Patrick Hénaff

Topics in Empirical Finance

with R and Rmetrics

v 2.0

An Open Access Publication

Contents

Preface to the second edition

Preface to the first edition

This textbook is about empirical finance, and focusses on the pricing and risk management of financial assets: bonds, futures contracts, and other derivative securities.

The emphasis of this text is empirical. We present models, and verify their relevance by testing them of real data. We emphasize:

- an incremental approach to model building, starting from simple models, and building upon that foundation to construct more complex models, as needed,
- a data-driven approach: we will implement all the models that are presented, using the R statistical package and the Rmetrics libraries,
- the systematic use of simulation as a way of validating modeling decisions.

Last but not least, a particular attention is given to model estimation, in order to measure the tradeoff between model complexity and the challenges of a robust calibration.

This course would not be possible without the R statistical program and without the Rmetrics packages. We extend our deep appreciation to the R community and to Diethelm Wuertz and the Rmetrics team.

This book is open access (free as in free beer). It's also open source: feel free to clone and submit additions. You can download a PDF copy

Discreet Models

1 Arbitrage-Free Pricing and Risk Neutral Valuation

```
library(fBasics)
library(xtable)
library(empfin)
```

`{r tufte::newthought("We")}` are exclusively concerned about the pricing of redundant securities, i.e. relative value pricing. Financial economics tries to explain the pricing of underlying securities, mathematical finance is about the relative value pricing of derivatives securities.

1.1 Arbitrage-free Pricing

Consider three produce baskets of apples and oranges.

Basket	Apples	Oranges	Price (€)
B1	2	3	4
B2	3	2	5
B3	2	2	?

Is € 3.5 a fair price for basket B3? The answer is no: I can buy 5 baskets B3 and sell 2 baskets B1 and 2 baskets B1 for a riskless profit of €0.5. The fair price for B3 is 3.6 €. We have determined the arbitrage-free price for B3 by constructing a replication out of baskets B1 and B2. This is the essence of derivatives pricing.

1.1.1 Arrow-Debreu Securities

Imagine an economy that can evolve between time $t = 0$ and $t = 1$ to take one out of 3 possible states. There is a consensus to attribute probability p_i to the occurence of state i. The interest rate is null.

We next define 3 securities; each one pays a certain pattern of cash-flow according to the future state, as pictured in figure

$$fig : 3states$$

Figure 1.1: Cashflows from 3 securities

.

Assume that prices at time $t = 0$ are: $S_1 = 1.3$, $S_2 = 1.25$, $S_3 = 1.3$. The price of these securities is determined by several factors:

1. the preference of market participants for earning a payoff in one state versus another: in a risk-adverse economy, earning 1 euro when the aggregate wealth is low is more valuable than earning the same amount when the aggregate wealth is high
2. the preference for holding money today rather than at time T
3. the likelihood of each state

We are now interested in the price of securities that pays 1 euro if state i is realized, and nothing otherwise. Such securities are called Arrow-Debreu securities, and can be represented by a vector giving the value of such security in each future state of the world.

The value of the first Arrow-Debreu security is determined by solving the following system:

$$\begin{pmatrix} 0 & 1 & 1 \\ 3 & 2 & 0 \\ 1 & 1 & 2 \end{pmatrix} W_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

which yields

```
P <- matrix(c(0, 3, 1, 1, 2, 1, 1, 0, 2), 3, 3)
A1 <- matrix(c(1, 0, 0), 3, 1)
W1 <- inv(P) %*% A1
W1
```

```
##      [,1]
## [1,] -0.8
## [2,]  1.2
## [3,] -0.2
```

The value of the first Arrow-Debreu security is thus:

```
S <- matrix(c(1.3, 1.25, 1.3), 3, 1)
a1 <- t(W1) %*% S
print(a1[1, 1])
```

```
## [1] 0.2
```

Similarly, the prices a_2 and a_3 of the other two Arrow-Debreu securities are computed by:

```
W2 <- inv(P) %*% matrix(c(0, 1, 0), 3, 1)
a2 <- t(W2) %*% S
W3 <- inv(P) %*% matrix(c(0, 0, 1), 3, 1)
a3 <- t(W3) %*% S
```

The replicating portfolios and prices of the 3 A-D assets are summarized below.

% latex table generated in R 4.2.2 by xtable 1.8-4 package % Wed Jan 11 21:53:20 2023

	W1	W2	W3	Price
1	-0.80	0.20	0.40	0.20
2	1.20	0.20	-0.60	0.25
3	-0.20	-0.20	0.60	0.55

A portfolio made of the three Arrow-Debreu securities guarantees a payoff of 1 euro. Therefore, this collection can be priced at time $t = 0$ by discounting the payoff at the risk-free rate (0 for now). Thus, we must have:

$$\sum a_i = 1$$

Which is indeed the case:

```
print(a1 + a2 + a3)
```

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

1.1.2 The Price of Traded Securities

The price of Arrow-Debreu securities is determined by the price of traded securities. We now consider how these price are determined.

