Chapter 6

Approximating the underlying stochastic process – The tree methodology

6.1 Introduction

In chapter 3 we discussed various models where the stock process follows a stochastic differential equation (SDE). We discussed the theoretical solution of such equations and when the solution exists we derived a formula for the vanilla option price.

In general, finding analytic formulas is complicated and often impossible. In this chapter we will present a general methodology to approximate the solution of a SDE with a discrete time, discrete state space process. We shall refer throughout this chapter to such approximating processes as trees. Specifically, the problem can be formulated in the following way.

Suppose we have a stochastic process S_t solving the SDE:

$$dS_t = \mu(S_t)dt + \sigma(S_t)dB_t. \tag{6.1}$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ are known function. The theory presented at the end of this chapter applies to d-dimensional stochastic processes S_t not just one dimensional. Later in this chapter we shall present an approximation to a stochastic volatility model which will involve a 2 dimensional process.

The stochastic process in (6.1) is a time homogeneous Markov process. A tree approximation will be a Markov chain approximating this Markov process. It is important to understand how we create trees that approximate the diffusion process in equation (6.1). The sections at the end of this chapter are very general and may be skipped for the purpose of simple tree approximations (i.e. Black Scholes Merton model). However, to approximate more general processes this theory is crucial. The theoretical sections in this chapter are a very simplified summary of a much larger and old theoretical development. We refer to [197] and [61] for much more detail and proofs of the results presented here. For more details and examples of Markov processes we refer to [72].

6.2 Tree Methods: The Binomial tree

We will start this chapter with the simplest possible tree. The binomial tree was developed during the late 1970's by [46] and perfected in the early 1980's. The binomial tree model generates a pricing tree in which every node represents the price of an underlying financial instrument at a given point in time. The methodology can be used to price options with non standard features such as path dependence and barrier options. The main idea behind the binomial tree is to construct a discrete version of the geometric Brownian motion model:

$$dS_t = \mu S_t dt + \sigma S_t dW_t$$

which is a simplified version of the model in (6.1).

The process in (6.1) is a continuous process. Furthermore, since it is stochastic every time one constructs a path, the path would look different. The present value of a stock price is known to be S_0. The binomial tree (and indeed any tree in fact) may be thought of forming a collection of "most probably paths". When the number of steps in the tree approximation increases the tree contains more and better paths until at the limit we obtain the same realizations as the process in (6.1).

6.2.1 One step binomial tree

We start the construction by looking at a single step. The idea is that, if this step is done properly, the extension to any number of steps is straightforward.

In order to proceed, we state the Girsanov's theorem [157].

Theorem 6.2.1. (Girsanov, One-dimensional Case) *Let $B(t), 0 \leq t \leq T$, be a Brownian motion on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Let $\mathcal{F}(t), 0 \leq t \leq T$, be the accompanying filtration, and let $\theta(t), 0 \leq t \leq T$, be a process adapted to this filtration. For $0 \leq t \leq T$, define*

$$\tilde{B}(t) = \int_0^t \theta(u)du + B(t),$$

$$Z(t) = \exp\{-\int_0^t \theta(u)dB(u) - \frac{1}{2}\int_0^t \theta^2(u)du\},$$

and define a new probability measure by

$$\tilde{\mathbb{P}}(A) = \int_A Z(T)d\mathbb{P}, \quad \forall A \in \mathcal{F}.$$

Under $\tilde{\mathbb{P}}$, the process $\tilde{B}(t), 0 \leq t \leq T$, is a Brownian motion.

Girsanov's theorem allows us to go from the unknown drift parameter μ to a riskfree rate r which is common to all the assets. Applying Girsanov's theorem to (6.1) and using the Itô's formula on the process $X_t = \log S_t$, we obtain the stochastic differential equation:

$$dX_t = \left(r - \frac{\sigma^2}{2}\right) dt + \sigma dW_t. \tag{6.2}$$

We note that the stochastic equation of the log process $X_t = \log S_t$ is much simpler than the original SDE for the stock process. This translates into a much simpler construction for the X_t than for the original process S_t. For each choice of the martingale measure (probabilities in the tree) a tree for X_t is going to be perfectly equivalent with a tree for S_t and vice-versa.

Specifically, suppose that we have constructed a tree for the process X_t which has $x + \Delta x_u$ and $x + \Delta x_d$ steps up and down from x respectively. Then an equivalent tree for S_t could be constructed immediately by taking the next steps from $S = e^x$ as $Su = e^{x+\Delta x_u} = Se^{\Delta x_u}$ and $Sd = e^{x+\Delta x_d} = Se^{\Delta x_u}$, and keeping the probabilities identical. Similarly the tree in X_t is constructed from an S_t tree by taking $\Delta x_u = \log u$ for the up node and similarly for the down node.

Due to the obvious way in which the successors in the tree are calculated, the tree for the process X_t is called an *additive tree*, and the tree for the stock process S_t is called a *multiplicative tree*. We shall be focusing on the additive tree in the remainder of this chapter since the construction is simpler.

To have two successors for every node, at any point in the tree we can go up to $x + \Delta x_u$ with probability p_u or down to $x + \Delta x_d$ with probability $p_d = 1 - p_u$. The one-step tree for this process is presented in Figure 6.1.

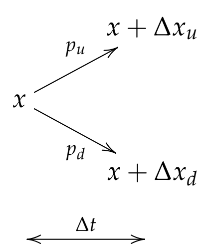

Figure 6.1: One step binomial tree for the return process

To determine the appropriate values for Δx_u, Δx_d, p_u, and p_d we use the diffusion approximation that will be presented in Section 6.9. However, the model in (6.2) is very simple and in this particular case it translates into needing to impose that the difference ($\Delta X_t = X_{t+\Delta t} - X_t$) between any two times calculated for the discrete tree model and the continuous time model should have the first two moments equal.

Fundamentally, in the simple case of a geometric BM, the process X_t is a normal random variable for any time t. We also know that a normal variable is entirely characterized by its mean and variance. Thus, equating the mean and variance increase for an infinitesimally small time step will be sufficient to make sure that the binomial tree will converge to the path given by the continuous process. Specifically, the conditions we need to enforce are:

$$\begin{cases} p_u \Delta x_u + p_d \Delta x_d & = \left(r - \frac{\sigma^2}{2}\right) \Delta t \\ p_u \Delta x_u^2 + p_d \Delta x_d^2 & = \sigma^2 \Delta t + \left(r - \frac{\sigma^2}{2}\right)^2 \Delta t^2 \\ p_u + p_d & = 1 \end{cases} \tag{6.3}$$

The system (6.3) has 3 equations and 4 unknowns, thus it has an infinite number of solutions. Indeed, the binomial tree approximation is not unique. In practice, we have a choice of a parameter - that choice will create a specific tree.

We note that for any choice of Δx_u and Δx_d the tree recombines. For example first step successors are $x + \Delta x_u$ and Δx_d. At the second step their successors will be:

$$x + 2\Delta x_u \text{ and } x + \Delta x_u + \Delta x_d \text{ for the upper node,}$$
$$\text{and } x + \Delta x_d + \Delta x_u \text{ and } x + 2\Delta x_d \text{ for the lower node.}$$

Clearly one of the nodes is identical and the tree recombines. The recombining feature in a tree is crucial. For example, consider a binomial tree with $n = 10$ steps that recombines and one that does not recombine. The recombining tree has a total number of nodes: $1 + 2 + 3 + \cdots + 11 = 11 \times 10/2 = 55$ nodes, whereas the one that does not recombine has $1 + 2 + 2^2 + 2^3 + \cdots + 2^{10} = 2^{11} - 1 = 2047$ nodes. The latter has a much bigger number of calculations that quickly become unmanageable.

As long as the resulting probabilities in (6.3) are positive the corresponding trees constructed with the particular choice of parameters are appropriate. Some popular choices are obtained by taking $\Delta x_u = \Delta x_d$ in the system (6.3). This produces what is called the Trigeorgis tree (see [198]) which supposedly has better approximation power.

Solving the system (6.3) by taking $\Delta x_u = \Delta x_d$ will yield:

$$\begin{cases} \Delta x & = \sqrt{\left(r - \frac{\sigma^2}{2}\right)^2 \Delta t^2 + \sigma^2 \Delta t} \\ p_u & = \frac{1}{2} + \frac{1}{2} \frac{\left(r - \frac{\sigma^2}{2}\right) \Delta t}{\Delta x} \end{cases} \tag{6.4}$$

The corresponding multiplicative tree for the S_t process obtained by setting the nodes as $S_t = \exp(X_t)$ and keeping all probabilities the same is the famous Cox-Ross-Rubinstein (CRR) tree (see [46]).

Another popular tree is obtained by setting the probabilities of jumps in (6.3) equal (i.e., $p_u = p_d = 1/2$). In this case the tree for S is called Jarrow Rudd tree (see [113]).

Remarks

- The system (6.3) could be further reduced by subtracting the constant term $\left(r - \frac{\sigma^2}{2}\right)\Delta t$ from each node, the resulting tree will be much simpler. One would have to remember to add the constant back in the final value of the tree at maturity.
- Finding the probabilities, p_u and p_d is equivalent to finding the martingale measure under which the pricing is made. For each solution one can construct an additive tree (moving from x to $x + \Delta x$ and $x - \Delta x$ with probabilities p_u and p_d respectively) or a multiplicative tree (moving from S to Su and Sd with probabilities p_u and p_d respectively). The two constructed trees are completely equivalent.
- Furthermore any tree works as long as the resulting probabilities are positive. The condition that the parameters need to satisfy to have proper probabilities are different for each type of tree. For example, for the symmetric steps tree ($\Delta x_u = -\Delta x_d = \Delta x$) the condition is $\Delta x > \left| r - \frac{\sigma^2}{2} \right| \Delta t$.

To construct a full tree we need to first solve the one step tree for a time interval $\Delta t = T/n$ where T is the maturity of the option to be priced and n is a parameter of our choice. We then use this tree to price options as in the next sections.

6.2.2 Using the tree to price an European Option

After constructing a binomial tree, we basically have 2^n possible paths of the process X_t or equivalently the process $S_t = \exp(X_t)$. Since these are possible paths we can use the binomial tree to calculate the present price of any path dependent option.

We begin this study by looking at the European Call. The European Call is written on a particular stock with current price S_0, and is characterized by maturity T and strike price K.

The first step is to divide the interval $[0, T]$ into $n + 1$ equally spaced points, then we will construct our tree with n steps, $\Delta t = T/n$ in this case. The times in our tree will be: $t_0 = 0, t_1 = T/n, t_2 = 2T/n, \dots, t_n = nT/n = T$.

Next we construct the return tree (X_t) as above starting with $X_0 = x_0 = \log(S_0)$, and the last branches of the tree ending in possible values for $X_{t_n} = X_T$. We remark that since we constructed the possible values at time T we can calculate for every terminal node in the tree the value of the Call option at that terminal node using:

$$C(T) = (S_T - K)_+ = \left(e^{X_T} - K\right)_+ \tag{6.5}$$

Thus now we know the possible payoffs of the option at time T. Suppose we want to calculate the value of the option at time $t = 0$. Using the Girsanov theorem and the Harrison and Pliska result [89], the discounted price of the process $\{e^{-rt}C(t)\}$ is a continuous time martingale. Therefore we may write that the value at time 0 must be:

$$C(0) = \mathbf{E}\left[e^{-rT}C(T)|\mathscr{F}_0\right].$$

Now we can use the basic properties of conditional expectation and the fact that $T/n = \Delta t$ or $e^{-rT} = \left(e^{-r\Delta T}\right)^n := \delta^n$ to write:

$$\begin{aligned} C(0) &= \mathbf{E}\left[\delta^n C(T)|\mathscr{F}_0\right] = \mathbf{E}\left[\delta^{n-1}\mathbf{E}\left[\delta C(T)|\mathscr{F}_1\right]|\mathscr{F}_0\right] \\ &= \delta\mathbf{E}\left[\delta\mathbf{E}\left[\dots\delta\mathbf{E}\left[\delta C(T)|\mathscr{F}_{n-1}\right]|\dots\mathscr{F}_1\right]|\mathscr{F}_0\right], \end{aligned}$$

where $\delta = e^{-r\Delta t}$ is a discount factor.

This formula allows us to recursively go back in the tree toward the time $t = 0$. When this happens, we will eventually reach the first node of the tree at C_0 and that will give us the value of the option. More precisely since we know the probabilities of the up and down steps to be p_u and p_d respectively, and the possible Call values one step ahead to be C^1 and C^2, we can calculate the value of the option at the previous step as:

$$C = e^{-r\Delta t}\left(p_d C^1 + p_u C^2\right). \tag{6.6}$$

What is remarkable about the above construction is that we constructed the return tree just to calculate the value of the option at the final nodes. Note that when we stepped backwards in the tree we did not use the intermediate values of the stock, we only used the probabilities. This is due to the fact that we are pricing the European option which only depends on the final stock price. This situation will change when pricing any path dependent option such as the American, Asian, barrier, etc.

Furthermore, we do not actually need to go through all the values in the tree for this option. For instance, the uppermost value in the final step has probability p_u^n. The next one can only be reached by paths with probability $p_u^{n-1}p_d$ and there are $\binom{n}{1}$ of them. Therefore, we can actually write the final value of the European call as:

$$C = \sum_{i=0}^{n}\binom{n}{i} p_u^{n-i} p_d^i (e^{x+(n-i)\Delta x_u + i\Delta x_d} - K)_+$$

as an alternative to going through the tree.

6.2.3 Using the tree to price an American Option

The American option can be exercised at any time, thus we will need to calculate its value at $t = 0$ and the optimal exercise time τ. Estimating τ is not a simple task, but in principle it can be done. τ is a random variable and therefore it has an expected value and a variance. The expected value can be calculated by looking at all the points in the tree where it is optimal to early exercise the option and the probabilities for all such points. Then we can calculate the expected value and variance.

For the value of the American option we will proceed in similar fashion as for the European option. We will construct the tree in the same way we did for the European option and then we will calculate the value of the option at the terminal nodes. For example for an American Put option we have:

$$P(T) = (K - S_T)_+ = \left(K - e^{X_T}\right)_+ \tag{6.7}$$

Then we recursively go back in the tree in a similar way as we did in the case of the European option. The only difference is that for the American option, we have the early exercise condition. So at every node we have to decide if it would have been optimal to exercise the option rather than to hold onto it. More precisely using the same notation as before, we again calculate the value of the option today if we would hold onto it as:

$$C = e^{-r\Delta t}\left(p_d C^1 + p_u C^2\right).$$

But what if we actually exercised the option at that point in time, say t_i? Then we would obtain $(K - S_{t_i})_+$. Since we can only do one of the two, we will obviously do the best thing, so the value of the option at this particular node will be the maximum of the two values. Then again, we recursively work the tree backward all the way to $C(0)$ and that will yield the value of the American Put option. A possible value of the optimal exercise time τ is encountered whenever the value obtained by exercising early $(K - S_{t_i})_+$ is larger than the expectation coming down from the tree.

We now discuss a more general approach of pricing path dependent options using the binomial tree techniques.

6.2.4 Using the tree to price any path dependent option

From the previous subsection we see that the binomial tree could be applied to any path dependent option. The trick is to keep track of the value of the option across various paths of the tree.

Please see [37] for a detailed example of a down and out American option. This is an American option (like before it can be exercised anytime during its lifespan). In addition if the asset price on which the option is dependent falls below a certain level H (down) then the option is worthless, it ceases to exist (out).

Binomial trees for dividend paying asset are very useful, however they are not sufficiently different from the regular binomial tree construction. The model developed thus far can be modified easily to price options on underlying assets other than dividend and non-dividend-paying assets.

Other applications of the binomial tree methodology is in the computation of hedge sensitivities. A brief introduction is presented below. Please refer to [37] for more details. In this application, we discuss how the binomial tree can be used to compute hedge sensitivities.

