Università della Svizzera italiana

Faculty of Economics

Nicola Fusari

Three Essays in Option Pricing

Submitted for the degree of Ph.D. in Economics at

Faculty of Economics
University of Lugano
Lugano, Switzerland

Thesis Committee:
Prof. G. Barone-Adesi, advisor, University of Lugano
Prof. E. De Giorgi, St. Gallen University
Prof. M. Chesney, University of Zurigo

7^{th} September 2009

swiss:finance:institute

To my family

ACKNOWLEDGEMENTS

I would like to thank my thesis supervisor Prof. Giovanni Barone-Adesi for his guidance during my doctoral studies.

I would also like to thank Prof. Fabio Trojani, Prof. Patrick Gagliardini, Prof. Francesco Franzoni, Prof. Andrea Gamba, Fulvio Corsi, Davide La Vecchia, Hakim Dall'O and all Phd. students at the Universtiy of Lugano for valuable discussions and suggestions.

Contents

Introduction

In the vast financial literature, option pricing theory has grown enormously in the last forty years. Since the seminal work of Black and Scholes (Black, F., Scholes, M. 1973. *The pricing of options and corporate liabilities.* Journal of Political Economy **81**, 637-659), principles and methods of option pricing are at the core of a wide area of financial reseach: derivatives pricing and hedging, corporate finance, real options, etc. The present work explores the option pricing world under three different perspectives: theoretical models, numerical methods and real options.

In the first chapter we develop a novel option pricing model. Specifically, we define a stochastic volatility process for the underlying evolution using the realized volatility as a proxy of the true but unobservable volatility of the underlying. That reduces enormously the estimation effort compared to standard filtering procedures and, thanks to the informational content of high frequency data, we are also able to produce pricing errors smaller than that of typical GARCH models.

The second chapter deals with the numerical approximation of Barrier option prices using binomial-trinomial trees. Lattices approaches are among the simplest and most used numerical methods in finance, but they are typically not reliable in pricing Asian-style options. We try to address this issue modifying the lattice transition probability taking into account the barrier crossing between two successive nodes. In a set of examples, we show how this procedure increases considerably the speed of convergence of the tree approximation towards the true analytical price (when available). The same procedure can be also applied to the pricing of Bermuda-style options.

Finally, the last chapter shows how option pricing principles can be used for capital budgeting decisions. Starting from the founding work of Baldwin and Clark (Baldwin, C. Y., K. B. Clark. 2000. *Design Rules: The Power of Modularity.* MIT press, Cambridge, MA) we develop a common and unified dynamic framework for the valuation of modular designs/projects. Given the complexity of the decisional process and the huge number of state variables involved, we proposed a numerical valuation approach based on the Monte Carlo Least Square (MCLS) of Longstaff and Schwartz (Longstaff, F. A., E. S. Schwartz. 2001. Valuing American options by simulation: A simple least-square approach. *Rev. Financial Stud.* **14**(1) 113-147). The approach is accurate, general and flexible.

Chapter 1

Option Pricing with Realized Volatilty

1.1 Introduction

Measurament of the istantaneous volatility of a diffusion process has been studied by Chesney et al. (1993) and Chesney and Elliott (1995). Nowdays, the availability of intraday data has facilitated the use of the so called Realized Volatility (RV henceforth) approach for measuring and forecasting volatility and has attracted much attention in the past few years. In its standard form, the RV is usually proxied by the sum of squared high-frequency returns over a discrete time interval of typically one day. The idea goes back to the seminal work of Merton (1980), showing that the integrated variance of a Brownian motion can be approximated to an arbitrary precision using the sum of a large number of intraday squared returns. This original intuition has been recently formalized and generalized in a series of papers (Andersen, Bollerslev, Diebold and Labys 2001, 2003 and Barndorff-Nielsen and Shephard 2001, 2002a 2002b, 2005 and Comte and Renault 1998) by applying the quadratic variation theory to the broad class of special (finite mean) semimartingales processes. The RV provides us with a consistent nonparametric measure of volatility which allows to exploit all the information contained in high-frequency data and thus treating volatility as an observable variable, rather than a latent one (as in the GARCH(1,1) model for example). A drawback of a latent volatility process is the use of complicated filtering procedures. In contrast, our approach allows for a direct and highly precise volatility measure.