The current price of an asset depends on the future payoffs, and on the states in which these payoffs occur: if the payoffs of an asset are positively correlated with the aggregate market value, it will be worth less, everything else being equal, than an asset whose payoffs are negatively correlated with the aggregate market value. The capital asset pricing model formalizes this observation. Let

S_t. Asset price at time t

M_t. Aggregate market wealth at time t

R_s. Return on asset: S_T/S_0

R_m. Market return: M_T/M_0

R_f. Risk-free return: $1 + r$

The model relates the expected return of a security to the beta value:

$$E(R_s) = R_f + \beta[E(R_m) - R_f]$$

where

$$\beta = \frac{\text{Cov}(R_s, R_m)}{\text{Var}(R_m)}$$

In that framework, it can be shown that S_0, the current asset price, is given by:

$$S_0 = \frac{E(S_T) - \lambda \text{Cov}(S_T, M_T)}{R_f}$$

where λ is the market price of risk times the current market value M_0:

$$\lambda = \frac{M_0[E(R_m) - R_f]}{\text{Var}(M_T)}$$

In a complete market, the asset and the market portfolio can be expressed in terms of Arrow-Debreu securities:

$$S_0 = \sum_i V_i a_i$$

$$M_0 = \sum_i U_i a_i$$

Assume that the states are ordered in order of increasing aggregate wealth. We have:

$$\begin{aligned} E(S_T) &= \sum_i V_i p_i \\ E(M_T) &= \sum_i U_i p_i \\ E(S_T M_T) &= \sum_i U_i V_i p_i \end{aligned}$$

Substitute in (1.1.2) to get:

$$S_0 = \sum_i V_i d_i p_i$$

where the discount factor d_i is given by:

$$d_i = \frac{1 - \lambda(U_i - E(M_T))}{R_f}$$

Equation (1.1.2) shows that as aggregate wealth U_i increases, the discount factor decreases. An asset is more valuable, everything else being equal, if its payoffs occur in the states where U_i is low, and therefore where the discount factor is high.

To generalize: any factor that affects supply and demand for traded securities, and the market price of risk, will have a direct influence on the Arrow-Debreu prices and therefore on the risk neutral probabilities.

As noted by Derman and Taleb((**???**)),

> The Nobel committee upon granting the Bank of Sweden Prize in honour of Alfred Nobel, provided the following citation: "Black, Merton and Scholes made a vital contribution by showing that it is in fact not necessary to use any risk premium when valuing an option. This does not mean that the risk premium disappears; instead it is already included in the stock price." It is for having removed the effect of μ
>
> $$thestockdrift$$
>
> on the value of the option, and not for rendering the option a deterministic and riskless security, that their work is cited.

1.1.3 *Complete Market*

A complete market is a market where all Arrow-Debreu securities can be traded, and therefore any payoff profile can be replicated as a portfolio of Arrow-Debreu securities. The existence of this replicating portfolio imposes a unique arbitrage-free price for any payoff. Going back to the elementary example above, consider now a new security that has the payoff profile illustrated in figure

$$fig : new - sec$$

.

$$grow' = right, siblingdistance = 1cm$$

child node 1 child node -0.5 child node 1;

$$grow' = right, siblingdistance = 1cm$$

child node 1 child node -0.5 child node 1;

This security is equivalent to a portfolio of three Arrow-Debreu securities, and is worth

```
V <- a1 - 0.5 * a2 + a3
```

If its market price of S_4 is less than $V = 0.62$, you can earn a riskless profit by buying a unit of S_4 and selling the portfolio P.

In general, a security with payoff X_i in state i is worth:

$$\sum_i a_i X_i$$

with

$$\sum_i a_i = 1$$

and

$$a_i > 0, \quad \forall i$$

and we can interpret the Arrow-Debreu prices as probabilities. Since preferences no longer play a role, we call them "risk-neutral" probabilities.

1.1.4 *Equivalent Probability Measures*

Two probability measures p and q are equivalent if they are consistent with respect to possible and impossible outcome:

$$p_i > 0 \Leftrightarrow q_i > 0$$

Let p be the real probability measure and q be the risk-neutral measure. It is easy to show that the two measures must be equivalent.

Consider a state i such that $p_i = 0$. then the corresponding Arrow-Debreu security cannot cost anything, or a riskless profit could be gained by selling this security. Thus, $q_i = 0$. A similar argument applies for the case $p_i > 0$.

1.1.5 *The Impact of Interest Rate*

What happens to Arrow-Debreu prices and risk-neutral probabilities when interest rate is not null? In the presence of interest rate, the value of a complete set of Arrow-Debreu securities must be

$$\sum_i a_i = e^{-rT}$$

The risk-neutral probabilities are now defined as:

$$q_i = a_i e^{rT}$$

so that we still have $\sum_i q_i = 1$. As before, an arbitrary security with payoff X_i in state i is worth,

$$P(X) = e^{-rT} \sum_i X_i q_i$$

or,

$$P(X) = e^{-rT} E^Q[X]$$

Risk-neutral probabilities are compounded Arrow-Debreu prices. Here, for expository purpose, we have derived risk-neutral probabilities from state prices, but in practice, we will do the opposite: to obtain state prices from risk-neutral probabilities.

1.1.6 Trading Strategy and Dynamic Completeness

Let's now consider an economy where decisions can be made at various stages. This economy is illustrated in Figure

$$fig : bin - tree - 0$$

. At the second time step, we have 3 distinct states, i.e. 3 Arrow-Debreu securities. If we only had one time step, we would need 3 linearly independent assets to construct these securities. Now, because of the intermediate trading opportunity, we may be able to construct the Arrow-Debreu securities with fewer (i.e. two) underlying assets. A market in which every Arrow-Debreu security can be constructed with a self-financing trading strategy is called dynamically complete.

1.1.7 Discounted Asset Prices as Martingales

Let's consider again a two-stage economy. What can we say at time 0 about the expected value of a derivative at a future time t?

$$E_0^Q[P_t(X)], \forall t \geq 0$$

$$E_0^Q[P_t(X)] = e^{-2r} E_0^Q[X]$$

Now let's consider the expected price at $t = 1$:

$$\begin{aligned}
E_0^Q[P_1(X)] &= E_0^Q[e^{-r} E_1^Q[X]] \\
E_0^Q[P_1(X)] &= E_0^Q[e^{-r} E_0^Q[X]] \\
E_0^Q[P_1(X)] &= e^{-r} E_0^Q[X]
\end{aligned}$$

Similarly,

$$\begin{aligned}
E_0^Q[P_2(X)] &= E_0^Q[E_2^Q[X]] \\
E_0^Q[P_2(X)] &= E_0^Q[X]
\end{aligned}$$

The price of each state-dependent payoff grows at the risk-free rate, simply because this growth rate is incorporated in the definition of each risk-neutral probability with which we weight the state-dependent payoffs.