6.2.5 Using the tree for computing hedge sensitivities - the Greeks

When hedging an option as we discussed earlier in chapter 3 section 3.3, it is important to calculate the changes in option price with the various parameters in the model. We also recall in 3 section 3.3 that these derivatives are denoted with *delta* Δ, *gamma* Γ, *Vega*, *theta* Θ and *rho* ρ.

Their mathematical definition is given as:

$$\begin{cases} \Delta = \frac{\partial C}{\partial S} \\ \Gamma = \frac{\partial^2 C}{\partial S^2} \\ \text{Vega} = \frac{\partial C}{\partial \sigma} \\ \rho = \frac{\partial C}{\partial r} \\ \Theta = \frac{\partial C}{\partial t} \end{cases} \tag{6.8}$$

In order to approximate the greeks, one can use the method discussed by [37] that uses only one constructed binomial tree to calculate an approximation for Δ and Γ one step in the future.

Alternatively, one can construct two binomial trees in the case of Δ and three in the case of Γ starting from slightly different initial asset prices and use the same formula to approximate Δ and Γ at the present time.

In order to approximate Θ derivative, it requires knowledge of stock prices at two moments in time. It could be calculated using a similar approach as for *Vega* and ρ but since option price varies in time with the Brownian motion driving the process S_t, it is unnecessary to calculate the slope of the path.

6.2.6 Further discussion on the American option pricing

We consider a multiplicative binomial tree:

$$\begin{array}{cccc} & & Suu & \ldots \\ & Su & & \\ & & & \ldots \\ S & & Sud & \\ & & & \\ & Sd & & \\ & & Sdd & \ldots \end{array}$$

In the following examples we work with a probability of an up step:

$$\frac{Se^{r\Delta t} - S_d}{S_u - S_d} = \frac{e^{r\Delta t} - d}{u - d} = p$$

Example: European Call option.
Suppose a strike price $K = 105$, $u = 1$, $d = 0.9$ and $r = 7\%$ (annual) with T semesters and $\Delta T = \frac{1}{2}$. Suppose that the original price of the asset is $S = 100$. In this case, we have the following tree:

$$
\begin{array}{ccc}
 & & 121 \\
 & & (16) \\
 & 110 & \\
 & 10,4764 & \\
100 & & 99 \\
(6,8597) & & (0) \\
 & 90 & \\
 & (0) & \\
 & & 81 \\
 & & (0)
\end{array}
$$

The probability is then $p = 0.6781$. Next, when we consider an European Put option, the situation is different. In this case, we have:

$$
\begin{array}{ccc}
 & & 121 \\
 & & (0) \\
 & 110 & \\
 & (1,865) & \\
100 & & 99 \\
(4,772) & & (6) \\
 & 90 & \\
 & (11,3885) & \\
 & & 81 \\
 & & (24)
\end{array}
$$

Remark 6.2.1. *We observe that there exists a relation between the values of the Put and the Call options:*

$$C_t + Ke^{-r(T-t)} = P_t + S_t.$$

This relation is called the Put-Call parity *as discussed in chapter 3 section 3.9.*

We know that it is possible to exercise an American Call option at any time or, in the discrete case, at any node. We compute the maximum between 0 and $K - S_t$ i.e.

$$P_t = \max\{0, K - S_t\}.$$

We recall that in a Call option the relation is the reverse of the previous relation i.e.:

$$C_t = \max\{0, S_t - K\}.$$

We recall in chapter 2 that without considering dividends, the values of an American and European Call options are the same (i.e. $C_E = C_A$). Suppose that we exercise at time t, (i.e.: $S_t - K > 0$) and at a future time T, it grows with a risk-free rate $e^{r(T-t)}$. As the expected payoff is $E(S(T)) = S_t e^{r(T-t)} > S_t e^{r(T-t)} - K$, then it is not convenient to exercise, because the value will go up. We recall that in chapter 3 section 3.9, we can write the Put-Call parity as:

$$C_t - P_t = S_t - Ke^{-r(T-t)}.$$

For $t = T$,

$$\begin{aligned} C_T &= \max\{S_T - K, 0\} \\ P_T &= \max\{K - S_T, 0\}, \end{aligned}$$

so that we have the equality:

$$C_T - P_T = S_T - K$$

Now consider the relation:

$$\pi_t = \begin{cases} 1 \text{ Call} & \text{in long} \\ 1 \text{ Put} & \text{in short} \\ 1 \text{ share} & \text{in short} \end{cases}$$

Then we have that,

$$\begin{aligned} \pi_t &= C_t - P_t - S_t \\ \pi_T &= C_T - P_T = S_T - K - S_T = -K \end{aligned}$$

and assuming no arbitrage,

$$\pi_t = e^{-r(T-t)} \pi_T.$$

In general, suppose that at time T, we have $n+1$ possible states i.e. $T = n\Delta t$. We assume that,

$$S^j(T) = Su^j d^{n-j}$$

with

$$p = \frac{e^{r\Delta t} - d}{u - d}.$$

Then the probability that S is in the state S^j at time T is given as

$$P\left(S(T) = S^j(T)\right) = \binom{n}{j} p^j (1-p)^{n-j}.$$

and using the Newton binomial formula, the expected value is:

$$\begin{aligned} E(S(T)) &= \sum_{j=0}^{n} \binom{n}{j} p^j (1-p)^{n-j} Su^j d^{n-j} \\ &= S(pu + (1-p)d)^n. \end{aligned}$$

On the other hand,

$$pu + (1-p)\,d = p\,(u-d) + d = e^{r\Delta t}$$

and from $T = n\Delta t$ we obtain

$$E\,(S\,(T)) = S\,(pu + (1-p)\,d)^n = Se^{rn\Delta t} = Se^{rT}.$$

This implies that the expected value is proportional to the basis value and the proportion depends on the interest rate.

Now suppose we have the same situation with an European Call with strike price K and the relation:

$$C_j = C_{u^j d^{n-j}} = \max\{Su^j d^{n-j} - K, 0\},$$

then

$$\begin{aligned} E\,(C\,(T)) &= \sum_{j=0}^{n} \binom{n}{j} p^j\,(1-p)^{n-j}\,C_j \\ &= \sum_{j=n_0}^{n} \binom{n}{j} p^j\,(1-p)^{n-j}\,C_j \end{aligned}$$

where

$$n_0 = \min\{j : C_j > 0\}$$

Then substituting C_j, in the above equation, we have

$$\begin{aligned} E\,(C\,(T)) &= \sum_{j=n_0}^{n} \binom{n}{j} p^j\,(1-p)^{n-j} \left(Su^j d^{n-j} - K\right) \\ &= S\sum_{j=n_0}^{n} \binom{n}{j} (pu)^j\,((1-p)\,d)^{n-j} - KP\,(U \geq n_0) \end{aligned}$$

where U is a stochastic variable that follows a binomial distribution with mean $E\,(U) = np$ and variance $Var\,(U) = np\,(1-p)$.

Using the central limit theorem we can approximate the binomial distribution by a normal distribution and obtain:

$$P\,(U \geq n_0) \approx P\left(Z \geq \frac{n_0 - np}{\sqrt{np\,(1-p)}}\right).$$

6.2.7 A parenthesis: The Brownian motion as a limit of simple random walk

We begin our discussion of the Wiener process (also known as the Brownian motion) with an empirical analysis. Assume we have a random walk, which has a value x at time t. At the next time step $t + \Delta t$ it moves with probability

$\frac{1}{2}$, to either $x+\Delta x$ or $x-\Delta x$. Thus the random walk has two variables: the time $t \in [0,T]$ and the position $x \in [-X, X]$. Intuitively, we note that a *Markov process* does not have memory, that is the history of how the process reached position x is not relevant to where it goes next. Only the present state x is relevant for this.

For each $j = 1, 2, ..., n$, consider

$$u_{k,j} = P\left(x_{k\Delta t} = j\Delta x\right),$$

where k represents the time and j the position.

We recall that for $P(B) \neq 0$ we can define the conditional probability of the event A given that the event B occurs as:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}.$$

To the position j in time $k+1$ we can arrive only from the position $j-1$, or j at time k, so we have:

$$u_{k+1,j} = \frac{1}{2}\left(u_{k,j} + u_{k,j-1}\right). \tag{6.9}$$

We can rewrite (6.9) as:

$$u_{k+1,j} = \frac{1}{2}\left[\left(u_{k,j+1} - u_{k,j}\right) - \left(u_{k,j} - u_{k,j-1}\right)\right] + u_{k,j}$$

or

$$u_{k+1,j} - u_{k,j} = \frac{1}{2}\left[\left(u_{k,j+1} - u_{k,j}\right) - \left(u_{k,j} - u_{k,j-1}\right)\right].$$

Using the notation,

$$u_{k,j} = u\left(t_k, x_j\right),$$

we obtain:

$$u\left(t_{k+1}, x_j\right) - u\left(t_k, x_j\right) = \frac{1}{2}\left[\left(u\left(t_k, x_{j+1}\right) - u\left(t_k, x_j\right)\right) - \left(u\left(t_k, x_j\right) - u\left(t_k, x_{j-1}\right)\right)\right]. \tag{6.10}$$

Now let Δt and Δx be such that $X = n\Delta x, T = n\Delta t$ and then $\frac{X}{\Delta x} = \frac{T}{\Delta t}$. Then multiplying (6.10) by $\frac{1}{\Delta t}$ we obtain:

$$\frac{1}{\Delta t}\left(u\left(t_{k+1}, x_j\right) - u\left(t_k, x_j\right)\right) = \frac{1}{2\Delta t}\left[\left(u\left(t_k, x_{j+1}\right) - u\left(t_k, x_j\right)\right) - \left(u\left(t_k, x_j\right) - u\left(t_k, x_{j-1}\right)\right)\right]. \tag{6.11}$$

If we take $\Delta t \to 0$ the first term in (6.11) converges to $\partial_t(u)$. For the second term, if we assume that:

$$\Delta t \approx (\Delta x)^2$$

taking into account that $\Delta t = \frac{T\Delta x}{X}$

$$\frac{1}{2\Delta t} = \frac{X}{2T\Delta x}$$

we can conclude that the second term converges to $\partial_{xx}(u)$. So, from the random walks we get a discrete version of the heat equation

$$\partial_t(u) = \frac{1}{2}\partial_{xx}(u)$$

As an example, consider the random walk with step $\Delta x = \sqrt{\Delta t}$, that is:

$$\begin{cases} x_{j+1} = x_j \pm \sqrt{\Delta t} \\ x_0 = 0 \end{cases}$$

We claim that the expected value after n steps is zero. The stochastic variables $x_{j+1} - x_j = \pm\sqrt{\Delta t}$ are independents and so

$$E(x_n) = VE\left(\sum_{j=0}^{n-1}(x_{j+1} - x_j)\right) = \sum_{j=0}^{n-1} VE(x_{j+1} - x_j) = 0$$

and

$$Var(x_n) = \sum_{j=0}^{n-1} Var(x_{j+1} - x_j) = \sum_{j=0}^{n-1} \Delta t = n\Delta t = T.$$

If we interpolate the points $\{x_j\}_{1\le j\le n-1}$ we obtain

$$x(j) = \frac{x_{j+1} - x_j}{\Delta t}(t - t_j) + x_j \tag{6.12}$$

for $t_j \le t \le t_{j+1}$. (6.12) is a *Markovian process*, because:

1. $\forall a > 0, \{x(t_k + a) - x(t_k)\}$ is independent of the history $\{x(s) : s \le t_k\}$
2.

$$\begin{aligned} E(x(t_k + a) - x(t_k)) &= 0 \\ Var(x(t_k + a) - x(t_k)) &= a \end{aligned}$$

Remark 6.2.2. *If $\Delta t \ll 1$ then $x \approx N(0, \sqrt{a})$ (i.e normal distributed with mean o and variance a) and then,*

$$P(x(t + a) - x(t) \ge y) \approx \frac{1}{\sqrt{2\pi}} \int_y^\infty e^{-\frac{t^2}{2a}} dt.$$

This is due to the Central Limit Theorem that guarantees that if N is large enough, the distribution can be approximated by Gaussian.

In the next section, we will study the pricing of options when the volatility varies. We will consider models of the form,

$$dS = \mu S dt + \sigma S dZ.$$

The parameter σ is the volatility. When $\sigma = 0$ the model is deterministic. The parameters μ and σ depend on the asset and we have to estimate them. We can estimate them from the historical data for the assets.

We present some examples as follows:

1. Consider the equation,

$$\begin{cases} u'' = 1 \\ u(-1) = u(1) = 0 \end{cases}$$

We fix $x \in (-1, 1)$ and define the stochastic process Z (Brownian motion) with $Z(0) = x$, then since the stochastic process Z does not depend on t , $u'' = 1$ and applying Itô's lemma we obtain:

$$du = du(Z(t)) = u'dZ + \frac{1}{2}u''dt = u'dZ + \frac{1}{2}dt.$$

Next, integrating between 0 and T, we obtain:

$$u(Z(T)) - u(x) = \int_0^T u'dZ + \int_0^T \frac{1}{2}dt.$$

If we let T_x be the time so that the process arrives at a boundary then it is possible to prove that this happens with probability one. Thus, using T_x instead of T we have that $u(Z(T_x)) = 0$ and we can therefore conclude that:

$$u(x) = -\frac{1}{2}E_x(T_x). \tag{6.13}$$

As a consequence we know the expected value of the time the process arrives at the boundary, because from the solution of our differential equation, we have

$$u = -\frac{1}{2}\left(1 - x^2\right). \tag{6.14}$$

Comparing equation 6.13 to equation 6.14, we observe that

$$E_x(T_x) = 1 - x^2$$

In general, the same computation can be performed for the equation in $\mathbb{R}^n$

$$\begin{cases} \Delta u = 1 \\ u|_{\partial\Omega} = 0 \end{cases}$$

In this case, we define a Brownian motion in $\mathbb{R}^n$ with multivariate normal distribution.

2. Consider the equation,

$$\begin{cases} u'' = 0 \\ u(-1) = a; u(1) = b \end{cases}$$

Let Z be a Brownian motion so that $Z(0) = x$. In the same way as in the previous example, applying Itô's lemma to $u(Z(t))$ we obtain:

$$du = u'dZ.$$

Integrating between 0 and T results in,

$$u(Z(t)) - u(x) = \int_0^T u' dZ.$$

We recall that $Z(T) \sim N(0,T)$. Therefore the time for arriving at the boundary is T_x, hence $Z(T_x) = \pm 1$. This shows that the solution is linear.

3. Consider the equation

$$u'' = 0$$

with initial point in $x \in (x-r, x+r)$. As in the previous examples: because we are considering a Brownian motion, the probability of arriving at the boundary is $\frac{1}{2}$ and so the solution of the equation is

$$u(x) = \frac{u(x+r) - u(x-r)}{2}.$$

We recall that if we consider a sphere centered at x with radius r i.e. $S_r(x)$ then the probability of the arriving time is uniform. Therefore since u is represented by its expected value, we obtain the mean value theorem:

$$u(x) = \int_{S_r(x)} u \frac{1}{|S_r(x)|} = \frac{1}{|S_r(x)|} \int_{S_r(x)} u.$$

In the next section we briefly discuss how the tree methodology can be used for assets paying dividends.

6.3 Tree methods for dividend paying assets

In this section we discuss tree modifications to accommodate the case when the underlying asset pays continuous dividends, known discrete dividends as well as known cash dividends at a pre-specified time point t.

6.3.1 Options on assets paying a continuous dividend

Suppose an asset pays a dividend rate δ per unit of time. For example, suppose that the equity pays $\delta = 1\%$ annual dividend. If the original model used is geometric Brownian motion, applying Girsanov's theorem to (6.1), we obtain the stochastic equation:

$$dS_t = (r - \delta) S_t dt + \sigma S_t dW_t. \tag{6.15}$$

Note that equation (6.15) is exactly a geometric Brownian motion with the riskfree rate r replaced with $r - \delta$. Since everything stays the same, any tree constructed for a stochastic process with r for the drift will work by just replacing it with $r - \delta$. For example, the Trigeorgis tree in this case becomes:

$$\Delta x = \sqrt{\sigma^2 \Delta t + (r - \delta - \frac{1}{2}\sigma^2)^2 \Delta t^2}$$
$$p_u = \frac{1}{2} + \frac{1}{2}\frac{(r - \delta - \frac{1}{2}\sigma^2)\Delta t}{\Delta x}$$
$$p_d = 1 - p_u.$$

In our example $\delta = 0.01$. We can obviously write a routine for a general delta and in the case when there are no dividends just set $\delta = 0$. As an observation the unit time in finance is always 1 year. Therefore in any practical application if for instance expiry is one month we need to use $T = 1/12$.