Volatility represents a key variable for option pricing. Surprisingly, even if the literature on RV is vast, to the best of our knowledge, little work has been devoted to combine the two strands of literature. A notable exception is the work of Stentoft (2008) in which an Inverse Gaussian model of a 30–minute return RV measure is used to price options on some individual stocks. However, this work does not provide a formal change of measure for the RV process, but it only considers the case when the risk neutral and physical dynamics of RV are the same (i.e. when the volatility risk is not priced).

In this paper we seek to partially fill this gap in the literature by combining a simple approximate long-memory model for a tick-by-tick RV measure with a general and flexible discrete time option pricing framework that allows us to provide a formal change of measure for the RV process. In particular, recognizing the importance of volatility long–memory in option pricing (Bollerslev and Ole Mikkelsen, 1996) and the distinct pricing impact of different volatility components (Adrian and Rosenberg, 2007), we model the strong persistence observed in the RV dynamics[1] using the Heterogeneous Auto-Regressive (HAR) (see Corsi

[1]See Andersen, Bollerslev, Diebold and Labys (2001, 2003) and Andersen, Bollerslev, Diebold and Ebens

(2009)) for the conditional mean of the RV. Despite the fact that the HAR model does not formally belong to the class of long memory processes, it is able to produce the same memory persistence observed in financial data. In order to have the transition density in explicit form, we specify the conditional distribution of the HAR model to be from the family of non–centered gamma. The resulting model can be classified as an Auto-Regressive Gamma process, a class of discrete–time affine processes introduced by Gourieroux and Jasiak (2006). Thus we label our new process Heterogeneous Auto-Regressive Gamma (HARG).

To derive the risk neutral dynamics of the RV we adopt the discrete time option pricing framework recently proposed by Bertholon et al. (2008). This general and flexible approach is based on three main ingredients: (i) the discrete-time historical ($\mathbb{P}$) dynamics; (ii) the Stochastic Discount Factor (SDF) representing the change of probability measure between the historical and risk-neutral world; (iii) the discrete-time risk-neutral (R.N. or $\mathbb{Q}$) dynamics. The fundamental mathematical tool employed in the description of the historical and risk-neutral dynamics is the conditional Laplace transform. Then, by assuming a standard exponentially affine SDF (as in Gourieroux and Monfort 2007 and Gagliardini, Gourieroux and Renault 2005) we can apply the simple direct pricing strategy, described in Bertholon et al. (2008).[2] The explicit R.N. dynamics obtained with our approach require the calibration of only one parameter which simplifies enormously the calibration procedure.

Summarizing, our option pricing model adopts a simple but effective long-memory RV model: the HARG model. This model combined within the option pricing framework of Bertholon et al. (2008) (we assume an exponential affine SDF) gives us an explicit discrete-time risk-neutral dynamics. Our specification relies on one free parameter that could be easily calibrated.

The rest of the paper is organized as follows. Section 1.2 defines our proposed model for the log-returns and RV, under the historical probability measure. Section 1.3 describes the change of probability measure and the resulting risk-neutral dynamics of the model. Section 1.4 presents the option pricing performances and finally in Section 1.5 we conclude.

(2001).

[2]In the direct modeling strategy, the historical dynamics and the SDF are specified to be of an affine and exponentially affine form, respectively. As a by-product, the R.N. dynamics are affine, too.