The expected price of any asset (as seen from time $t = 0$) grows at the risk-free rate. Not the price, but the expectation of the price. Generalizing, we have:

$$P_0(X) = e^{-rt} E_0[P_t(X)], \; \forall t > 0$$

2 Risk-Neutral Pricing in a Binomial Framework

```
## Loading required package: timeDate

## Loading required package: timeSeries

## Loading required package: fBasics
```

In this chapter we use the binomial tree framework to introduce the key concepts of option valuation.

2.1 Introduction

Consider first a one-period model: An asset S_t is worth \$40 at $t = 0$, and suppose that a month later, at $t = T$, it will be either \$45 or \$35. We are interested in buying a call option struck at $K = 40$, expiring at T. Interest rate is 1% per month. What is the fair price of this option?

The option payoff is

$$c_T = (S_T - K)^+ = \begin{cases} 5 & \text{if } S_T = 45 \\ 0 & \text{if } S_T = 35 \end{cases}$$

Now consider a portfolio made of one unit of stock and 2 call options:

$$\Phi = S - 2c$$

$$S_T - 2c_T = \begin{cases} 35 & \text{if } S_T = 45 \\ 35 & \text{if } S_T = 35 \end{cases}$$

This portfolio is worth today the same as a bank deposit that would provide \$35 in a month, or

$$\frac{35}{1 + 1\%} = 34.65$$

In this simple one-period economy, the option is thus worth:

$$c_o = \frac{40 - 34.65}{2} = 2.695$$

2.1.1 The Binomial Model for Stocks

Consider again the one-period binomial model of the previous section, and introduce notation to characterize the value of the three assets of interest, today and in the two future states, labeled "Up" and "Down". The notation is summarized in Table

$$tab : binomial$$

.

State	Stock	Bond	Call
Today	S_0	B_0	c_0
Up	$S^u_T = S_0 u$	$(1 + rT)B_0$	$c^u_T = (S^u_T - K)^+$
Down	$S^d_T = S_0 d$	$(1 + rT)B_0$	$c^d_T = (S^d_T - K)^+$

Construct a risk-free portfolio made of the option and the stock:

$$\Pi_0 = c_0 - \Delta S_0$$

To be riskless, one must have:

$$\Pi_T = (1 + rT)\Pi_0$$

In particular, this is true in the two scenarios for S_T:

$$\begin{aligned} c^u_T - \Delta S_0 u = & \quad (1 + rT)(c_0 - \Delta S_0) \\ c^d_T - \Delta S_0 d = & \quad (1 + rT)(c_0 - \Delta S_0) \end{aligned}$$

Solve for Δ:

$$\Delta = \frac{c^u_T - c^d_T}{S_0(u - d)}$$

Set $(1 + rT) = \rho$. The option value at time $t = 0$ is:

$$\begin{aligned} c_0 = & \quad \frac{1}{\rho}\Pi_T + \Delta S_0 \\ = & \quad \frac{1}{\rho}\left(\frac{\rho - d}{u - d}c^u_T + \frac{u - \rho}{u - d}c^d_T\right) \end{aligned}$$

Assume $d < \rho < u$. and define:

$$\begin{aligned} q_u = & \quad \frac{\rho - d}{u - d} \\ q_d = & \quad \frac{u - \rho}{u - d} \end{aligned}$$

Rewrite:

$$c_0 = \frac{1}{\rho}\left(q_u c^u_T + q_d c^d_T\right)$$

One can observe that $0 < q_u, q_d < 1$ and that $q_u + q_d = 1$, and therefore interpret q_u and q_d as probabilities associated with the events $S_T = S_T^u$ and $S_T = S_T^d$. Let Q be this probability measure. This leads us to write:

$$c_0 = \frac{1}{\rho} E^Q(c_T)$$

The option price is the discounted expected future payoff, under the probability Q.

2.2 *Pricing With Arrow-Debreu Securities*

An alternative derivation of the same result can be obtained with Arrow-Debreu securities. Let's first compute the price a_1 and a_2 of the two Arrow-Debreu securities in this economy. The price of the option will then be, by definition:

$$c_0 = a_1 c_T^u + a_2 c_T^d$$

where a_1 and a_2 are the prices of the Arrow-Debreu securities for the up and down scenarios.

Let's now determine the prices of these Arrrow-Debreu securities. To do so, we construct a portfolio made of x units of stock and y units of bond, that has the same payoff as an Arrrow-Debreu security. Setting $B_0 = 1$, the replicating portfolio for the first Arrow-Debreu security is obtained by solving the system:

$$\begin{pmatrix} S_0 u & \rho \\ S_0 d & \rho \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Which yields:

$$x = \frac{1}{S_0(u-d)}, \quad y = -\frac{d}{\rho(u-d)}$$

The price of the first Arrow-Debreu security is thus:

$$\begin{aligned} a_1 = & \quad xS_0 + yB_0 \\ = & \quad \frac{1}{\rho} \frac{\rho - d}{u - d} \end{aligned}$$

Similarly, the second Arrow-Debreu security is found to be worth:

$$a_2 = \frac{1}{\rho} \frac{\rho - u}{u - d}$$

and we obtain therefore the same option price as in (

$$eq : cox - ross - 1$$

)