6.3.2 Options on assets paying a known discrete proportional dividend

In this instance, the asset pays at some time τ in the future, a $\hat{\delta}$ dividend amount which is proportional to the stock value at that time τ. Specifically, the dividend amount is $\hat{\delta} S_\tau$. Due to the way the binomial tree is constructed we need a very simple modification to accommodate the dividend payment. We only care about the dividend payment if it happens during the option lifetime. If that is the case suppose $\tau \in [(i-1)\Delta t, i\Delta t]$.

When the dividend is paid the value of the asset drops by that particular amount. The justification is simple - the asset share value represents the value of the company. Since a ceratin amount is paid per share to the shareholders - an entity outside the company - the value of the company drops by that exact amount. To accommodate this we change the value of all nodes at time $i\Delta t$ by multiplying with $1 - \hat{\delta}$ for a multiplicative tree in S_t and by adding $-\hat{\delta}$ for an additive tree. Specifically for multiplicative trees at node $(i, j) = (i\Delta t, S_j)$, the value is $S_0(1 - \hat{\delta})u^j d^{i-j}$ where $u = e^{\Delta X_u}$ and $d = e^{\Delta X_d}$.

6.3.3 Options on assets paying a known discrete cash dividend

This is the most complex case and unfortunately this is the realistic case (most assets pay cash dividends). In this case, the major issue is that the tree becomes non-recombining after the dividend date. Suppose $\tau \in [(i-1)\Delta t, i\Delta t]$ and at that time the stock pays cash dividend D. The value of nodes after this time are supposed to be subtracted by D. Specifically at the node value (i, j) we have:

$$S_0(e^{\Delta X_u})^j (e^{\Delta X_d})^{i-j} - D.$$

At the next step the successor down will be $S_0(e^{\Delta X_u})^j (e^{\Delta X_d})^{i-j-1} - De^{\Delta X_d}$. On the other hand the successor up from the lower node: $S_0(e^{\Delta X_u})^{j-1} (e^{\Delta X_d})^{i-j-1} -$

D will be $S_0(e^{\Delta X_u})^j(e^{\Delta X_d})^{i-j-1} - De^{\Delta X_u}$. It is easy to see that the two nodes will not match unless $e^{\Delta X_u} = e^{\Delta X_d} = 1$ which makes the tree a straight line. Therefore, after step i the tree becomes non-recombining and for example at time $(i+m)\Delta t$ there will be $m(i+1)$ nodes rather than $i+m+1$ as in regular tree. That is a quadratic number of calculations that quickly become unmanageable.

The trick to deal with this problem is to make the tree non-recombining in the beginning rather than at the end. Specifically, we assume that S_t has 2 components i.e.

- $\tilde{S}_t$ a random component
- remainder depending on the future dividend stream:

$$\tilde{S}_t = \begin{cases} S_t & \text{when} \quad t > \tau \\ S_t - De^{-r(\tau-t)} & \text{when} \quad t \leq \tau \end{cases}$$

Suppose $\tilde{S}_t$ follows a geometric Brownian motion with $\tilde{\sigma}$ constant. We calculate $p_u, p_d, \Delta X_u, \Delta X_d$ in the usual way with σ replaced with $\tilde{\sigma}$. The tree is constructed as before however the tree values are:

$$\begin{aligned} &\text{at } (i,j), \text{ when} \quad t = i\Delta t < \tau : \tilde{S}_t(e^{\Delta X_u})^j(e^{\Delta X_d})^{i-j} + De^{-r(\tau-t)} \\ &\text{at } (i,j), \text{ when} \quad t = i\Delta t \geq \tau : \tilde{S}_t(e^{\Delta X_u})^j(e^{\Delta X_d})^{i-j}. \end{aligned}$$

This tree will be mathematically equivalent with a discretization of a continuous Brownian motion process that suddenly drops at the fixed time τ by the discrete amount D.

6.3.4 Tree for known (deterministic) time varying volatility

Suppose the stochastic process has a known time varying volatility $\sigma(t)$ and drift $r(t)$. That is the stock follows:

$$dS_t = r(t)S_t dt + \sigma(t)S_t dW_t$$

Fixing a time interval Δt, suppose at times $0, \Delta t, 2\Delta t, \ldots, \mu\Delta t$ the volatility values are

$$\sigma(i\Delta t) = \sigma_i, r(i\Delta t) = r_i$$

Then, the corresponding drift term for the log process $X_t = \log S_t$ is

$$\nu_i = r_i - \frac{\sigma_i^2}{2}.$$

We keep the $\Delta X_u, \Delta X_d$ fixed ($\Delta X_u = \Delta X; \Delta X_d = -\Delta X$), this insures that the tree is recombining. However, we vary the probabilities at all steps so that the tree adapts to the time varying coefficients. We let the probabilities at time step i be denoted using $p_u^i = p_i$ and $p_d^i = 1 - p_i$. Thus, we need to have

$$\begin{cases} p_i \Delta x - (1-p_i)\Delta x & = \nu_i \Delta t_i \\ p_i \Delta x^2 + (1-p_i)\Delta x^2 & = \sigma_i^2 \Delta t_i + \nu_i^2 \Delta t_i^2 \end{cases} \tag{6.16}$$

Simplifying (6.16) we obtain

$$\begin{cases} 2p_i \Delta x - \Delta x & = \nu_i \Delta t_i \\ \Delta x^2 & = \sigma_i^2 \Delta t_i + \nu_i^2 \Delta t_i^2 \end{cases} \tag{6.17}$$

Equation (6.17) can be further simplified to:

$$p_i = \frac{1}{2} + \frac{\nu_i \Delta t_i}{2\Delta x} \tag{6.18}$$

Rearranging terms in (6.18) we obtain:

$$\nu_i^2 \Delta t_i^2 + \sigma_i^2 \Delta t_i - \Delta x^2 = 0 \tag{6.19}$$

Using the quadratic formula we solve for Δt_i to obtain

$$\Delta t_i = \frac{-\sigma_i^2 \pm \sqrt{\sigma_i^4 + 4\nu_i^2 \Delta x^2}}{2\nu_i^2} \tag{6.20}$$

From (6.18), we have that

$$\Delta x^2 = \sigma_i^2 \Delta t_i + \nu_i^2 \Delta t_i^2 \tag{6.21}$$

So now the issue is that Δx must be kept constant but obviously the i's will pose a problem.

So here are two possible approaches to this issue of determining the proper parameter values.

If we sum over i in the last expression, we obtain

$$N\Delta x^2 = \sum_i^N \sigma_i^2 \Delta t_i + \sum_i^N \nu_i^2 \Delta t_i^2 \tag{6.22}$$

Dividing both sides by N we obtain:

$$\Delta x^2 = \frac{1}{N}\sum_i^N \sigma_i^2 \Delta t_i + \frac{1}{N}\sum_i^N \nu_i^2 \Delta t_i^2$$

Taking $\overline{\Delta t} = \frac{1}{N}\sum_i^N \Delta t_i$ and solving for Δx we obtain:

$$\Delta x = \sqrt{\overline{\sigma^2}\,\overline{\Delta t} + \overline{\nu^2}\Delta t^2} \tag{6.23}$$

where $\overline{\sigma^2} = \frac{1}{N}\sum_{i=1}^{N}\sigma_i^2$ and $\overline{\nu^2} = \frac{1}{N}\sum_{i=1}^{N}\nu_i^2$. These are approximate values. From the practical perspective we just created a circular reasoning (we need the Δt_i values to calculate the Δx to calculate the Δt_i values.

So to actually implement it one needs to initialize the algorithm with $\Delta t_i = T/n$ for all i. Then calculate Δx using (6.23). Then repeat

1. Calculate all Δt_i's using equation (6.20)
2. Recompute Δx using (6.23)
3. Calculate the difference between two consecutive values of Δx. If smaller than ε stop. If larger repeat steps 1-3.

The tree thus constructed is approximately correct. However, the effects of the extra modifications to the value of the option are small.

6.4 Pricing path dependent options: Barrier Options

Barrier options are triggered by the action of the underlying asset hitting a prescribed value at some time before expiry. For example, as long as the asset remains below a predetermined barrier price during the whole life of the option, the contract will have a call payoff at expiry. Barrier options are clearly path dependent options. Path dependent options are defined as the right, but not the obligation, to buy or sell an underlying asset at a predetermined price during a specified time period, however the exercise time or price is dependent on the underlying asset value during all or part of the contract term. A path dependent option's payoff is determined by the path of the underlying asset's price.

There are two main types of barrier options:

1. The **In** type option, that exercises as specified in the contract as long as a particular asset level called barrier is reached before expiry. If the barrier is reached then the option is said to have knocked in. If the barrier level is not reached the option is worthless.
2. The **Out** type option, that becomes zero and is worthless if the Barrier level is not reached. If the barrier is reached then the option is said to have knocked out.

We further characterize the barrier option by the position of the barrier relative to the initial value of the underlying:

1. If the barrier is above the initial asset value, we have an Up option.
2. If the barrier is below the initial asset value, we have a Down option.

We note that if an Out option starts as regular American and barrier is hit option becomes zero and similarly that an In option typically starts worthless and if the barrier is hit the option becomes a regular American option.

Given a barrier B and path = $S_{t1}, \ldots, S_{tN}$, the terminal payoff of the barrier options can be written as:

1. Down and Out Call $(S_T - K)_+ 1_{\{\min(S_{t1},\ldots,S_{tN})>B\}}$
2. Up and Out Call $(S_T - K)_+ 1_{\{\max(S_{t1},\ldots,S_{tN})<B\}}$
3. Down and In Call $(S_T - K)_+ 1_{\{\min(S_{t1},\ldots,S_{tN})\leq B\}}$
4. Up and In Call $(S_T - K)_+ 1_{\{\max(S_{t1},\ldots,S_{tN})\geq B\}}$
5. Down and Out Put $(K - S_T)_+ 1_{\{\min(S_{t1},\ldots,S_{tN})>B\}}$
6. Up and Out Put $(K - S_T)_+ 1_{\{\max(S_{t1},\ldots,S_{tN})<B\}}$
7. Down and In Put $(K - S_T)_+ 1_{\{\min(S_{t1},\ldots,S_{tN})\leq B\}}$
8. Up and In Put $(K - S_T)_+ 1_{\{\max(S_{t1},\ldots,S_{tN})\geq B\}}$

Remark 6.4.1. *Generally, the In type options are much harder to solve than Out type options. However, it is easy to see that we have the following Put-Call parity relations.*

Down and Out Call (K, B, T) + Down and In Call $(K, B, T) =$ Call (K, T) for all B.

This is easy to prove by looking at the payoffs and remarking that:

$$1_{\min(S_{t1},\ldots,S_{tN})>B} + 1_{\min(S_{t1},\ldots,S_{tN})\leq B} = 1,$$

and recalling that option price at t is the discounted expectation of the final payoff. Therefore applying expectations will give exactly the relation needed. Thus one just needs to construct a tree method for Down type options and use the In-Out Parity above to obtain the price of an In type option.

6.5 Trinomial tree method and other considerations

The trinomial trees provide an effective method of numerical calculation of option prices within the Black-Scholes model. Trinomial trees can be built in a similar way to the binomial tree. To create the jump sizes u and d and the transition probabilities p_u and p_d in a binomial tree model we aim to match these parameters to the first two moments of the distribution of our geometric Brownian motion. The same can be done for our trinomial tree for u, d, p_u, p_m, p_d.

For the trinomial tree, one cannot go up and down by different amounts to keep it recombining. The condition imposed is the fact that the discrete increment needs to match the continuous once that is:

$$E(\Delta x) = \Delta x p_u + 0 p_u + (-\Delta x) p_d = D dt, \quad D = r - \frac{\sigma^2}{2}$$
$$E(\Delta x^2) = \Delta x^2 p_u + 0 p_u + (-\Delta x)^2 p_d = \sigma^2 \Delta t + D^2 \Delta t^2$$
$$p_u + p_m + p_d = 1$$

These are 3 equations and 4 unknowns. So it has an infinite number of solutions. However we have that:

$$p_u = \frac{1}{2}\left(\frac{\sigma^2\Delta t + D^2\Delta t^2}{\Delta x^2} + \frac{D\Delta t}{\Delta x}\right)$$
$$p_m = 1 - \frac{\sigma^2\Delta t + D^2\Delta t^2}{\Delta x^2}$$
$$p_d = \frac{1}{2}\left(\frac{\sigma^2\Delta t + D^2\Delta t^2}{\Delta x^2} - \frac{D\Delta t}{\Delta x}\right)$$

These probabilities need to be numbers between 0 and 1. Imposing this condition we obtain a sufficient condition: $\Delta x \geq \sigma\sqrt{3\Delta t}$. This condition will be explained later in this section. Any Δx with this property produces a convergent tree.

The trinomial tree is an alternate way to approximate the stock price model. The stock price once again follows the equation:

$$dS_t = rS_t dt + \sigma S_t dW_t. \tag{6.24}$$

In [37] the authors work with a continuously paying dividend asset and the drift in the equation is replaced by $r - \delta$. All the methods we will implement require an input r. One can easily obtain the formulas for a continuously paying dividend asset by just replacing this parameter r with $r - \delta$. Once again, it is equivalent to work with the return $X_t = \log S_t$ instead of directly with the stock and we obtain:

$$dX_t = \nu dt + \sigma dW_t, \quad \text{where } \nu = r - \frac{1}{2}\sigma^2. \tag{6.25}$$

The construction of the trinomial tree is equivalent to the construction of the binomial tree described in previous sections. A one step trinomial tree is presented below:

$$\begin{array}{cc} & S_u \\ S & S \\ & S_d \end{array}$$

Trinomial trees allows the option value to increase, decrease or remain stationary at every time step as illustrated above.

Once again we match the expectation and variance. In this case the system contains three equations and three unknowns so we do not have a free choice as in the binomial tree case. In order to have a convergent tree, numerical experiments have shown that we impose a condition such that

$$\Delta x \geq \sigma\sqrt{3\Delta t} \tag{6.26}$$

Hence for stability there must be restrictions on the relative sizes of Δx and Δt. The condition ensures that our method is stable and converges to the exact solution. Please refer to [37] for details about the stability condition. Once the tree is constructed we find an American or European option value by stepping back through the tree in a similar manner with what we did for the binomial tree. The only difference is that we calculate the discounted expectation of three node values instead of two as we did for the binomial tree. The main advantage of the trinomial tree over the binomial tree construction is the fact that the trinomial tree is appropriate for pricing options when the volatility varies. This is because when we vary the volatility, we introduce other parameters in the model that increases the number of equations which is easily solved by the trinomial tree construction method.

The trinomial tree produces more paths (3^n) than the binomial tree (2^n). Surprisingly the order of convergence is not affected for this extra number. In both cases the convergence of the option values is of the order $O(\Delta x^2 + \Delta t)$. Optimal convergence is always guaranteed due to condition(6.26).

Condition (6.26) makes a lot of difference when we deal with barrier options i.e. options that are path dependent. This is due to the fact that the trinomial tree contains a larger number of possible nodes at each time in the tree. The trinomial tree is capable of dealing with the situations when the volatility changes over time i.e., is a function of time. For a detailed example of recombining trinomial tree for valuing real options with changing volatility, please refer to [86]. In that study, the trinomial tree is constructed by choosing a parameterization that sets a judicious state space while having sensible transition probabilities between the nodes. The volatility changes are modeled with the changing transition probabilities while the state space of the trinomial tree is regular and has a fixed number of time and underlying asset price levels.

