[5,3] <- q(3,2)*Lambda[3,2]

Lambda[1,4] <- p(1,3)*Lambda[1,3]
Lambda[2,4] <- (1-p(1,3)-q(1,3))*Lambda[1,3] + p(2,3)*Lambda[2,3]
Lambda[3,4] <- q(1,3)*Lambda[1,3] + (1-p(2,3)-q(2,3))*Lambda[2,3] + p(3,3)*Lambda[3,3]
Lambda[4,4] <- q(2,3)*Lambda[2,3] + (1-p(3,3)-q(3,3))*Lambda[3,3] + p(4,3)*Lambda[4,3]
Lambda[5,4] <- q(3,3)*Lambda[3,3] + (1-p(4,3)-q(4,3))*Lambda[4,3] + p(5,3)*Lambda[5,3]
Lambda[6,4] <- (1-p(5,3)-q(5,3))*Lambda[5,3] + q(4,3)*Lambda[4,3]
Lambda[7,4] <- q(5,3)*Lambda[5,3]
```

Since interest rate is 0, the state prices at each time step should sum up to 1. This is verified by:

```
z <- apply(Lambda, 2, function(x){sum(x[!is.na(x)])})
print(z)
```

```
## [1] 1 1 1 1
```

Having determined the state prices, we record the transition probabilities in grids, in order to facilitate the display. The up and down probabilities associated with each node are displayed in Figure 1.4).

```
Pup <- matrix(data=NA, nrow=nr, ncol=nc)
Pdn <- matrix(data=NA, nrow=nr, ncol=nc)
Pmd <- matrix(data=NA, nrow=nr, ncol=nc)
for(i in seq(1,nr)) {
  for(j in seq(1, nc-1))
    if(!is.na(S[i,j])) {
      Pup[i,j] <- p(i,j)
      Pdn[i,j] <- q(i,j)
```

```
        Pmd[i,j] <- 1-Pup[i,j]-Pdn[i,j]
      }
  }
```

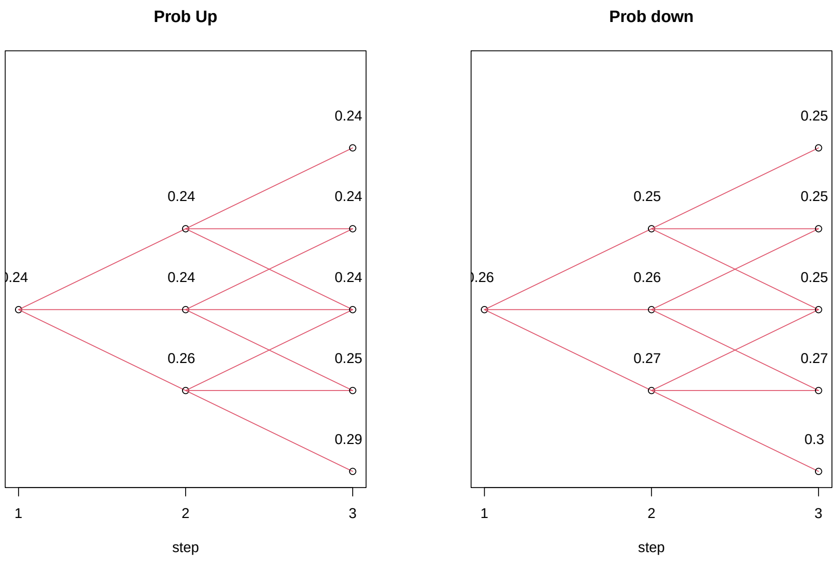

Figure 1.4: Up and down probabilitites in the implied tree.

Finally, the local volatility at each node can be computed from the transition probabilities:

```
  lvol <- function(i,j) {
    SP2 <- S[i, j+1]
    SP1 <- S[i+1, j+1]
    SP <- S[i+2, j+1]
    F <- S[i,j]*exp(r*dt)
    sqrt((Pup[i,j]*(SP2-F)^2 + Pdn[i,j]*(SP-F)^2 + Pmd[i,j]*(SP1-F)^2)/(F^2*dt))
  }

  LVol <- matrix(data=NA, nrow=nr, ncol=nc)
  for(i in seq(1,nr)) {
    for(j in seq(1, nc-1))
      if(!is.na(S[i,j])) {
        LVol[i,j] <- lvol(i,j)
      }
  }
```

Figure 1.5 shows the local volatility associated with each node, and, for comparison, the Black-Scholes volatility, which is the average volatility from time 0 to expiry associated with a particular strike and expiry date.

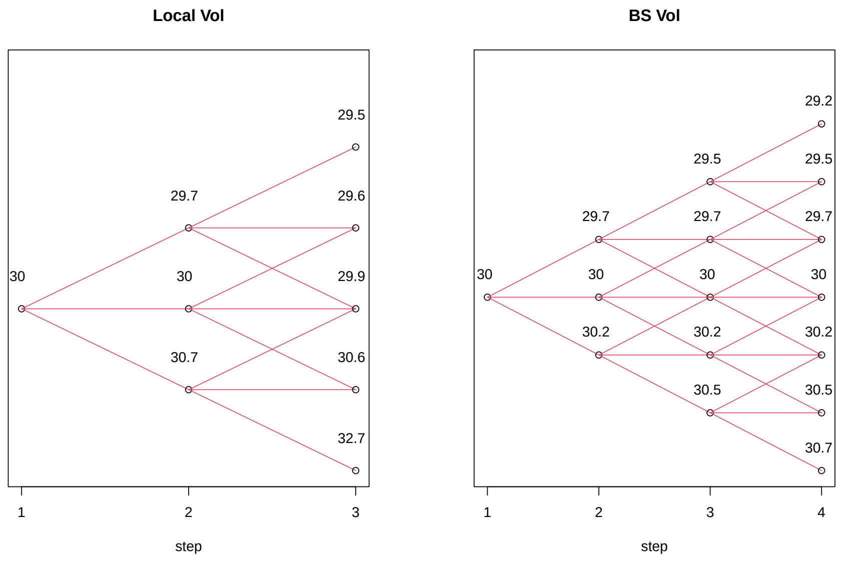

Figure 1.5: Local volatility and Black-Scholes average volatility in the implied tree.