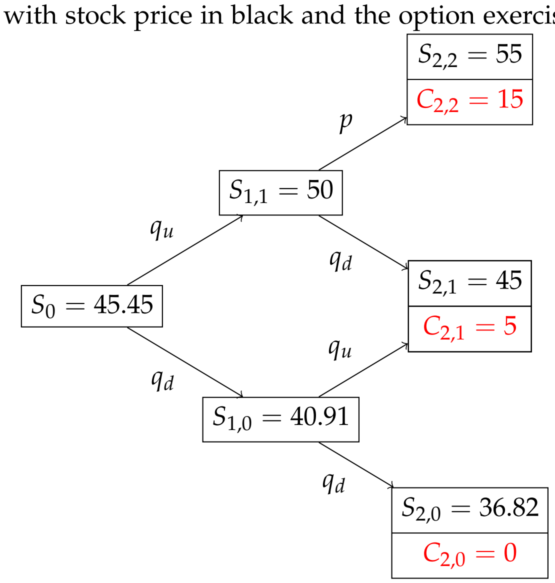
2.3 *Multi-Period Binomial Model*

The logic of the previous sections can be extented to multiple periods. When dealing with multiple periods, it is important in practice to construct recombining trees, i.e. trees where an up move followed by a down move results in the same state than a down move followed by a up move. N steps in a recombining tree generate $N+1$ states, but 2^N states in a non-recombining binomial tree.

The calculation process in a multi-period tree is best explained through an example. Consider a stock priced at 45.45€. At each period, the stock may appreciate or depreciate by 10%. The riskless rate is 5%, and we want to price a call option struck at 40€. The two-period tree is represented in figure

$$fig : 2 - per - bin$$

, with stock price in black and the option exercise value in red.



The risk-neutral probabilities are identical at all nodes:

$$
\begin{aligned}
q_u &= \frac{\rho - d}{u - d} = \frac{1.05 - .9}{1.1 - .9} = .75 \\
q_d &= .25
\end{aligned}
$$

Using these values, the option value one step prior to expiry can be computed using (

$$eq : cox - ross - 1$$

):

$$
\begin{aligned}
C_{1,1} &= \frac{1}{\rho}(q_u C_{2,2} + q_d C_{2,1}) = 11.90 \\
C_{1,0} &= \frac{1}{\rho}(q_u C_{2,1} + q_d C_{2,0}) = 3.57
\end{aligned}
$$

The reasoning that led to (

$$eq : cox - ross - 1$$

) applies however to any node, and in particular to node (0,0). The option price at that node is thus:

$$C_{0,0} = \frac{1}{\rho}(q_u C_{1,1} + q_d C_{1,0}) = 9.35$$

The process is pictured in Figure

$$fig : bin - tree - 3$$

.

2.4 *American Exercise*

American exercise refer to the right to exercise the option at any time before expiry. Clearly, an option with American exercise is worth more than the comparable option with European exercise. To price an American option, we introduce the notion of continuation value. The continuation value V_t^i at node i and time t is the option value, assuming that it will be exercised after time t. At each step, one determines the optimal decision by comparing the value of an immediate exercise to the continuation value. The option value is therefore, for a call:

$$\begin{aligned} C_t^i = & \quad \max(S_t^i - K, V_t^i) \\ = & \quad \max(S_t^i - K, \frac{1}{\rho}(q_u C_{t+1}^{i+1} + q_d C_{t+1}^i)) \end{aligned}$$

Starting from the data of the previous section, we now assume that the stock pays a dividend of 3 € in period 2. The modified tree is represented in figure

$$fig : bin - tree - 4$$

.

We price an American call struck at 40€ in this tree. Exercise values in period 2 are computed as before, giving the following values:

$$\begin{aligned} C_{2,2} = & \quad 12 \\ C_{2,1} = & \quad 0 \\ C_{2,0} = & \quad 0 \end{aligned}$$

At period 1, the option holder determines the optimal policy: exercise immediately or hold the option until period 2. The resulting values are:

$$
\begin{aligned}
C_{1,1} = & \quad \max((50-40), \frac{1}{\rho}(q_u 10 + q_d 2)) = 10 \\
C_{1,0} = & \quad \max((40.91-40), \frac{1}{\rho}(q_u 2 + q_d 0)) = 1.42
\end{aligned}
$$

The option value today is finally determined to be:

$$
\begin{aligned}
C_{0,0} = & \quad \max(5.45, \frac{1}{\rho}(q_u 10 + q_d 1.42)) \\
= & \quad 7.48
\end{aligned}
$$

Under the same conditions, a European option is worth $C_{0,0} = 6.79$. The difference comes from the early exercise of the American option in node (1,1).

2.5 *Calibration of the Binomial Model*

With interest rate assumed to be known, option prices are determined by the terms u and d that describe the binomial branching process. How should u and d be chosen?

The time interval $[0, T]$ is divided into N equal steps $[t_j, t_{j+1}]$ of length Δt. Assume that the process for the underlying asset S_t is such that $V_t = \ln(S_t/S_{t-\Delta t})$ are iid random variables with mean $\mu\Delta t$ and variance $\sigma^2\Delta t$.

$$
S_{t_j} = S_{t_{j-1}} e^{V_j}
$$

Let's determine u and d by matching the mean and variance of V_t and of the binomial process:

$$
\begin{aligned}
E(V_j) = & \quad p \ln u + (1-p) \ln d \\
= & \quad \mu\Delta t \\
V(V_j) = & \quad E(V_j^2) - E(V_j)^2 \\
= & \quad \sigma^2\Delta t
\end{aligned}
$$

Which forms a system of 2 equations and three unknown. Without loss of generality, set $p = 1/2$, to obtain:

$$
\begin{aligned}
\frac{1}{2}(\ln u + \ln d) = & \quad \mu\Delta t \\
\frac{1}{4}(\ln u + \ln d)^2 = & \quad \sigma^2\Delta t
\end{aligned}
$$

The solution is:

$$
\begin{aligned}
u = & \quad e^{\mu\Delta t + \sigma\sqrt{\Delta t}} \\
d = & \quad e^{\mu\Delta t - \sigma\sqrt{\Delta t}}
\end{aligned}
$$

The corresponding value of the risk-neutral up probability, q_u is

$$q_u = \frac{e^{r\Delta t} - e^{\mu\Delta t - \sigma\sqrt{\Delta t}}}{e^{\mu\Delta t + \sigma\sqrt{\Delta t}} - e^{\mu\Delta t - \sigma\sqrt{\Delta t}}}$$

This is the single-period risk-neutral probability, not the objective probability p. We next compute the limit of q_u as $\Delta t \to 0$.