The meaning of little o and big O Suppose we have two functions f and g. We say that f is of order little o of g at x_0:

$$f \sim o(g) \Leftrightarrow \lim_{x \to x_0} \frac{f(x)}{g(x)} = 0$$

We say that f is of order big O of g at x_0:

$$f \sim O(g) \Leftrightarrow \lim_{x \to x_0} \frac{f(x)}{g(x)} = C$$

where C is a constant.

In our context if we calculate the price of an option using an approximation (e.g., trinomial tree) call it $\hat{\Pi}$, and the real (unknown) price of the option call it Π, then we say that the approximation is of the order $O(\Delta x^2 + \Delta t)$ and we mean that:

$$|\hat{\Pi} - \Pi| = C(\Delta x^2 + \Delta t)$$

whenever Δx and Δt both go to zero for some constant C.

6.6 Markov Process

The remaining sections in this chapter are dedicated to a general theory of approximating solutions of Stochastic Differential Equations using trees. We shal present in the final section of this chapter what we call the quadrinomial tree approximation for stochastic volatility models. The full version of the presentation may be found in [70].

Markov processes were briefly introduced in chapter 1 section 1.4 of this book. Here we give a formal definition.

Definition 6.6.1 (Markov process)**.** Let $\{X_t\}_{t\geq 0}$ be a process on the space $(\Omega, \mathcal{F}, \mathcal{P})$ with values in E and let $\mathcal{G}_t$ a filtration such that X_t is adapted with respect to this filtration. In the most basic case the filtration is generated by the stochastic process itself: $\mathcal{G}_t = \mathcal{F}_t^x = \sigma(X_s : s \leq t)$.

The process X_t is a Markov process if and only if:

$$\mathcal{P}(X_{t+s} \in \Gamma | \mathcal{G}_t) = \mathcal{P}(X_{t+s} \in \Gamma | X_t), \forall s, t \geq 0 \text{ and } \Gamma \in \mathcal{B}(E). \tag{6.27}$$

Here, the collection $\mathcal{B}(E)$ denote the Borel sets of E.

Essentially, the definition says in order to decide where the process goes next knowing just the current state is the same as knowing the entire set of past states. The process does not have memory of past states.

The defining equation (6.27) is equivalent to

$$E\left[f(X_{t+s})|\mathcal{F}_t^X\right] = E\left[f(X_{t+s})|X_t\right],$$

$\forall$ f Borel measurable functions on E.

Now this definition looks very complicated primarily because of those spaces and the daunting probability distribution: $\mathcal{P}(X_{t+s} \in \Gamma | \mathcal{G}_t)$. We wrote it this way to cover any imaginable situation. In practice we use a transition function to define this transition distribution.

6.6.1 Transition function

Remember that a stochastic process is completely characterized by its distribution. Since knowing the current state is enough to determine the future, the random variable $X_t|X_s$ for any s and t determines the process trajectories. This random variable $X_t|X_s$ has a distribution and in fact that is what we denoted with $\mathcal{P}(X_t \in \Gamma|X_s)$. If this distribution has a density we will call it **the transition density function**.

Specifically, the random variable $X_t|X_s = x$, has a density denoted $p(s,t,x,y)$, that is:

$$\mathcal{P}(s,t,x,\Gamma) = \int_\Gamma p(s,t,x,y)dy.$$

This transition density completely characterizes the Markov process. Note that in general the transition probability for a Markov process depends on the time of the transition (s), and it is a 4 dimensional function. Such process appears as the solution of a general SDE such as:

$$dS_t = \mu(t, S_t)dt + \sigma(t, S_t)dB_t. \tag{6.28}$$

However, in finance we work with simplified SDE's where the drift and diffusion functions $\mu(\cdot,\cdot)$ and $\sigma(\cdot,\cdot)$ are not dependent on time t. The solution of a SDE where the coefficients are not time dependent are a particular type of Markov process.

Definition 6.6.2 (Homogeneous Markov process)**.** A Markov process is homogeneous if the transition function does not depend on the time when the transition happens, just on the difference between points. Mathematically, the random variable $X_t|X_s$ has the same distribution as $X_{t-s}|X_0$ for all s and t, or written even clearer: $X_{s+h}|X_s$ is the same as distribution of $X_h|X_0$. This should make it clear that the point when the transition is made s is irrelevant for homogeneous processes. In terms of transition function we have:

$$p(s,t,x,y) = p(t-s,x,y)$$

or in the other notation:

$$p(s, s+h, x, y) = p(h, x, y).$$

As we mentioned, all solutions to diffusion equations of the type:

$$dS_t = \mu(S_t)dt + \sigma(S_t)dB_t. \tag{6.29}$$

where μ and σ are functions that do not have a t argument, are homogeneous Markov processes.

Formally, a function $\mathcal{P}(t,x,\Gamma) = \int_\Gamma p(t,x,y)dy$ defined on $[0,\infty) \times E \times \mathcal{B}(E)$ is a (time homogeneous) transition function if:

1. $\mathcal{P}(t,x,\cdot)$ is a probability measure on $(E, \mathcal{B}(E))$.

2. $\mathcal{P}(0,x,\cdot) = \delta_x$ (Dirac measure).

3. $\mathcal{P}(\cdot,\cdot,\Gamma)$ is a Borel measurable function on $[0,\infty) \times E$.

4. The function satisfies the Chapman-Kolmogorov equation:

$$\mathcal{P}(t+s, x, \Gamma) = \int \mathcal{P}(s, y, \Gamma)\mathcal{P}(t, x, dy),$$

where $s, t \geq 0, x \in E$ and $\Gamma \in \mathcal{B}(E)$.

The Chapman-Kolmogorov equation is a property of all Markov processes basically saying that to go from x to some point in Γ in $t+s$ units of time one has to go through some point at time t. Rewriting the expression using the transition function:

$$p(t+s, x, y) = \int p(s, z, y)p(t, x, z)dz,$$

where the integral in z is over all the state space points.

The connection between a function with such properties and the Markov process is the following. A transition function is the transition function for a time homogeneous Markov process X_t if and only if:

$$\mathcal{P}(X_{t+s} \in \Gamma | \mathcal{F}_t^x) = \mathcal{P}(s, X_t, \Gamma),$$

that means it actually expresses the probability that the process goes into a set Γ at time $t+s$ given that it was at X_t at time t. Furthermore, all these properties may be expressed in terms of the density function $p(t, x, y)$

$$\mathcal{P}(t, x, \Gamma) = \int_\Gamma p(t, x, y)dy,$$

if the transition density function exists.

The next theorem stated without proof insures that our discussion of transition functions is highly relevant.

Theorem 6.6.1. *Any Markov process has a transition function. If the metric space (E, r) is complete and separable then for any transition function there exists a unique Markov process with that transition function.*

If the Markov process is defined on $\mathbb{R}$ (which is a separable space) then the theorem is valid and there is a one to one equivalence between Markov processes and their transition functions. However, $\mathbb{R}^2$ and generally $\mathbb{R}^n$ are not separable spaces. So, there may exist multiple Markov processes with the same transition function. This seemingly disappointing property is in fact very useful and it is the basis on the Markov Chain Monte Carlo (MCMC) methods. Specifically, suppose we have a very complicated process and we need to create Monte Carlo paths but the transition distribution is complicated. So the idea is to map the process in $\mathbb{R}^2$ or higher dimensions then find a simpler process with the same transition distribution. Then use the simpler process to generate the transition distribution of the more complicate one.

Definition 6.6.3 (Strong Markov process)**.** A Markov process $\{X_t\}_{t\geq 0}$ with respect to $\{\mathcal{G}_t\}_{t\geq 0}$ is called a strong Markov process at ζ if:

$$\mathcal{P}(X_{t+\zeta} \in \Gamma | \mathcal{G}_\zeta) = \mathcal{P}(t, X_\zeta, \Gamma) \tag{6.30}$$

where ζ is a stopping time with respect to $\mathcal{G}_t$ and $\zeta < \infty$ almost surely (a.s). A process is strong Markov with respect to $\mathcal{G} = \{\mathcal{G}\}_t$ if equation (6.30) holds for all ζ stopping times with respect to $\{\mathcal{G}_t\}$.

In other words what a strong Markov process has going for it is that the Markov property not only holds at any times but also at any stopping time. This is important since it allows us to use Markov processes in very interesting ways. Clearly any strong Markov process is Markov.

Remark 6.6.1. *Any Markov process is a strong Markov at ζ if ζ is discrete valued.*

Example 6.6.1. *The 1-D Brownian motion is a homogeneous Markov process, with the transition density function:*

$$\mathcal{P}(t, x, \Gamma) = \int_\Gamma \frac{1}{\sqrt{2\pi t}} e^{-\frac{(y-x)^2}{2t}} dy.$$

The Brownian motion is one of the few examples where we can actually write the transition distribution explicitly. In general, it is not possible to calculate these transition probabilities exactly. The next section introduces a generalization that will allow us to tap into a much older and more developed theory.

6.7 Basic elements of operators and semigroup theory

To be able to better characterize Markov processes we need to present some elements of semigroup theory. To this end, using the transition probability we define an operator

$$T(t)f(x) = \int f(y)\mathcal{P}(t, x, dy) = \int f(y)p(t, x, y)dy.$$

Note that $T(t)$ is a functional operator. Specifically, it takes as the argument a function $f(x)$ and outputs another function $T(t)f(x)$ (which is a function of x).

We say that $\{X_t\}$ a Markov process corresponds to a semigroup operator $T(t)$ if and only if:

$$T(t)f(X_s) = \int f(y)\mathcal{P}(t, X_s, dy) = \int f(y)p(t, X_s, y)dy = E[f(X_{t+s})|X_s],$$

for any measurable function f. Recall the definition of transition probability: $p(t, x, y)$ is the density of the random variable $X_t|X_0$ which, by homogeneity of the process, is the same as the distribution of $X_{t+s}|X_s$. This operator $T(t)$ and the initial distribution of the process at time 0 completely characterize the Markov process.

Since $\mathcal{P}(t,x,dy)$ is a transition distribution we can use the Chapman-Kolmo-gorov equation and in this operator notation, we obtain:

$$\begin{aligned}
T(s+t)f(x) &= \int f(y)p(t+s,x,y)dy = \int f(y)\int p(s,z,y)p(t,x,z)dzdy \\
&= \int\int f(y)p(s,z,y)p(t,x,z)dydz \\
&= \int\left(\int f(y)p(s,z,y)dy\right)p(t,x,z)dz \\
&= \int (T(s)f(z))p(t,x,z)dz = T(t)\left(T(s)f(x)\right) \\
&= T(s)\circ T(t)f(x).
\end{aligned}$$

This particular expression tells us that $T(t)$ is a semigroup operator, specifically a contraction operator $||T(t)|| \leq 1$. We shall define exactly what this means next.

Let L be a Banach space of functions. A Banach space is a normed linear space that is a complete metric space with respect to the metric derived from its norm. To clarify some of these notions we will provide several definitions.

Definition 6.7.1 (Linear space/Vector Space)**.** A linear (sometimes called vector space) L is a collection of elements endowed with two basic operations generally termed addition and multiplication with a scalar. A linear space L has the following properties:

1. Commutativity: $f+g=g+f$,
2. Associativity of addition: $(f+g)+h=f+(g+h)$,
3. Null element: there exists an element 0 such that $f+0=0+f=f$, for all $f\in L$,
4. Inverse with respect to addition: for any $f\in L$ there exists an element $-f\in L$ such that $f+(-f)=(-f)+f=0$
5. Associativity of scalar multiplication: $a(bf)=(ab)f$
6. Distributivity: $(a+b)f=af+bf$ and $a(f+g)=af+ag$
7. Identity scalar: There exists a scalar element denoted 1 such that $1f=f1=f$

Throughout the definition these properties hold for all $f,g,h\in L$, and a,b scalars. Scalars are typically 1 dimensional real numbers.

Example 6.7.1. *Most spaces of elements you know are linear spaces. For example, $L = \mathbb{R}^n$, with the scalar space $\mathbb{R}$ is the typical vector space and has all the properties above. L, the space of squared matrices with dimension n and again the scalar space is $\mathbb{R}$ is another classical example. In fact this is sometimes called the general linear space and is the prime object of study for Linear Algebra. Here we shall use L as a space of functions. For example $L = \mathcal{C}(0, \infty)$, the space of all continuous functions defined on the interval $(0, \infty)$. The notions that follow are common in functional analysis which studies spaces of functions and operations on spaces of functions. For example the common derivation is just an operator (operation) on these spaces of functions.*

A normed space in nothing more than a space that has a norm defined on it. The definition of a norm follows:

Definition 6.7.2. Given a linear space L a norm on this space is any function $\| \cdot \| : L \to [0, \infty)$ with the following properties:

1. Positive definite: if $\|f\| = 0$ then f is the 0 element of L
2. Homogeneous: $\|af\| = |a|\|f\|$, for any scalar a and $f \in L$
3. Subadditive (triangle inequality) $\|f + g\| \leq \|f\| + \|g\|$

As a side note a seminorm is a function with properties 2 and 3 only.

Example 6.7.2. *We can introduce many norms on the same space. Take $L = \mathbb{R}^n$ the space of vectors. The classical absolute value*

$$\|f\| = |f| = \sqrt{f_1^2 + f_2^2 + \cdots + f_n^2},$$

where f_i's are the components of f, is a norm. Typically it is denoted $\|f\|_2$. The generalization of this is the p-norm:

$$\|f\|_p = \sqrt[p]{\sum_{i=1}^{n} f_i^p} = \left(\sum_{i=1}^{n} f_i^p\right)^{\frac{1}{p}}.$$

The infinity norm is:

$$\|f\|_\infty = \max_{1 \leq i \leq n} |f_i|.$$

These norms can be defined generally on any linear space. Other spaces for example when L is a space of squared matrices n dimensional have other classical norms. Say A is a matrix we can define:

$$\|A\| = \max_{1 \leq j \leq n} \sum_{i=1}^{n} |a_{ij}|,$$

which is the maximum of sum of elements of each column in the matrix. The Frobenius norm is an equivalent to the absolute value for vectors:

$$\|A\|_F = \sqrt{\sum_{i=1}^{n} \sum_{i=1}^{n} a_{ij}^2}.$$

For a space of functions for example $L = C(0, \infty)$ we can define similar norms for example:

$$\|f\|_0 = \max_{x \in (0,\infty)} f(x)$$

or norm 2:

$$\|f\|_2 = \left(\int_0^\infty f^2(x) dx \right)^{1/2}$$

Throughout this chapter and indeed throughout the constructions involving stochastic processes we will be using L the space of continuous, bounded functions. On this space of functions we will work with operators which essentially are transformations of functions. As a simple example the differential of a function $\frac{\partial}{\partial x}$ is an operator that transform a function into its derivative (itself a function).

Definition 6.7.3 (Semigroup)**.** A one parameter family $\{T(t)\}_{t \geq 0}$ of bounded linear operators is called a semigroup if :

1. $T(0) = I$ (identity operator)
2. $T(s+t) = T(s) \circ T(t) \; \forall t, s \geq 0$

A semigroup is called **strongly continuous** if:

$$\lim_{t \to 0} T(t)f = f \text{ for all } f \in L \quad (\text{equivalently } ||T(t)f - f|| \to 0 \quad as \quad t \to 0) \tag{6.31}$$

A semigroup is called a **contraction** if $||T(t)|| \leq 1$ for all t.

The norm in the definition needs to be specified and the family is a semigroup with respect to that particular norm.

Example 6.7.3. *Let $E = \mathbb{R}^n$ and let B a $n \times n$ matrix.*
Define:

$$e^{tB} = \sum_{k=0}^{\infty} \frac{1}{k!} t^k B^k,$$

where B^k are powers of the matrix which are well defined since the matrix is square and the convention $B^0 = I_n$ the identity matrix.

Then, the operator: $T(t) = e^{tB}$ forms a strongly continuous semigroup and

$$\|e^{tB}\| \leq \sum_{k=0}^{\infty} \frac{1}{k!} t^k \|B^k\| \leq \sum \frac{1}{k!} t^k \|B\|^k \leq e^{t\|B\|},$$

where the norm is the supremum norm.

In general for a strongly continuous semigroup we have $\|T(t)\| \leq Me^{mt}$, $t \geq 0$ for some constants $M \geq 1$ and $m \geq 0$.

In our specific case when the semigroup is defined from a Markov process X_t, i.e.,

$$T(t)f(X_s) = \int f(y)p(t, X_s, y)dy = E[f(X_{t+s})|X_s],$$

we already shown that Chapman-Kolmogorov implies the second defining property of a semigroup. For the first one it is simply:

$$T(0)f(X_s) = E[f(X_s)|X_s] = f(X_s),$$

since $f(X_s)$ is measurable with respect to X_s. For the contraction we have:

$$\|T(t)f(x)\| = \left\| \int f(y)p(t,x,y)dy \right\| \leq \|f(y)\| \| \int p(t,x,y)dy\| = \|f(x)\|.$$

This will imply that $\|T(t)\| \leq 1$. The contraction property is true for any norm and is based on the fact that $\int p(t,x,y)dy = 1$. It is left as an exercise to prove this inequality for various specific norms.

6.7.1 Infinitesimal operator of semigroup

The notion of semigroup is important since it allows us to characterize long behavior $T(s+t)$ by looking at short behavior $T(s)$ and $T(t)$. Think about transition probability for a two year period $T(2)$ as being characterized by transitions in only one year $T(1)$. Also note (very importantly) that this only works for homogeneous Markov processes. If the process is not homogeneous then we have $T(0,2) = T(0,1)T(1,2)$ and even though the two transitions on the right are 1 year long they are different functions that need to be studied separately.

In the case when the process is homogeneous we can extend the idea to go from a shot interval extended to obtain everything. But how short an interval may we characterize? In real analysis if we look at the derivative of a function that tells us at an infinitesimal distance what the next value of the function will be. So the idea here would be: can we define a sort of derivative of an operator? If you remember from calculus the derivative is defined as a limit.

Definition 6.7.4. An *Infinitesimal operator* of a semigroup $\{T(t)\}_{t \geq 0}$ on L is a linear operator defined as:

$$Af = \lim_{t \to 0} \frac{1}{t}(T(t)f - f).$$

The domain of definition is those functions for which the operator exists (i.e., $D(A) = \{f | Af \quad \text{exists}\}$). The infinitesimal generator can be thought in some sense as the right derivative of the function: $t \to T(t)f$.

If you remember that a strong continuous semigroup has $\lim_{t \to 0} T(t)f = f$ then the infinitesimal generator can be exactly interpreted as the derivative. The next theorem formalizes this case

Theorem 6.7.1. *If A is the infinitesimal generator of $T(t)$ and $T(t)$ is strongly continuous then the integral $\int_0^t T(s)f ds \in D(A)$ and we have:*

1. *If $f \in L$ and $t \geq 0$ then*
$$T(t)f - f = A\int_0^t T(s)f ds$$
2. *If $f \in D(A)$, then*
$$\frac{d}{dt}T(t)f = AT(t)f = T(t)Af$$
where $T(t)f \in D(A)$
3. *$f \in D(A)$, then*
$$T(t)f - f = \int_0^t AT(s)f ds = \int_0^t T(s)Af ds$$

6.7.2 Feller semigroup

A Feller semigroup is a strongly continuous, positive, contraction semigroup defined by:
$$T(t)f(x) = \int f(y)\mathcal{P}(t, x, dy) \quad \forall \quad t, x$$
with a corresponding infinitesimal generator A. A Feller semigroup's infinitesimal generator is defined for all continuous bounded functions.

This $T(t)$ corresponds to a special case of a homogeneous Markov process with transition $\mathcal{P}(t, x, dy)$ for which the infinitesimal generator exists. The Markov process with these properties is called a Feller process. Recall, if the Feller process is denoted X_t we have:
$$T(t)f(X_s) = E[f(X_{t+s})|\mathcal{F}_s^x] = \int f(y)\mathcal{P}(t, X_s, dy) = \int f(y)p(t, X_s, y)dy.$$

The next theorem makes the connection between the infinitesimal generator of a Markov process and a corresponding martingale expressed in terms of the original process. The big deal about the Feller processes is that if X_t is Feller, the next theorem works for any function f. Often the domain $D(A)$ of the infinitesimal generator of a regular Markov process is hard to find and thus describe.

Theorem 6.7.2. *Let X be a Markov process with generator A.*

1. *If $f \in D(A)$ then $M_t = f(X_t) - f(X_0) - \int_0^t Af(X_0)ds$ is a martingale.*
2. *If $f \in C_0(E)$ and there exists a function $g \in C_0(E)$ such that $f(X_t) - f(X_0) - \int_0^t g(X_s)ds$ is a martingale then $f \in D(A)$ and $Af = g$.*

Proof. In this proof for simplicity of notations we neglect $f(X_0)$. We have:

$$\begin{aligned} E(M_{t+s}|\mathcal{F}_t) &= E[f(X_{t+s})|\mathcal{F}_t] - E\left[\int_0^{t+s} Af(X_u)du|\mathcal{F}_t\right] \\ &= T(s)f(X_t) - \int_0^t Af(X_u)du - \int_t^{t+s} E[Af(X_u)|\mathcal{F}_t]du \\ &= T(s)f(X_t) - \int_0^t Af(X_u)du - \int_0^s T(z)Af(X_t)dz. \end{aligned}$$

But A is an infinitesimal generator and so,

$$T(t)f - f = \int_0^t T(s)Afds,$$

which implies that

$$f = T(t)f - \int_0^t T(s)Afds$$

Substituting we obtain,

$$E(M_{t+s}|\mathcal{F}_t) = T(s)f(X_t) - \int_0^s T(z)Af(X_t)dz - \int_0^t Af(X_u)du = M_t.$$

where $f(X_t) = T(s)f(X_t) - \int_0^s T(z)Af(X_t)dz$. Note there is an $f(X_0)$ missing from the expression, that is the one we neglected.

We finally set

$$T(t)f - f = \int_0^t T(s)gds,$$

which is the property of the infinitesimal generator. □

6.8 General Diffusion process

In this section, we briefly describe a general diffusion process and present some useful examples. We are finally in position to connect with with the semigroup theory we have been describing. The martingale representation theorem 6.7.2 is the connector with option pricing theory.

Let $a = (a_{ij})_{ij}$ be a continuous symmetric, nonnegative definite $d \times d$ matrix valued function on $\mathbb{R}^d$. Let $b : \mathbb{R}^d \to \mathbb{R}^d$ be a continuous function. Define:

$$Af = \frac{1}{2}\sum_{i,j=1}^{d} a_{ij}\frac{\partial}{\partial x_i}\frac{\partial}{\partial x_j}f + \sum_{i=1}^{d} b_i\frac{\partial}{\partial x_i}f, \text{ for all } f \in C_c^\infty(\mathbb{R}^d).$$

In general, for non-homogeneous process we need to define:

$$A_t f(x) = \frac{1}{2}\sum_{i,j=1} a_{ij}(t,x)\partial_{x_i}\partial_{x_j}f(x) + \sum_{i=1}^{d} b_i(t,x)\partial_{x_i}f(x).$$

A Markov process is a diffusion process with infinitesimal generator L if it has continuous paths and

$$E[f(X_t) - f(X_0)] = E\left[\int_0^t A(f(X_s))ds\right].$$

Conversely, if $\{X_t\}$ has continuous paths its generator is given by:

$$Af = c(x)f + \sum_i b_i(x)\frac{\partial f}{\partial x_i} + \sum_{ij} a_{ij}(x)\frac{\partial^2 f}{\partial x_i \partial x_j}.$$

Given the connection above, we use Itô's lemma to give a general characterization for diffusion processes.

Definition 6.8.1. If X_t solves a $d-$dimensional stochastic differential equation:

$$dX_t = b(X_t)dt + \sigma(X_t)dB_t$$

where b is a vector and σ is a $d \times d$ matrix $(\sigma_{ij})_{i,j}$ then X_t is a diffusion (Feller) process with infinitesimal generator:

$$A = \sum_{j=1}^{d} b_j(X_t)\frac{\partial}{\partial x_j} + \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d}\sum_{k=1}^{d}\sigma_{ik}(X_t)\sigma_{kj}(X_t)\frac{\partial^2}{\partial x_i \partial x_j}.$$

Note that writing $a_{ij} = \sum_{k=1}^{d}\sigma_{ik}\sigma_{kj}$, puts the process in the classical form presented above.

Next, we present some useful examples of diffusion processes.

Example 6.8.1 (Pure jump Markov process)**.** *Let X_t be a compound Poisson process, with λ the jump intensity function and μ the probability distribution of jumps. That is in the interval $[0, t]$ the process jumps N times where N is a Poisson random variable with mean λt. Each times it jumps it does so with a magnitude Y with distribution μ. Each jump is independent. The process X_t cumulates (compounds) all these jumps. Mathematically:*

$$X_t = \sum_{i=1}^{N} Y_i.$$

This process is a Markov process and its infinitesimal generator is:

$$Af(x) = \lambda(x)\int (f(y) - f(x))\mu(x, dy)$$

Example 6.8.2 (Lévy process)**.** *Levy processes will be discussed in details in chapter 12 of this book. Their infinitesimal generator has the form:*

$$\begin{aligned} A_t f &= \frac{1}{2}\sum_{ij} a_{ij}(t,x)\partial_i\partial_j f(x) + \sum_i b_i(t,x)\partial_i f(x) \\ &+ \int_{\mathbb{R}^d}\left(f(x+y) - f(x) - \frac{y \cdot \nabla f(x)}{1+|y|^2}\right)\mu(t, x; dy) \end{aligned}$$

In the case when jumps are not time dependent, or dependent on the current state of the system (i.e., μ is independent of (t,x)) then the integral is

$$\int_{\mathbb{R}} \left(f(x+y) - f(x) - \frac{y \cdot \bigtriangledown f(x)}{1+|y|^2} \right) \mu(dy)$$

A more specific example is presented below.

Example 6.8.3. *Suppose a stock process can be written as:*

$$dS_t = rS_t dt + \sigma S_t dW_t + S_t dJ_t,$$

where J_t is a compound Poisson process. This model is knows as a jump diffusion process.

Let

$$X_t = \log S_t$$

then using Itô lemma, under the equivalent martingale measure:

$$dX_t = \mu dt + \sigma dB_t + dJ_t,$$

where μ is $r - \frac{\sigma^2}{2} + \lambda(1 - E(e^Z))$, Z is the random variable denoting the jump rate. The process

$$J_t = \sum_i^{N_t} Z_i,$$

where $N_t \sim$ Poisson(λt) and Z_i are the jump rates. In this case we can write the infinitesimal generator:

$$Af(x) = \frac{1}{2}\sigma^2 \frac{\partial^2 f}{\partial x^2} + \mu \frac{\partial f}{\partial x} + \lambda \int_{\mathbb{R}} \left[f(x+z) - f(x) \right] p(z) dz,$$

where $p(z)$ is the distribution of jumps Z_i.

Below are some specific examples of the jump distributions $p(z)$.

Merton:

$$p(z) = \frac{1}{\sqrt{2\pi s^2}} e^{-\frac{(z-\mu)^2}{2s^2}}$$

and

$$E(e^z) = e^{\mu + \frac{\sigma^2}{2}}$$

Kou:

$$p(z) = py_1 e^{-y_1 z} 1_{\{z \geq 0\}} + (1-p) y_2 e^{y_2 z} 1_{\{z \leq 0\}} \quad \text{(double exponential)}$$

and

$$\begin{aligned} E(e^z) &= 1 - (1-p)\frac{1}{y_2+1} + p\frac{1}{y_1+1} \\ &= (1-p)\frac{y_2}{y_2+1} + p\frac{y_1}{y_1-1} \end{aligned}$$

6.8.1 Example: Derivation of Option pricing PDE

We recall from Theorem 6.7.2 that for all f, $f(X_t) - f(X_0) - \int_0^t Af(X_s)ds$ is a martingale, if the process X_t is Feller and A is its infinitesimal generator. Note that this martingale has expected value at time 0 equal to 0.

Suppose we want to price an option on X_t and further assume that its value is a function of the current value X_t only. This has been shown to happen for European type option when the final payoff is of the form $\psi(X_T) = F(e^{X_T})$. We know that in this case:

$$V(t, X_t) = E[e^{-r(T-t)}\psi(x_T)|\mathcal{F}_t] = e^{-r(T-t)}E[\psi(x_T)|\mathcal{F}_t] = e^{-r(T-t)}g(X_t),$$

is the option value at any time t, where the expectation is under the equivalent martingale measure. We used the function $g(\cdot)$ to denote the expectation in the case when X_t is a Markov process.

If X_t solves a homogeneous SDE, it will be a Feller process and applying the Theorem 6.7.2 for g we obtain: $g(X_t) - g(X_0) - \int_0^t Ag(X_s)ds$ is a martingale. The initial value of this martingale is 0 and taking the derivative we get:

$$\frac{\partial g}{\partial t} - Ag(X_t) = 0.$$

Now using that $g(x) = e^{r(T-t)}V(t,x)$ and substituting the derivatives also using that since the process is homogeneous A involves only derivatives in x we finally get:

$$\frac{\partial V}{\partial t} - AV - rV = 0.$$

This creates the PDE whose solution is the option price. Please note that this is the same PDE we derived in the previous chapters with a much different derivation, and furthermore that this derivation is much more general.

As an example, say X_t solves the 1 dimensional SDE:

$$dX_t = b(X_t)dt + \sigma(X_t)dB_t$$

The infinitesimal generator is:

$$A = b(x)\frac{\partial}{\partial x} + \frac{1}{2}\sigma^2(x)\frac{\partial^2}{\partial x^2}.$$

Therefore the option price will solve:

$$\frac{\partial V}{\partial t} - b(x)\frac{\partial V}{\partial x} - \frac{1}{2}\sigma^2(x)\frac{\partial^2 V}{\partial x^2} - rV = 0.$$

which in the case when the coefficients are constant reduces to the Black-Scholes-Merton PDE.

6.9 A general diffusion approximation method

In this section, we discuss how to approximate general diffusion processes with discrete processes. This discussion forms the basics of all tree approximations. We follow the ideas presented in our original paper [70].

For our underlying continuous-time stochastic process model, we assume that the price process S_t and the volatility driving process Y_t solve the equations:

$$\begin{cases} dS_t &= rS_t dt + \sigma(Y_t) S_t dW_t \\ dY_t &= \alpha(\nu - Y_t)dt + \psi(Y_t)dZ_t \end{cases} \tag{6.32}$$

This model is written under the equivalent martingale measure and it is a generalization of all the stochastic volatility models traditionally used in Finance. For example, the Heston model is obtained by chosing $\sigma(x) = \sqrt{x}$ and $\psi(x) = \beta\sqrt{x}$. SABR is the notable exception since the diffusion term contains a S_t^β term.

For simplicity, we assume W_t and Z_t are two *independent* Brownian motions. The case when they are correlated is more complex but it may be treated by constructing an approximation to a modified pair $X_t, \bar{Y}_t$ where $\bar{Y}_t$ is suitably modified to be independent of X_t. The ideas have never been written in the general model but the transformation may be found in the tree approximation for the Heston model in [20].

For convenience, we work with the logarithm of the price $X_t = \log S_t$, equations (6.32) become:

$$\begin{cases} dX_t &= \left(r - \frac{\sigma^2(Y_t)}{2}\right) dt + \sigma(Y_t)dW_t \\ dY_t &= \alpha(\nu - Y_t)dt + \psi(Y_t)dZ_t \end{cases}. \tag{6.33}$$

Here r is the short-term risk-free rate of interest. The goal is to obtain discrete time discrete space versions of the processes (X_t, Y_t) which would converge in distribution to the continuous processes in (6.33). Using the fact that the price of the European Option can be written as a conditional expectation of a continuous function of the price, we can use the continuous mapping theorem to ensure convergence in distribution of the option price calculated using the discrete approximation to the real price of the option.