We use the following first order approximation:

$$\begin{aligned} e^{\mu\Delta t + \sigma\sqrt{\Delta t}} &= (1 + \mu\Delta t + \sigma\sqrt{\Delta t} + \frac{1}{2}(\mu\Delta t + \sigma\sqrt{\Delta t})^2 + \dots \\ &= 1 + \mu\Delta t + \sigma\sqrt{\Delta t} + \frac{1}{2}\sigma^2\Delta t + O(\Delta t^{3/2}) \end{aligned}$$

and similarly,

$$e^{\mu\Delta t - \sigma\sqrt{\Delta t}} = 1 + \mu\Delta t - \sigma\sqrt{\Delta t} + \frac{1}{2}\sigma^2\Delta t + O(\Delta t^{3/2})$$

Combining these approximations with (

$$eq : qu$$

) yields,

$$\begin{aligned} q_u = &\quad \frac{\sigma\sqrt{\Delta t} + (r - \frac{1}{2}\sigma^2 - \mu)\Delta t + O(\Delta t^{3/2})}{2\sigma\sqrt{\Delta t} + O(\Delta t^{3/2})} \\ = &\quad \frac{1}{2} + \frac{r - \frac{1}{2}\sigma^2 - \mu}{2\sigma}\sqrt{\Delta t} + O(\Delta t) \end{aligned}$$

Let's now use these expressions for q_u to compute the mean and variance of V_j.

$$E(V_j) = q_u \ln u + (1 - q_u) \ln d$$

Use (

$$eq : ud$$

) to get:

$$E(V_j) = q_u(2\sigma\sqrt{\Delta t}) + \mu\Delta t - \sigma\sqrt{\Delta t}$$

Substitute q_u by its value (

$$eq : qu$$

) to get:

$$E(V_j) = (r - \frac{1}{2}\sigma^2)\Delta t + O(\Delta t^{3/2})$$

Similarly,

$$Var(V_j) = \sigma^2 \Delta t + O(\Delta t^{3/2})$$

The remarkable point of this result is that μ no longer appears.

Extending the reasoning of Section

$$subsec : binomial$$

to multiple periods, we write that under the risk neutral probability q_u, q_d, the option value is the discounted expected value of the payoff:

$$\begin{aligned} P(S_0) = & \quad e^{-rT} E^Q(f(S_T)) \\ = & \quad e^{-rT} E^Q(f(S_0 e^{\sum_{i=1}^N V_i})) \end{aligned}$$

where $f(S_T)$ is the payoff at expiry. We next need to compute the limit:

$$\lim_{N \to \infty} \sum_{i=1}^{N} V_i$$

The variables V_i are a function of N, thus the Central Limit Theorem cannot be used as such. However, we can invoke Lindeberg's condition to obtain the same result:

(Lindeberg's Condition) Let $X_k, k \in N$ be independent variables with $E(X_k) = \mu_k$ and $V(X_k) = \sigma_k^2$. Let $s_n^2 = \sum_i^n \sigma_i^2$. If the variables satisfy Lindeberg's condition, then

$$Z_n = \frac{\sum_{k=1}^n (X_k - \mu_k)}{s_n} \to N(0,1)$$

To simplify notation, let $E^Q(V_i) = a$, $Var^Q(V_i) = b$, Lindeberg's condition yields:

$$\frac{\sum_i^N V_i - Na}{b\sqrt{N}} \to N(0,1)$$

$$\frac{\sum_i^N V_i - Na}{b\sqrt{N}} = \frac{\sum_i^N V_i - (r - \frac{1}{2}\sigma^2)T + O(N^{-1/2})}{\sigma\sqrt{T} + O(N^{-1/2})}$$

$$\frac{\sum_i^N V_i - (r - \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T})} \to N(0,1)$$

Thus,

$$\sum_i^N V_i \to N\left((r - \frac{1}{2}\sigma^2)T, \sigma^2 T\right)$$

Finally, as $N \to \infty$, (

$$eq : pso$$

) becomes:

$$P(S_0) = \frac{e^{-rT}}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(S_0 e^{(r-\frac{1}{2}\sigma^2)T+\sigma\sqrt{T}} e^{-\frac{1}{2}u^2} du$$

which is the Black-Scholes valuation formula, as derived by Cox, Ross, and Rubinstein (1979). Again, the significance of this result is that μ does not appear in the formula.

2.5.1 *Tree Geometry*

We now use the result from the previous section to determine the geometry of the tree and the risk-neutral transition probabilities, consistent with the parameters of the diffusion process.

Recall from the previous section that u and d are defined by:

$$\begin{aligned} u = & \quad e^{\mu\Delta t+\sigma\sqrt{\Delta t}} \\ d = & \quad e^{\mu\Delta t-\sigma\sqrt{\Delta t}} \end{aligned}$$

Ultimately, μ does not appear in the valuation formula, it can thus be set to an arbitrary value without loss of generality.