We will present such a tree approximation in Section 6.11 starting on page 159. In this section, we present a general Markov chain convergence theorem. The theorem is based on Section 11.3 in the book [197] (see also [61]).

We assume we observe historical stock prices $S_{t_1}, S_{t_2}, \dots, S_{t_K}$ (and also $X_{t_1} = \log S_{t_1}, X_{t_2} =_{t_2}, \dots, X_{t_n} = \log S_{t_K}$. We will use this history of prices to estimate the volatility process in the next Section 6.10, on page 155. This method will produce an approximating process Y_t^n that converges in distribution, for each time $t = t_i$, $i = 1, 2, \dots, K$, as $n \to \infty$, to the conditional law of the volatility process Y_t in (6.33) given $S_{t_1}, S_{t_2}, \dots, S_{t_i}$.

We run this filter for all the past times and at the final time t_K (the present time $t = 0$) we will price our option. For simplicity of notation we will drop the subscript: we use $Y^n = Y_0^n$ to denote the discrete process distribution at time $t_K = 0$, and we use $Y = Y_0$ for the distribution of the continuous process at time $t_K = 0$. The convergence result we prove in this section will be applied to provide a quadrinomial-tree approximation to the solution of the following equation:

$$dX_t = \left(r - \frac{\sigma^2(Y)}{2}\right) dt + \sigma(Y) dW_t. \tag{6.34}$$

In modeling terms, we shall construct two tree approximations. In one approximation the "Static model" uses the distribution of the random variable $Y = Y_0$ for all future steps $\{Y_s : s \leq 0\}$ in (6.33). The distribution of Y is unchanged from time 0 (the present) into the future. This static assumption is in sharp contrast to the dynamic in (6.33) however it will construct a simpler and faster tree which will be shown to converge to the same number. We shall also present a second tree approximation the"Dynamic model", which is a straightforward approximation to the SV model in (6.33). For the latter model a simple Euler simulation, rather than a tree, will be used.

Let T be the maturity date of the option we are trying to price and N the number of steps in our tree. Let us denote the time increment by $h = \Delta t = \frac{T}{N}$. We start with a discrete Markov chain $(x(ih), \mathcal{F}_{ih})$ with transition probabilities denoted $p_x^z = p(h, x, z)$ of jumping from the point x to the point z in h units of time. For a homogeneous Markov chain (as is our case) these transition probabilities only depend on h, x and z. For each h let $\mathbf{P}_x^h$ be the probability measure on $\mathbb{R}$ characterized by:

$$\begin{cases} (i) & \mathbf{P}_x^h\left(x(0) = x\right) = 1 \\ (ii) & \mathbf{P}_x^h\Big(x(t) = \frac{(i+1)h-t}{h}\, x(ih) + \frac{t-ih}{h}\, x((i+1)h) \\ & \qquad\qquad , \quad ih \leq t < (i+1)h \Big) = 1, \quad \forall\, i \geq 0 \\ (iii) & \mathbf{P}_x^h\left(x((i+1)h) = z | \mathcal{F}_{ih}\right) = p_x^z, \quad \forall\, z \in \mathbf{R} \text{ and } \forall\, i \geq 0 \end{cases} \tag{6.35}$$

Remarks 6.9.1. The obscure equations above say the following:

1. Properties (i) and (iii) say that $(x(ih), \mathcal{F}_{ih})$, $i \geq 0$ is a time-homogeneous Markov Chain starting at x with transition probability p_x^z under the probability measure $\mathbf{P}_x^h$.

2. Condition (ii) assures us that the process $x(t)$ is linear between $x(ih)$ and $x((i+1)h)$. In turn, this means that the process $x(t)$ we construct is a tree.

3. We will show in Section 6.11 precisely how to construct this Markov chain $x(ih)$

Conditional on being at x and on the Y^n variable, we construct the following quantities as functions of $h > 0$:

$$b_h(x, Y^n) = \frac{1}{h} \sum_{\text{z successor of x}} p_x^z (z - x) = \frac{1}{h} E^Y \left[\Delta x(ih)\right]$$

$$a_h(x, Y^n) = \frac{1}{h} \sum_{\text{z successor of x}} p_x^z (z - x)^2 = \frac{1}{h} E^Y \left[\Delta^2 x(ih) \right],$$

where the notation $\Delta x(ih)$ is used for the increment over the interval $[ih, (i+1)h]$, and E^Y denotes conditional expectation with respect to the sigma algebra $\mathcal{F}_{t_K}^Y$ generated by the Y variable. Here the successor z is determined using both the predecessor x and the Y^n random variable. We will see exactly how z is defined in Section 6.11 when we construct our specific Markov chain. Similarly, we define the following quantities corresponding to the infinitesimal generator of the equation (6.33):

$$b(x, Y) = r - \frac{\sigma^2(Y)}{2},$$
$$a(x, Y) = \sigma^2(Y).$$

We make the following assumptions, where $\xrightarrow{\text{D}}$ denotes convergence in distribution:

$$\lim_{h \searrow 0} b_h(x, Y^n) \xrightarrow{\text{D}} b(x, Y), \text{ when } n \to \infty \tag{6.36}$$

$$\lim_{h \searrow 0} a_h(x, Y^n) \xrightarrow{\text{D}} a(x, Y), \text{ when } n \to \infty \tag{6.37}$$

$$\lim_{h \searrow 0} \max_{z \text{ successor of } x} |z - x| = 0. \tag{6.38}$$

Theorem 6.9.1. *Assume that the martingale problem associated with the diffusion process X_t in* (6.34) *has a unique solution $\mathbf{P}_x$ starting from $x = \log S_K$ and that the functions $a(x, y)$ and $b(x, y)$ are continuous and bounded. Then conditions* (6.36), (6.37) *and* (6.38) *are sufficient to guarantee that $\mathbf{P}_x^h$ as defined in* (6.35) *converges to $\mathbf{P}_x$ as $h \searrow 0$ and $n \to \infty$. Equivalently, $x(ih)$ converges in distribution to X_t the unique solution of the equation* (6.34)

Proof. The proof of the theorem mirrors that of the Theorem 11.3.4 in [197]. In fact the theorem states the same result but in our, more specific case.

The proof consists in showing the convergence of the infinitesimal generators formed using the discretized coefficients $b_h(.,.)$ and $a_h(.,.)$ to the infinitesimal generator of the continuous version.

It should be noted also that hypothesis (6.38), though implying condition (2.6) in the above cited book, is a much stronger assumption. However, since it is available to us, we shall use it. □

It should be noted that lost throughout this mathematical notation the Theorem above states a very important fact. When approximating diffusion processes one needs to calculate the expected instantaneous increment $b_h(x, Y^n)$ and the instantaneous second moment of the increment $a_h(x, Y^n)$ and make sure that when $h \to 0$ they converge to the drift coefficient and the squared diffusion coefficient of the continuous time stochastic process.

6.10 Particle filter for the distribution of the unobservable stochastic volatility process Y_t

To construct a tree approximation we need the distribution of the Y process at time $t = 0$. One can use a theoretical distribution for Y if such distribution exists. For example, in the Heston model the Y distribution is given by the Cox-Ingersoll-Ross (CIR) [47] process and this process has a stationary non-central Chi squared (χ^2) distribution. If the volatility is known analytically we just need to generate random numbers from the distribution as detailed in the next section.

For more general processes the distribution of Y cannot be calculated analytically. In this section we present a general method to calculate an approximating distribution. We assume that the coefficients ν, α and the functions $\sigma(y)$ and $\psi(y)$ are known or have already been estimated.

The particle filtering method is based on an algorithm due to Del Moral, Jacod, and Protter [53] adapted to our specific case. The method approximates for all $i = 1, \cdots, K$,

$$p_i\,(dy) = \mathbf{P}\left[Y_{t_i} \in dy | X_{t_1}, \cdots, X_{t_i}\right].$$

That is the filtered stochastic volatility process at time i given all discrete passed observations of the stock price. If $X_{t_1}, \cdots, X_{t_i}$ are observed, then p_i's depend explicitly on these observed values.

In [53], Section 5, the authors provide the particle filtering algorithm. The algorithm produces n time varying particles $\left\{Y_i^j : i = 1, \cdots, K; j = 1, \cdots, n\right\}$ and their corresponding probabilities $\{p_i^j : i = 1, \cdots, K; j = 1, \cdots, n\}$. These form an approximate probability distribution that converges for each i to the limiting probability defined by $p_i\,(dy)$. The algorithm is a two step genetic-type algorithm with a Mutation step followed by a Selection step. We refer to the above cited article (Theorem 5.1) for the proof of convergence.

We present the algorithm in detail next. The data we work with is a sequence of returns: $\{x_0 = \log S_0, x_1 = \log S_1, \ldots, x_K = \log S_K\}$, observed from the market. We note that we are abusing notation here. Recall that the goal is the distribution at time $t = 0$ (now). However, all these observations $x_0, \ldots, x_K$ are in the past. It will become too confusing to use negative indices. Rather we use this notation and the goal is to approximate the distribution of the variable Y_K (now).

We need an initial distribution for the volatility process Y_t at time $t = 0$. In our implementation we use $\delta_{\{\nu\}}$ (all paths start from ν the long term mean of the distribution). We can use any distribution here for example a uniform distribution centered on ν. Here $\delta_{\{x\}}$ is a notation for the Dirac point mass. The only mathematical condition we need is that the functions $\sigma(x)$ and $\psi(x)$ be twice differentiable with bounded derivatives of all orders up to 2.

We need to define a weight function that will be used to measure how far the particles are from their target. The only requirement on this function (denoted using ϕ) is to have finite L^1 norm. In order to obtain good results, we need ϕ to be concentrated near 0.

In our application we use:

$$\phi(x) = \begin{cases} 1 - |x| & \text{if } -1 < x < 1 \\ 0 & \text{otherwise} \end{cases}$$

Another function that also produces good results is $\phi(x) = e^{-2|x|}$.

The algorithm generates n particles. For $n > 0$ we define the contraction corresponding to $\phi(x)$ as:

$$\phi_n(x) = \sqrt[3]{n}\,\phi(x\sqrt[3]{n}) = \begin{cases} \sqrt[3]{n}\left(1 - |x\sqrt[3]{n}|\right) & \text{if } -\frac{1}{\sqrt[3]{n}} < x < \frac{1}{\sqrt[3]{n}} \\ 0 & \text{otherwise} \end{cases}. \tag{6.39}$$

We choose $m = m_n$ an integer.

Step 1: We start with $X_0 = x_0$ and $Y_0 = y_0 = \nu$.

Mutation step: This part calculates a random variable with approximatelly the same distribution as (X_1, Y_1) using the well known Euler scheme for the equation (6.33). More precisely we set:

$$\begin{aligned} Y(m, y_0)_{i+1} \quad &: \quad = Y_{i+1} = Y_i + \frac{1}{m}\alpha(\nu - Y_i) + \frac{1}{\sqrt{m}}\psi(Y_i)U_i \\ X(m, x_0)_{i+1} \quad &: \quad = X_{i+1} = X_i + \frac{1}{m}(r - \frac{\sigma^2(Y_i)}{2}) + \frac{1}{\sqrt{m}}\sigma(Y_i)U_i'. \end{aligned} \tag{6.40}$$

Here U_i and U_i' are iid Normal random variables with mean 0 and variance 1 so that $\frac{1}{\sqrt{m}}U_i$ is the distribution of the increment of the Brownian motion. At the end of this first evolution step we obtain:

$$\begin{aligned} X_1 &= X(m, x_0)_m, \\ Y_1 &= Y(m, y_0)_m. \end{aligned} \tag{6.41}$$

Selection step: We repeat the **Mutation step** n times and obtain n pairs: $\{(X_1^j, Y_1^j)\}_{j=\overline{1,n}}$.

For each resulting Y_1^j we assign a discrete weight given by $\phi_n(X_1^j - x_1)$. Since ϕ_n has most weight at 0, the closer the resulting endpoint X_1^j is to to actual observation x_1 the larger the weight. Since we can have multiple paths ending in the same value for Y_1 we accumulate these into a probability distribution. Mathematically:

$$\Phi_1^n = \begin{cases} \frac{1}{C}\sum_{j=1}^{n} \phi_n(X_1^j - x_1)\delta_{\{Y_1^j\}} & \text{if } C > 0 \\ \delta_{\{0\}} & \text{otherwise.} \end{cases} \tag{6.42}$$

Here the constant C is chosen so that Φ_1^n is a probability distribution, ($C = \sum_{j=1}^n \phi_n(X_1^j - x_1)$). The idea is to "**select**" only the values of Y_1 which correspond to values of X_1 not far away from the realization x_1. We end the first Selection step by simulating n iid variables $\{Y'^j_1\}_{j=\overline{1,n}}$ from the distribution Φ_1^n we just obtained. These are the starting points of the next particle mutation step.

Steps 2 to K: For each subsequent step $i = 2, 3, \ldots, K$, we first apply the mutation step to each of the particles selected at the end of the previous step. Specifically, in the same equations (6.40) we start with $X_0 = x_{i-1}$ and $Y_0 = Y'^j_{i-1}$ for each $j = 1, 2, \ldots, n$. Again we obtain n mutated pairs $\{(X_i^j, Y_i^j)\}_{j=1,2,\ldots,n}$. Then we apply the selection step to these pairs. That is, we use them in the distribution (6.42) using $\{(X_1^j, Y_1^j)\}_{j=1,2,\ldots,n}$ for the pairs, and using x_i instead of x_1 in the weight function.

At the end of each step i we obtain a discrete distribution Φ_i^n, and this is our estimate for the transition probability of the process Y_t at the respective time t_i. In our construction of the quatrinomial tree, we use only the latest estimated probability distribution, i.e., Φ_K^n. We will denote this distribution using the set of particles $\{\bar{Y}_1, \bar{Y}_2, \cdots, \bar{Y}_n\}$ together with their corresponding probabilities $\{\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_n\}$.

Next we present an implementation of the filter. The implementation is in R and uses two input vectors. x is a vector containing logarithm of stock values and $DATE$ is a corresponding vector of dates. The model we are exemplifying is:

$$dX_t = \left(r - \frac{e^{-|Y_t|}}{2}\right) dt + e^{-|Y_t|} dW_t$$
$$dY_t = \alpha\,(m - Y_t)\,dt + \beta Y_t dZ_t$$

The dynamics can of course be changed in the four functions at the beginning of the code. The parameter values have to be known and in the example are fixed $r = 0.05$, $\alpha = 0.1$, $m = \log(0.125)$, $\beta = 0.1$. We assume we are working with daily data and in the code below we are using $\Delta t = \frac{1}{252}$ the conventional number of trading days in a year. The vector *logvect* contains logarithm of daily equity values.

```
#Coefficients functions:

miu.y <- function(x, alpha=.1, m=log(0.125)) alpha*(m-x)
sigma.y <- function(x, alpha=.1) alpha*x

sigma.x <- function(x) exp(x)
miu.x <- function(x,r=0.05) r-sigma.x(x)^2/2

psi.select<-function(x) #this is the weight function
{ifelse((x>-1)&&(x<1),1-abs(x),0)}
```

```
# an alternative second function:
#psi.select<-function(x)
# {exp(-abs(x))/2}

psi.select.n<-function(x,n) # this is the contraction
{n^(1/3)*psi.select(x*n^(1/3))}

#the mutation step function
euler.mutation <-function(x0,y0,m)
{w<-rnorm(m); z<-rnorm(m); y<-y0; x<-x0;delta<-1/(252*m);
for(i in 1:m) {
        y<- y + miu.y(y)*delta +sigma.y(y)*w[i]*sqrt(delta);
        x <- x + miu.x(y)*delta + sigma.x(y)*z[i]*sqrt(delta)
        };
return(c(x,y))
}

#the selection step function
euler.selection <- function(a,x1,n)
{b<-sapply(a[1,]-x1,psi.select.n,n=n);
cc<-matrix(c(round(a[2,],2),b/sum(b)),2,length(b),byrow=T)
if(all(b==0)) {d<-matrix(c(0,1),2,1)}
    else{d<-density.estimation(cc[,cc[2,]!=0])};
return(d)
}

#this function combines the two steps and creates the final
#distribution
vol.distrib<-function(logvect,a2,n,m)
{
for (i in 1:(length(logvect)-1))
{y<-sample(a2[1,],n,replace=T,prob=a2[2,]);
a1<-sapply(y,euler.mutation,x0=logvect[i],m=m);
a2<-euler.selection(a1,logvect[i+1],n)
};
return(a2)
}
```

In the last function the parameters n and m have to be given. n is the number of paths (particles) simulated at each selection step and m is the number of the intermediate steps when mutating particles. In the last function *logvect* is a vector containing logaritms of asset prices while $a2$ is a matrix with two rows. The first row contains the values for the filtered distribution at the previous step $\{\bar{Y}_1, \bar{Y}_{2,} \cdots, \bar{Y}_n\}$, while the second row contains the respective probabilities, $\{\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_n\}$.

6.11 Quadrinomial Tree approximation for Stochastic Volatility models

The goal is to price an option with maturity T written on the underlying process S_t. We note that if the option is about variability (a variance swap for example) the static model constructed here will not provide a good approximation. We refer to our subsequent work [218, 217] for an approach in this case.

The purpose of this section is to construct a discrete tree which will assist in calculating an estimate of the option's price. The data available is the value of the stock price today S, and a history of earlier stock prices. As described in the previous section, we use the historical values to compute a set Y^n of particles $\{\bar{Y}_1, \bar{Y}_2, \cdots, \bar{Y}_n\}$ with weights $\{\bar{p}_1, \bar{p}_2, \dots, \bar{p}_n\}$, whose empirical law approximates the volatility process Y_0 at time 0.

Remarks 6.11.1. The market is incomplete. Thus, the option price is not unique.

It is easy to see that the remark is true as is the case with all the stochastic volatility models since the number of sources of randomness (2) is bigger than the number of tradable assets (1). Remember that the volatility process is not a tradable asset, and cannot, in practice, be observed. This means that the price of a specific derivative will not be completely determined by just observing the dynamics of (X, Y) in equation (6.33), and by the arbitrage free assumption. However, the requirement of no arbitrage will imply that the prices of various derivatives will have to satisfy certain internal consistency relationships, in order to avoid arbitrage possibilities on the derivative market.

To take advantage of this fact and to be able to use the classical pricing idea in incomplete markets, we make an assumption:

Assumption 6.11.1. There is a liquid market for every contingent claim.

This assumption assures us that the derivatives are tradable assets. Thus, taking the price of one particular option (called the "benchmark" option) as given will allow us to find a unique price for all the other derivatives. Indeed, we would then have two sources of randomness and two tradable assets (the stock and the benchmark), and the price of any derivative would be uniquely determined.

Let us divide the interval $[0, T]$ into N subintervals each of length $\Delta t = \frac{T}{N} = h$. At each of the points $i\Delta t = ih$ the tree is branching. The nodes on the tree represent possible values for $X_t = \log S_t$.

6.11.1 Construction of the one period model

Now, assume that we are at a point x in the tree. What are the possible successors of x?

We sample a volatility value from the discrete approximating distribution Y^n at each time period ih, $i \in \{1, 2, \dots, N\}$. Denote the value drawn at step

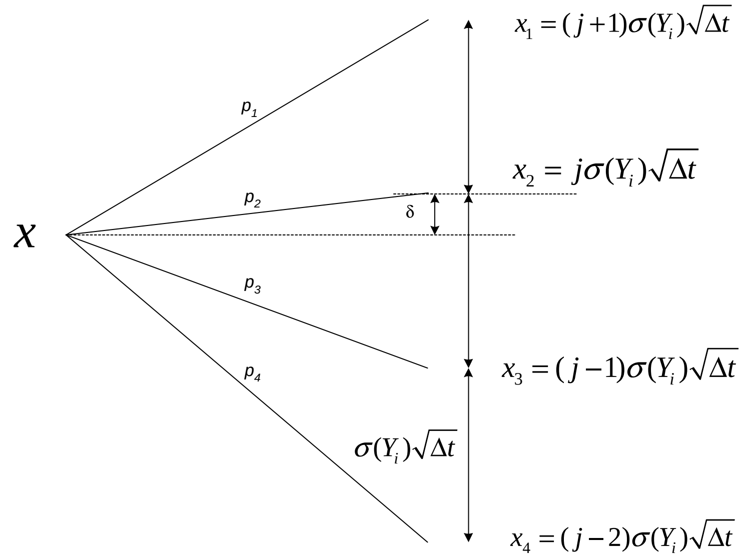

Figure 6.2: The basic successors for a given volatility value. Case 1

i corresponding to time ih by Y_i. Corresponding to this volatility value Y_i we will construct the successors in the following way.

We consider a grid of points of the form $l\sigma(Y_i)\sqrt{\Delta t}$ with l taking integer values. The parent node x will fall at one such point or between two such grid points. We let j denote the integer that corresponds to the point directly above x. Mathematically, j is the point that attains:

$$\min\left\{l \in \mathbf{N} \quad | \quad l\,\sigma(Y_i)\sqrt{\Delta t} \geq x\right\}.$$

We will have two possible cases: either the point $j\,\sigma(Y_i)\sqrt{\Delta t}$ on the grid corresponding to j is closer to x, or the point $(j-1)\sigma(Y_i)\sqrt{\Delta t}$ corresponding to $j-1$ is closer. We will treat the two cases separately. This is needed so that that math work.

Case 1. $j\,\sigma(Y_i)\sqrt{\Delta t}$ is the point on the grid closest to x.

Figure 6.2 on page 160 refers to this case.

Let us denote $\delta = x - j\,\sigma(Y_i)\sqrt{\Delta t}$.

Remarks 6.11.2. In this case we have $\delta \in \left[-\frac{\sigma(Y_i)\sqrt{\Delta t}}{2}, 0\right]$ or $\frac{\delta}{\sigma(Y_i)\sqrt{\Delta t}} \in \left[-\frac{1}{2}, 0\right]$

One of the assumptions we need to verify is (6.36), which asks the mean of the increment to converge to the drift of the X_t process in (6.34). In order to simplify this requirement, we add the drift quantity to each of the successors.

This trick will simplify the conditions (6.36) to ask now the convergence of the mean increment to zero. This idea has been previously used by many authors including Leisen as well as Nelson & Ramaswamy.

Explicitly, we take the 4 successors to be:

$$\begin{cases} x_1 &= (j+1)\sigma(Y_i)\sqrt{\Delta t} + \left(r - \frac{\sigma^2(Y_i)}{2}\right)\Delta t \\ x_2 &= j\sigma(Y_i)\sqrt{\Delta t} + \left(r - \frac{\sigma^2(Y_i)}{2}\right)\Delta t \\ x_3 &= (j-1)\sigma(Y_i)\sqrt{\Delta t} + \left(r - \frac{\sigma^2(Y_i)}{2}\right)\Delta t \\ x_4 &= (j-2)\sigma(Y_i)\sqrt{\Delta t} + \left(r - \frac{\sigma^2(Y_i)}{2}\right)\Delta t \end{cases} \tag{6.43}$$

First notice that condition (6.38) is trivially satisfied by this choice of successors. The plan is to set a system of equations verifying the variance condition (6.37), and the mean condition (6.36). We then solve the system to find the joint probabilities p_1, p_2 , p_3 and p_4. Algebraically, we write: $j\,\sigma(Y_i)\sqrt{\Delta t} = x - \delta$, and using this we infer that the increments over the period Δt are:

$$\begin{cases} x_1 - x &= \sigma(Y_i)\sqrt{\Delta t} - \delta + \left(r - \frac{\sigma^2(Y_i)}{2}\right)\Delta t \\ x_2 - x &= -\delta + \left(r - \frac{\sigma^2(Y_i)}{2}\right)\Delta t \\ x_3 - x &= -\sigma(Y_i)\sqrt{\Delta t} - \delta + \left(r - \frac{\sigma^2(Y_i)}{2}\right)\Delta t \\ x_4 - x &= -2\sigma(Y_i)\sqrt{\Delta t} - \delta + \left(r - \frac{\sigma^2(Y_i)}{2}\right)\Delta t \end{cases} \tag{6.44}$$

Conditions (6.36) and (6.37) translate here as:

$$\mathbf{E}[\Delta x|Y_i] = \left(r - \frac{\sigma^2(Y_i)}{2}\right)\Delta t$$

$$\mathbf{V}[\Delta x|Y_i] = \sigma^2(Y_i)\Delta t$$

where by Δx we denote the increment over the period Δt.

We will solve the following system of equations with respect to p_1, p_2 p_3 and p_4:

$$\begin{cases} \left(\sigma(Y_i)\sqrt{\Delta t} - \delta\right)p_1 & +(-\delta)p_2 + \left(-\sigma(Y_i)\sqrt{\Delta t} - \delta\right)p_3 + \left(-2\sigma(Y_i)\sqrt{\Delta t} - \delta\right)p_4 = 0 \\ \left(\sigma(Y_i)\sqrt{\Delta t} - \delta\right)^2 p_1 & +(-\delta)^2 p_2 + \left(-\sigma(Y_i)\sqrt{\Delta t} - \delta\right)^2 p_3 + \left(-2\sigma(Y_i)\sqrt{\Delta t} - \delta\right)^2 p_4 \\ & -\mathbf{E}[\Delta x|Y_i]^2 = \sigma^2(Y_i)\Delta t \\ & p_1 + p_2 + p_3 + p_4 = 1 \end{cases} \tag{6.45}$$

Eliminating the terms in the first equation of the system we get:

$$\sigma(Y_i)\sqrt{\Delta t}\ (p_1 - p_3 - 2p_4) - \delta = 0$$

or

$$p_1 - p_3 - 2p_4 = \frac{\delta}{\sigma(Y_i)\sqrt{\Delta t}}. \tag{6.46}$$

Neglecting the terms of the form $\left(r - \frac{\sigma^2(Y_i)}{2}\right)\Delta t$ when using (6.46) in the second equation in (6.45) we obtain the following:

$$\begin{aligned}\sigma^2(Y_i)\Delta t = \sigma^2(Y_i)\Delta t\,(p_1 + p_3 + 4p_4) + 2\delta\sigma(Y_i)\sqrt{\Delta t}\,(p_3 - p_1 + 2p_4) + \delta^2 \\ - \left(\sigma(Y_i)\sqrt{\Delta t}\,(p_1 - p_3 - 2p_4) - \delta\right)^2.\end{aligned}$$

After simplifications, we obtain the equation:

$$(p_1 + p_3 + 4p_4) - (p_1 - p_3 - 2p_4)^2 = 1.$$

So now the system of equations to be solved looks like:

$$\begin{cases} p_1 + p_3 + 4p_4 = 1 + \frac{\delta^2}{\sigma^2(Y_i)\Delta t} \\ p_1 - p_3 - 2p_4 = \frac{\delta}{\sigma(Y_i)\sqrt{\Delta t}} \\ p_1 + p_2 + p_3 + p_4 = 1 \end{cases} \tag{6.47}$$

Note that this is a system with 4 unknowns and 3 equations. Thus, there exists an infinite number of solutions to the above system. Since we are interested in the solutions in the interval $[0, 1]$, we are able to reduce somewhat the range of the solutions. Let us denote by p the probability of the branch furthest away from x. In this case $p := p_4$. Also, let us denote $q := \delta / \left(\sigma(Y_i)\sqrt{\Delta t}\right)$ and, using Remark 6.11.2, we see that $q \in [-\frac{1}{2}, 0]$. Expressing the other probabilities in term of p and q, we obtain:

$$\begin{cases} p_1 = \frac{1}{2}\left(1 + q + q^2\right) - p \\ p_2 = 3p - q^2 \\ p_3 = \frac{1}{2}\left(1 - q + q^2\right) - 3p \end{cases} \tag{6.48}$$

Now using the condition that every probability needs to be between 0 and 1, we solve the following three inequalities:

$$\begin{cases} \frac{1}{2}\left(-1 + q + q^2\right) \le p \le \frac{1}{2}\left(1 + q + q^2\right) \\ \frac{q^2}{3} \le p \le \frac{1+q^2}{3} \\ \frac{1}{6}\left(-1 - q + q^2\right) \le p \le \frac{1}{6}\left(1 - q + q^2\right) \end{cases} \tag{6.49}$$

It is not difficult to see that the solution of the inequalities (6.49) is $p \in [\frac{1}{12}, \frac{1}{6}]$. Thus we have the following result.

Lemma 6.11.1. *If we are in the conditions of* **Case 1** *with the successors given by* (6.43)*, then the relations* (6.48) *together with $p_4 = p$ give an Equivalent Martingale Measure for every $p \in [\frac{1}{12}, \frac{1}{6}]$.*

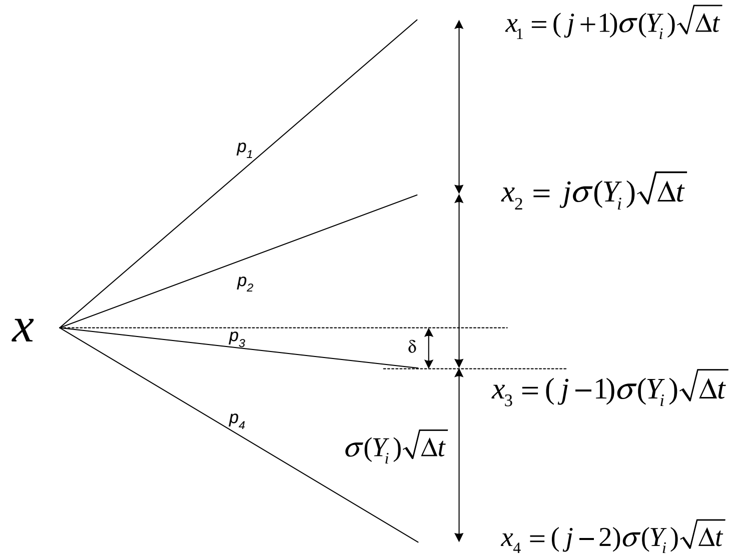

Figure 6.3: The basic successors for a given volatility value. Case 2.

It is clear from the above that we obtain an equivalent martingale measure for every $p \in [\frac{1}{12}, \frac{1}{6}]$ thanks to the first equation in (6.45).

Case 2. $(j-1)\sigma(Y_i)\sqrt{\Delta t}$ is the point on the grid closest to x.

Figure 6.3 on page 163 refers to this case.
Let us denote $\delta := x - (j-1)\,\sigma(Y_i)\sqrt{\Delta t}$.

Remarks 6.11.3. In this case we have $\delta \in \left[0, \frac{\sigma(Y_i)\sqrt{\Delta t}}{2}\right]$ or $\frac{\delta}{\sigma(Y_i)\sqrt{\Delta t}} \in \left[0, \frac{1}{2}\right]$

The 4 successors are the same as in Case 1; the increments are going to be:

$$\begin{cases} x_1 - x &= 2\sigma(Y_i)\sqrt{\Delta t} - \delta + \left(r - \frac{\sigma^2(Y_i)}{2}\right)\Delta t \\ x_2 - x &= \sigma(Y_i)\sqrt{\Delta t} - \delta + \left(r - \frac{\sigma^2(Y_i)}{2}\right)\Delta t \\ x_3 - x &= -\delta + \left(r - \frac{\sigma^2(Y_i)}{2}\right)\Delta t \\ x_4 - x &= -\sigma(Y_i)\sqrt{\Delta t} - \delta + \left(r - \frac{\sigma^2(Y_i)}{2}\right)\Delta t \end{cases} \tag{6.50}$$

Remarks 6.11.4. This second case is just the mirror image of the first case with respect to x.