In the original CRR model, $\mu = 0$, which leads to:

$$\begin{aligned} u = & \quad e^{\sigma\sqrt{t}} \\ d = & \quad e^{-\sigma\sqrt{t}} \\ q_u = & \quad \frac{e^{rt} - e^{-\sigma\sqrt{t}}}{e^{\sigma\sqrt{t}} - e^{-\sigma\sqrt{t}}} \end{aligned}$$

There are many other possibilities: a popular choice introduced by Jarrow and Rudd (1993) is to set μ so that transition probabilities are equal to $\frac{1}{2}$. Using (

$$eq : qu - 2$$

), this involves setting $\mu = r - \frac{1}{2}\sigma^2$, and the branching process is then:

$$\begin{aligned} u = & \quad e^{(r-\frac{1}{2}\sigma^2)\Delta t+\sigma\sqrt{\Delta t}} \\ d = & \quad e^{(r-\frac{1}{2}\sigma^2)\Delta t-\sigma\sqrt{\Delta t}} \\ q_u = & \quad \frac{1}{2} \end{aligned}$$

There are many other possible choices, but no significant differences in the convergence rate to the Black-Scholes option value. Most of the models, however, suffer from a form of instability which is now described.

3 Stability of the Binomial Model

{r tufte::newthought("We")} would like to verify the convergence of the Cox-Ross model to the Black-Scholes price as the number of steps N increases. This can be investigated with the following script:

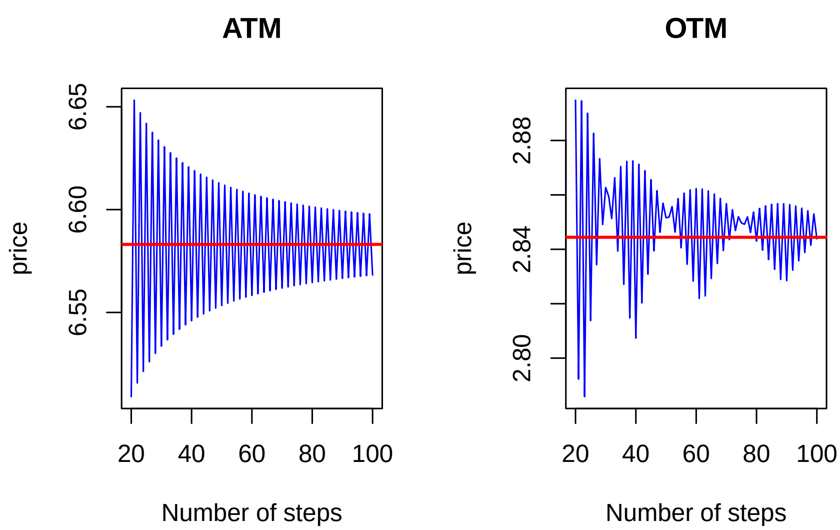
```
Strike <- 100
Spot <- 100
T1 <- 1/4
r <- 0.05
b <- 0.05
sigma <- 0.3

NN <- seq(20, 100, 1)
nb <- length(NN)
res <- matrix(nrow = nb, ncol = 2)
bs <- rep(0, 2)

# The Black-Scholes price
bs[1] <- GBSOption(TypeFlag = "c", S = Spot, X = Strike,
    Time = T1, r = r, b = b, sigma = sigma)@price
bs[2] <- GBSOption(TypeFlag = "c", S = Spot, X = Strike +
    10, Time = T1, r = r, b = b, sigma = sigma)@price

# Binomial price, function of number of
# steps
res[, 1] <- sapply(NN, function(n) CRRBinomialTreeOption(TypeFlag = "ce",
    S = Spot, X = Strike, Time = T1, r = r, b = b,
    sigma = sigma, n)@price)

res[, 2] <- sapply(NN, function(n) CRRBinomialTreeOption(TypeFlag = "ce",
    S = Spot, X = Strike + 10, Time = T1, r = r,
    b = b, sigma = sigma, n)@price)
```

A plot of the prices as a function of the number of steps (Figure

$$fig : bin - conv$$

) shows an oscillating pattern:

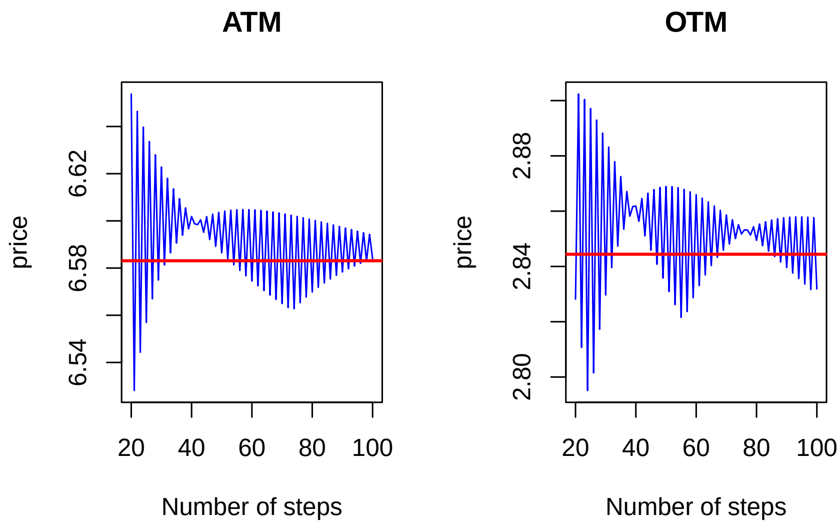
```
par(mfrow = c(1, 2))
plot(NN, res[, 1], type = "l", main = "ATM", xlab = "Number of steps",
    ylab = "price", col = "blue", ylim = c(min(res[,
        1]), max(res[, 1])))
abline(h = bs[1], lwd = 2, col = "red")

plot(NN, res[, 2], type = "l", main = "OTM", xlab = "Number of steps",
    ylab = "price", col = "blue", ylim = c(min(res[,
        2]), max(res[, 2])))
abline(h = bs[2], lwd = 2, col = "red")
par(mfrow = c(1, 1))
```



Other binomial algorithms, such as Tian's, exhibit a similar pattern, as evidenced in Figure

$$fig : bin - conv - 2$$

. The horizontal line marks the Black-Scholes price.



This sawtooth pattern is due to the position of the strike relative to the sequence of nodes at expiry; we describe below some computational strategies for smoothing these oscillations and speeding up the convergence of binomial trees. See Joshi (2007) for an extensive survey of binomial models with improved convergence properties.

Since the oscillations are caused by the variations in the relative position of the strike with respect to nodes at expiry, a natural strategy, introduced by Leisen and Reimer (1996), is to construct the tree such that the strike coincides with a node. This is achieved by setting

$$\mu = \frac{1}{T} \log\left(\frac{K}{S_0}\right)$$

The resulting tree is centered on K in log space. The pricing method is implemented as follows:

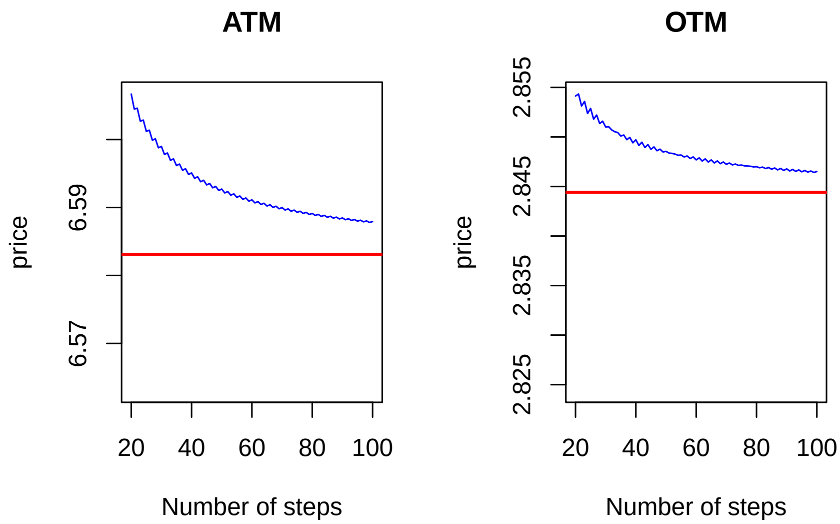
```
CRRWithDrift <- function(TypeFlag = c("ce", "pe",
    "ca", "pa"), S, X, Time, r, mu, sigma, n) {
    TypeFlag = TypeFlag[1]
    z = NA
    if (TypeFlag == "ce" || TypeFlag == "ca")
        z = +1
    if (TypeFlag == "pe" || TypeFlag == "pa")
        z = -1
    if (is.na(z))
        stop("TypeFlag misspecified: ce|ca|pe|pa")
    dt = Time/n
    u = exp(mu * dt + sigma * sqrt(dt))
    d = exp(mu * dt - sigma * sqrt(dt))

    p = (exp(r * dt) - d)/(u - d)
    Df = exp(-r * dt)
```

```
    # underlying asset at step N-1
    ST <- S * (d^(n - 1)) * cumprod(c(1, rep((u/d),
        n - 1)))
    # at step (n-1), value an European
    # option of maturity dt
    BSTypeFlag <- substr(TypeFlag, 1, 1)
    OptionValue <- GBSOption(BSTypeFlag, ST, X,
        dt, r, b, sigma)@price

    if (TypeFlag == "ce" || TypeFlag == "pe") {
        for (j in seq(from = n - 2, to = 0, by = -1)) OptionValue <- (p *
            OptionValue[2:(j + 2)] + (1 - p) *
            OptionValue[1:(j + 1)]) * Df
    }

    if (TypeFlag == "ca" || TypeFlag == "pa") {
        for (j in seq(from = n - 2, to = 0, by = -1)) {
            ContValue <- (p * OptionValue[2:(j +
                2)] + (1 - p) * OptionValue[1:(j +
                1)]) * Df
            ST <- S * (d^j) * cumprod(c(1, rep((u/d),
                j)))
            OptionValue <- sapply(1:(j + 1), function(i) max(ST[i] -
                X, ContValue[i]))
        }
    }

    OptionValue[1]
}
```

Convergence of the model as N increases is significantly improved, as evidenced by the graphs in Figure

$fig : CRRWithDrift$

.



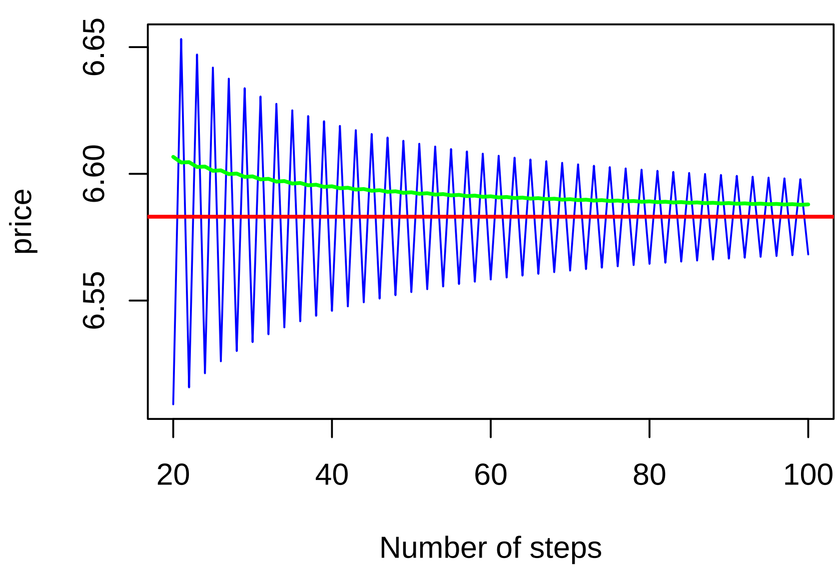
In the context of an American option, note that if the option has not been exercised at step $N-1$, the option is now European, and can be priced at these nodes with the Black-Scholes model, rather than with the backward recursion from step N (the expiry date). This simple modification smooths the option value at step $N-1$ and cancels the oscillations, as illustrated in figure