Using the previous Remark, we can see that the same conditions (6.36) and

(6.37) will give the following system:

$$\begin{cases} 4p_1 + p_2 + p_4 = 1 + \frac{\delta^2}{\sigma^2(Y_i)\Delta t} \\ 2p_1 + p_2 - p_4 = \frac{\delta}{\sigma(Y_i)\sqrt{\Delta t}} \\ p_1 + p_2 + p_3 + p_4 = 1 \end{cases} \tag{6.51}$$

Notice that this is simply the system (6.47) with the roles of p_1 and p_4, and the roles of p_2 and p_3, reversed.

Again, denoting by p the probability of the successor furthest away, in this case p_1, and by $q := \delta / \left(\sigma(Y_i)\sqrt{\Delta t}\right)$, and using this time Remark 6.11.3 (i.e. $q \in [0, \frac{1}{2}]$) we obtain,

$$\begin{cases} p_2 = \frac{1}{2}\left(1 + q + q^2\right) - 3p \\ p_3 = 3p - q^2 \\ p_4 = \frac{1}{2}\left(1 - q + q^2\right) - p \end{cases} \tag{6.52}$$

This is just the solution given in (6.48) with $p_1 \rightleftarrows p_4$ and $p_2 \rightleftarrows p_3$ taking into account the interval for δ. Thus, we will have the following result exactly like in case 1.

Lemma 6.11.2. *If we are in the conditions of* **Case 2** *with the successors given by* (6.43) *then the relations* (6.52) *together with* $p_1 = p$ *give an Equivalent Martingale Measure for every* $p \in [\frac{1}{12}, \frac{1}{6}]$.

Next we present a simple implementation in *R* of the one step quadrinomial tree.

```
#this function determines which case is applicable case 1 or 2.
odd<-function(x,y)
{return(ifelse(trunc(x/y)/2-trunc(trunc(x/y)/2)==0,trunc(x/y)+1,trunc(x/y)))}


#construction of the one step successors
successors.quatri<-function(x,y,r,delta,p)
{ss<-sigma.x(y)*sqrt(delta);s<-sigma.x(y);
if(trunc(x/ss)==round(x/ss))
         {        j<-trunc(x/ss)+1;
                  x1<-(j+1)*ss+(r-s^2/2)*delta;
                  x2<-j*ss+(r-s^2/2)*delta;
                  x3<-(j-1)*ss+(r-s^2/2)*delta;
                  x4<-(j-2)*ss+(r-s^2/2)*delta;
                  dd<-(x-(j-1)*ss)/ss;
                  p1<-p;
                  p2<-(1+dd+dd^2)/2-3*p;
                  p3<-3*p-dd^2;
                  p4<-(1-dd+dd^2)/2-p;;
```

```
        }
else
        {       j<-round(x/ss);
                x1<-(j+1)*ss+(r-s^2/2)*delta;
                x2<-j*ss+(r-s^2/2)*delta;
                x3<-(j-1)*ss+(r-s^2/2)*delta;
                x4<-(j-2)*ss+(r-s^2/2)*delta;
                dd<-(x-j*ss)/ss;
                p1<-(1+dd+dd^2)/2-p;
                p2<-3*p-dd^2;
                p3<-(1-dd+dd^2)/2-3*p;
                p4<-p;
        };
return(c(x1,x2,x3,x4,p1,p2,p3,p4))
}
```

The p value in the successors function is to be chosen in the interval $p \in [\frac{1}{12}, \frac{1}{6}]$. In practical examples we found that there isn't one particular value for p that guarantees a good valuation. We found that a lower range closer to $1/12$ is good for put options while a higher value close to $1/6$ is good for call options. The better range of values change depending on whether the option is out/in/at the money. In our practical experiments we picked a particular value fixed. We calibrated p using this chosen option value and then the p thus found served to generate the entire option chain.

We will continue these functions in the next section that talks about the multiple steps construction.

6.11.2 Construction of the multi-period model. Option valuation.

Suppose now that we have to compute an option value. For illustrative purposes we will use an European type option, but the method should work with any kind of path dependent option on S_t (e.g., American, Asian, Barrier etc).

Assume that the payoff function is: $\Phi(X_T)$. The maturity date of the option is T, and the purpose is to compute the value of this option at time $t = 0$ (for simplicity) using our model (6.33). We divide the interval $[0, T]$ into N smaller ones of length $\Delta t := \frac{T}{N}$. At each of the points $i\Delta t$ with $i \in \{1, 2, \ldots, N\}$ we then construct the successors in our tree as in the previous section. This tree converges in distribution to the solution of the stochastic model (6.34). A proof of this may be found in [70].

In order to calculate an estimate for the option price we will use the approximate discrete distribution for the initial volatility Y calculated in section 6.10 on page 155. Assume we know the initial Y distribution, i.e. we know the stochastic volatility particle filter values $\{\bar{Y}_1, \bar{Y}_2, \ldots, \bar{Y}_n\}$, each with probability $\{\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_n\}$. To construct a tree with N steps we sample N values from this distribution, and use them like the realization of volatility process Y

along the future N steps of the tree. Call these sampled values $Y_1, \cdots, Y_N$, we start with the initial value x_0. We then compute the 4 successors of x_0 as in the previous section using for the volatility the first sampled value, Y_1. After this, for each one of the 4 successors we compute their respective successors using the second sampled volatility value Y_2, and so on.

The tree we construct this way allows us to compute one instance of the option price using the standard no-arbitrage pricing technique. That is, after creating the quadrinomial tree based on the sampled values, we compute the value of the payoff function Φ at the terminal nodes of the tree. Then, working backward in the path tree, we compute the value of the option at time $t = 0$ as the discounted expectation of the final node values. Because the tree is recombining by construction, the level of computation implied is manageable, typically of a polynomial order in N.

However, unlike the regular binomial tree a single tree is not enough for the stochastic volatility models. We iterate this procedure by using repeated samples $\{Y_1, \cdots, Y_N\}$, and we take the average of all prices obtained for each tree generated using each separate sample. This volatility sampling Monte Carlo method converges, as the number of particles n increases, to the true option price for the quadrinomial tree in which the original distribution of the volatility is the true law of Y_0 given past observations of the stock price. The proof of this uses the following fact proved by Pierre del Moral, known as a *propagation of chaos* result: as n increases, for a fixed number N of particles $\{Y_1, \cdots, Y_N\}$ sampled from the distribution of particles $\{\bar{Y}_1, \bar{Y}_2, \ldots, \bar{Y}_n\}$ with probabilities $\{\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_n\}$, the Y_i's are asymptotically independent and all identically distributed according to the law of Y_0 given all past stock price observations. Chapter 8, and in particular Theorem 8.3.3 in [52], can be consulted for this fact. The convergence proof in the next section is also based on del Moral's propagation of chaos.

The next R code presents an implementation of the one instance calculation using the quadrinomial tree. The code below contains two special functions: concatenation and deconcatenation. Due to the way the tree is constructed even though each node has 4 successors many of these successors overlap. Specifically, suppose that we have k nodes at a particular level (step) in the tree. Initially, at the next step there are $4k$ generated successors. However, many of these are the same since the tree is recombining. The concatenate function creates the list of distinct successor nodes to be used for the next step in the tree construction. However, this maping needs to be used later when we go backward through the tree and calculate the option value. This is accomplished by the deconcatenate function.

```
payoff<-function(x,k) ifelse(x-k>0,x-k,0)

concatenation<-function(a)
{y<-a[1];
for (i in 2:length(a))
{
```

```
if (all(y!=a[i])) y<-c(y,a[i])};
return(sort(y))
}

deconcatenation<-function(a,d,b)
{dd<-a;
for(i in 1:length(d)) {dd[a==d[i]]<-b[i]};
return(dd)
}
```

The tree.eur.quatri.optim function constructs one instance of the multi-period quadrinomial tree and evaluates the option value.

```
tree.eur.quatri.optim<-function(x,y.val,y.prob,n,r,Maturity,Strike,p)
{delta<-Maturity/n;
y<-sample(y.val,n,replace=T,prob=y.prob);prob<-rep(0,length(y));
for(i in 1:n){prob[i]<-y.prob[y.val==y[i]]};
a<-list("");b<-list("");d<-list("");
a[[1]]<-matrix(successors.quatri(x,y[1],r,delta,p),8,1);
d[[1]]=a[[1]][1:4,];
for(i in 2:n)
        {
        a.succ<-c(a[[i-1]][1,],a[[i-1]][2,],a[[i-1]][3,],a[[i-1]][4,]);
        d[[i]]<-concatenation(a.succ);
    a[[i]]<-sapply(d[[i]],successors.quatri,y=y[i],r=r,delta=delta,p=p)
        }
b[[n]]<-matrix(payoff(c(exp(a[[n]][1,]),exp(a[[n]][2,]),
                exp(a[[n]][3,]), exp(a[[n]][4,])),Strike),4,
                length(a[[n]][1,]),byrow=T);
for(i in n:2)
    {
    counter<-b[[i]][1,]*a[[i]][5,]+b[[i]][2,]*a[[i]][6,]
                +b[[i]][3,]*a[[i]][7,]+b[[i]][4,]*a[[i]][8,];
    b[[i-1]]<-deconcatenation(matrix(c(a[[i-1]][1,],a[[i-1]][2,],
                a[[i-1]][3,], a[[i-1]][4,]),4,length(a[[i-1]][1,]),
                byrow=T),d[[i]],counter);
    };
return(b[[1]][1,]*a[[1]][5,]+b[[1]][2,]*a[[1]][6,]+
                b[[1]][3,]*a[[1]][7,]+b[[1]][4,]*a[[1]][8,])
}
```

In this function x is the current log stock value, y.val and y.prob are vectors of filtered volatility values and probabilities. In the function the a list contains the list of successors and probabilities at every step. The d list contains the concatenated node values. Finally, the list b contains the option value associated with every node in a. We use lists since we not need to specify dimensions for elements in a list. Even though it looks like there will be a lot of elements in these lists in practice that isn't the case and the code is very fast. To illustrate this, Table 6.1 shows an example of the progression of the number of elements in these lists for a few steps.

These numbers were obtained for a specific example pricing IBM options. Please note that the dimension of d the concatenated map is random and varies every time the tree is ran.

Table 6.1: An illustration of the typical progression of number of elements in lists a, d, and b. For comparison we also included the number of elements (node values and probabilities) needed to be stored for a binomial tree and for a non-recombining (exploding tree)

Step	a dim	d **dim**	b dim	binomial	non-recombining
1	8	4	4	4	8
2	32	7	16	6	32
3	56	9	28	8	128
4	72	13	36	10	512
5	104	17	52	12	2,048
6	136	19	68	14	8,192
7	152	21	76	16	32,768
8	168	27	84	18	131,072

We refer to [70] for a full discussion of the convergence order of the tree and for empirical results obtained applying the tree methodology to pricing options.

We also note that the code above only calculates the option value using one tree.

6.12 Problems

1. For a Markov process define its infinitesimal operator:

$$T(t)f(X_s) = \int f(y)p(t, X_s, y)dy = E[f(X_{t+s})|X_s],$$

   Show that this operator is a contraction with respect to:

   (a) Norm 0: $\|f(x)\|_0 = \max_x f(x)$

   (b) Norm 1: $\|f(x)\|_1 = \left|\int f(x)dx\right|$

   (c) Norm 2: $\|f(x)\|_2 = \left(\int f(x)^2 dx\right)^{\frac{1}{2}}$

   (d) Norm p: $\|f(x)\|_p = \left(\int f(x)^p dx\right)^{\frac{1}{p}}$, for all $p \geq 1$

2. (The Martingale problem on L). Let (f, g) be a pair of functions in L. Find a process $\{X_t\}_t$ defined on E such that:

$$M_t = f(X_t) - f(X_0) - \int_0^t g(X_0)ds$$

   is a martingale.

3. Construct a binomial tree ($u = \frac{1}{d}$) with three quarters for an asset with present value \$100. If $r = 0.1$ and $\sigma = 0.4$, using the tree compute the prices of:

(a) An European Call option with strike 110.

(b) An European Put option with strike 110.

(c) An American Put option with strike 110.

(d) An European Call option with strike 110, assuming that the underlying asset pays out a dividend equivalent to 1/10 of its value at the end of the second quarter.

(e) An American Call option with strike 110 for an underlying as in iv).

(f) An *asian option*, with strike equal to the average of the underlying asset values during the period.

(g) A *barrier option* (for different barriers).

4. In the binomial model obtain the values of u, d and p given the volatility σ and the risk free interest rate r, for the following cases:

   (a) $p = \frac{1}{2}$.

   (b) $u = \frac{1}{d}$. **Hint**: $E\left(S_{t+\Delta t}^2\right) = S_t^2 e^{(2r+\sigma^2)\Delta t}$

5. Compute the value of an option with strike \$100 expiring in four months on underlying asset with present value by \$97, using the binomial model. The risk free interest rate is 7% per year and the volatility is 20%. Assume that $u = \frac{1}{d}$.

6. In a binomial tree with n steps, let $f_j = f_{u...ud...d}$ (j times u and $n-j$ times d). For an European Call option expiring at $T = n\Delta t$ with strike K, show that

$$f_j = \max\{Su^j d^{n-j}, 0\}$$

   In particular, if $u = \frac{1}{d}$, we have that

$$f_j \geq 0 \Longleftrightarrow j \geq \frac{1}{2}(n + \frac{log(K/S)}{log(u)})$$

   (a) Is it true that the probability of positive payoff is the probability of $S_T \geq K$? Justify your answer.

   (b) Find a general formula for the present value of the option.

7. We know that the present value of a share is \$40, and that after one month it will be \$42 or \$38. The risk free interest rate is 8% per year continuously compounded.

   (a) What is the value of an European Call option that expires in one month, with strike price \$39?

   (b) What is the value of a Put option with the same strike price?

8. Explain the difference between pricing an European option by using a binomial tree with one period and assuming no arbitrage, and by using risk neutral valuation.

9. The price of a share is \$100. During the following six months the price can go up or down in a 10% per month. If the risk free interest rate is 8% per year, continuously compounded.

   (a) what is the value of an European Call option expiring in one year with strike price \$100?

   (b) Compare with the result obtained when the risk free interest rate is monthly compounded.

10. The price of a share is \$40, and it is incremented in 6% or it goes down in 5% every three months. If the risk free interest rate is 8% per year, continuously compounded, compute:

    (a) The price an European Put option expiring in six months with a strike price of \$42.

    (b) The price of an American Put option expiring in six months with a strike price of \$42.

11. The price of a share is \$25, and after two months it will be \$23 or \$27. The risk free interest rate is 10% per year, continuously compounded. If S_T is the price of the share after two months, what is the value of an option with the same expiration date (i.e. two months) and payoff S_T^2?

12. Calculate the price for a forward contract, that is: there exists the obligation to buy at the expiration. Hint: construct a portfolio so that taking it back on time, it is risk free.

13. The price of a share is \$ 40. If $\mu = 0.1$ and $\sigma^2 = 0.16$ per year, find a 95 % confidence interval for the price of the share after six months (i.e. an interval $I_{0.95} = (\underline{S}, \overline{S})$ so that $p(S \in I_{0.95}) = 0.95$). Hint: use that if Z is a stochastic variable with standard normal distribution, then $p(-1.96 \leq Z \leq 1.96) \simeq 0.95$).

14. Suppose that the price of a share verifies that $\mu = 16\%$ and the volatility is 30%. If the closing price of the share at a given day is \$ 50, compute:

    (a) The closing expected value of the share for the next day.

    (b) The standard deviation of the closing price of the share for the next day.

    (c) A 95 % confidence interval for the closing price of the share for the next day.

15. American Down and Out Call.
    Given $K = 100, T = \frac{1}{30}, S = 100, \sigma = 0.6, r = 0.01$ and $B = 95$. Construct a tree for all nodes below the barrier value = 0.

16. Consider the parabolic equation,

$$\begin{cases} \frac{\partial u}{\partial t} + \alpha\left(x,t\right)\frac{\partial^2 u}{\partial x^2} + \beta\left(x,t\right)\frac{\partial u}{\partial x} = 0; \alpha > 0 \\ u\left(x,T\right) = \phi\left(x\right) \end{cases}$$

Using a convenient change of variable transform equation (16) into the 1-dimensional heat equation.