$$fig : bin - conv - crr - bs$$

, but at the price of a substantial increase in computation time.

```
CRRWithBS <- function(TypeFlag = c("ce", "pe",
    "ca", "pa"), S, X, Time, r, b, sigma, n) {
    TypeFlag = TypeFlag[1]
    z = NA
    if (TypeFlag == "ce" || TypeFlag == "ca")
        z = +1
    if (TypeFlag == "pe" || TypeFlag == "pa")
        z = -1
    if (is.na(z))
        stop("TypeFlag misspecified: ce|ca|pe|pa")
    dt = Time/n
    u = exp(sigma * sqrt(dt))
    d = 1/u
    p = (exp(b * dt) - d)/(u - d)
    Df = exp(-r * dt)

    # underlying asset at step N-1
    ST <- S * (d^(n - 1)) * cumprod(c(1, rep((u/d),
        n - 1)))
    # at step (n-1), value an European
```

```
    # option of maturity dt
    BSTypeFlag <- substr(TypeFlag, 1, 1)
    OptionValue <- GBSOption(BSTypeFlag, ST, X,
        dt, r, b, sigma)@price

    if (TypeFlag == "ce" || TypeFlag == "pe") {
        for (j in seq(from = n - 2, to = 0, by = -1)) OptionValue <- (p *
            OptionValue[2:(j + 2)] + (1 - p) *
            OptionValue[1:(j + 1)]) * Df
    }

    if (TypeFlag == "ca" || TypeFlag == "pa") {
        for (j in seq(from = n - 2, to = 0, by = -1)) {
            ContValue <- (p * OptionValue[2:(j +
                2)] + (1 - p) * OptionValue[1:(j +
                1)]) * Df
            ST <- S * (d^j) * cumprod(c(1, rep((u/d),
                j)))
            OptionValue <- sapply(1:(j + 1), function(i) max(ST[i] -
                X, ContValue[i]))
        }
    }

    OptionValue[1]
}
```



4 Trinomial Models

A natural extension of the binomial model is a trinomial model, that is, a model with three possible future states at each time step and current state.

4.1 The Trinomial Tree

The stock price at time t is S_t. From t to $t + \Delta t$, the stock may move up to S_u with probability p_u, down to S_d with probability p_d, or move to a middle state S_m with probability $1 - p_u - p_d$. The probabilities and future states must satisfy the following constraints:

1. The expected value of the stock at $t + \Delta t$ must be the forward price:

$$p_u S_u + p_d S_d + (1 - p_u - p_d) S_m = F = S e^{(r-\delta)\Delta t}$$

2. Variance:

$$p_u (S_u - F)^2 + p_d (S_d - F)^2 + (1 - p_u - p_d)(S_m - F)^2 = S^2 \sigma^2 \Delta t$$

The first method for constructing trinomial trees is simply to combine two steps of any binomial tree.

Recall that a CRR binomial tree is defined by:

$$\begin{aligned} u &= e^{\sigma\sqrt{\Delta t}} \\ d &= e^{-\sigma\sqrt{\Delta t}} \\ p &= \frac{e^{rt} - e^{-\sigma\sqrt{\Delta t}}}{e^{\sigma\sqrt{\Delta t}} - e^{-\sigma\sqrt{\Delta t}}} \end{aligned}$$

Combining two steps at a time, we obtain a trinomial tree with

$$
\begin{aligned}
S_u &= Se^{\sigma\sqrt{2\Delta t}} \\
S_m &= S \\
S_d &= Se^{-\sigma\sqrt{2\Delta t}} \\
p_u &= \left(\frac{e^{r\Delta t} - e^{-\sigma\sqrt{\Delta t}}}{e^{\sigma\sqrt{\Delta t}} - e^{-\sigma\sqrt{\Delta t}}}\right)^2 \\
p_d &= (1 - \sqrt{p_u})^2
\end{aligned}
$$

To every binomial tree corresponds a trinomial tree, obtained by aggregating two steps.

Another geometry can be defined by setting the middle node to S (~[1] p. 360):

1

$$
\begin{aligned}
S_m &= S \\
S_u &= S_m e^{\sigma\sqrt{3\Delta t}} \\
S_d &= S_m e^{-\sigma\sqrt{3\Delta t}} \\
p_u &= -\sqrt{\frac{\Delta t}{12\sigma^2}}\left(r - \frac{\sigma^2}{2}\right) + \frac{1}{6} \\
p_d &= \sqrt{\frac{\Delta t}{12\sigma^2}}\left(r - \frac{\sigma^2}{2}\right) + \frac{1}{6}
\end{aligned}
$$

5 Bibliography

Cox, J C, Stephen A Ross, and Mark Rubinstein. 1979. "Option pricing: a simplified approach." *Journal of Financial Econometrics* 7: 229–63.

Jarrow, Robert, and Andrew Rudd. 1993. *Option pricing*. Richard D. Irwin.

Joshi, Mark S. 2007. "The Convergence of Binomial Trees for Pricing the American Put." `ssrn.com/abstract=1030143`.

Leisen, D P J, and M Reimer. 1996. "Binomial Models for Option Valuation - Examining and Improving Convergence." *Applied Mathematical Finance* 3: 319–46.