1.2 The model under historical measure

1.2.1 Probability model for log–returns

A well established result in the financial econometrics literature is that the marginal distribution of daily log–returns is not Gaussian but is typically leptokurtic, a fact that motivates the use of heavy-tailed distributions. However, Clark (1973) and Ane and Geman (2000) theoretically argue that, for an underlying continuous time diffusion process, rescaling the log–returns by an appropriate measure of the market activity allows to recover the standard Gaussian distribution. The basic intuition is that the log–return process is a Brownian motion with a random time and re–scaling by the activity measure is equivalent to a time–change that restores the standard Brownian motion in fixed time. Empirically, Andersen, Bollerslev, Diebold and Labys (2000, 2001, 2003) and Andersen, Bollerslev, Diebold and Ebens (2001) indeed showed that, when daily returns are standardized by the corresponding daily RV, the resulting distribution is nearly Gaussian. Following this approach we standardize the S&P 500 log–returns by the daily RV. Here, we employ the tick-by-tick Two-Scale RV estimator proposed by Zhang et al. (2005) (with a fast scale of two ticks and the slower one of 20 ticks).[3] When standardized by the Two-Scale estimator, the marginal distribution of log–returns becomes much closer to the normal one. This feature can be clearly seen from the QQ-plots and density plots of Figure 1.1 and 1.2, and from the values of the kurtosis passing from 7.32 to 3.11.

[Figure 1.1 should be here]

[Figure 1.2 should be here]

Given the previous considerations, we assume the following dynamics for the log–returns:

$$\ln\left(\frac{S_{t+1}}{S_t}\right) := y_{t+1} = \mu_{t+1} + \sqrt{RV_{t+1}}\epsilon_{t+1}, \tag{1.1}$$

$$\epsilon_{t+1} | \, RV_{t+1} \sim N(0,1).$$

In our notation, S_{t+1}, y_{t+1}, and RV_{t+1} are respectively the price, the log–returns and the RV at time $t+1$. For the drift of the log–returns under the physical measure we propose the following specification:

$$\mu_{t+1} = r + \gamma RV_{t+1},$$

[3] See also Aït-Sahalia and Mancini (2008) for a comprehensive analysis of the advantage of using the Two-Scale estimator over the standard RV measure.

where r represents the risk–free rate minus the dividend yield and $\gamma = \tilde{\gamma} - 1/2$. The term $-RV_{t+1}/2$ is a convexity adjustment (conditionally on the RV_{t+1}) introduced in order to obtain an explicit expression for the risk–premium ($\tilde{\gamma}$). Indeed, the conditional expectation of log–returns becomes $\mathbb{E}\left[\exp y_{t+1}|RV_{t+1}\right] = \exp\left(r + \tilde{\gamma}RV_{t+1}\right)$.
We observe that our specification introduces a contemporaneous effect of RV_{t+1} on y_{t+1}. Specifically, we observe that the functional form we are proposing implies a stochastic drift, changing with the daily RV. This feature has an interesting probabilistic implication, since our model can be embedded into the class of normal variance–mean mixture as in Barndorff-Nielsen et al. (1982), with $y_{t+1}|RV_{t+1} \sim N\left(r + \left(\tilde{\gamma} - \frac{1}{2}\right)RV_{t+1}, RV_{t+1}\right)$. For the moment, we do not specify any distribution for the conditioning random variable RV_{t+1}. In Section 1.2.2, we will complete our model formulation and we will introduce a probability distribution also for the RV process.

1.2.2 Probability model for the Realized Volatility

To complete our model specification under the physical measure, we need to set up the model for the dynamics of the RV process. To this end, we propose a three steps construction. (i) To capture the well documented feature of strong persistence in volatility, we follow Corsi (2009) and we model the conditional mean of the RV (given its past values) using the conditional expected value of an HAR (Heterogeneous AutoRegressive) process. The HAR model for RV is a multi-component volatility model specified as a sum of different volatility components defined over different time horizons. The multi-component structure of the HAR model permits to closely approximate the long–memory behavior of volatility, which has considerable option pricing implications as documented by Bollerslev and Ole Mikkelsen (1996). In addition, Adrian and Rosenberg (2007) show that a multi-component volatility model substantially improves the cross-sectional pricing of volatility risk. Furthermore, we here extend the original HAR model by including also a daily leverage effect.[4] Formally,

$$\mathbb{E}\left[RV_{t+1}|\mathcal{F}_t\right] = \mu_0 + \bar{\beta}_1\widetilde{RV}_t^d + \bar{\beta}_2\widetilde{RV}_t^w + \bar{\beta}_3\widetilde{RV}_t^m + \bar{\beta}_4 L_t \tag{1.2}$$

where $\widetilde{RV}_t^d = RV_t$, $\widetilde{RV}_t^w = \sum_{i=0}^4 RV_{t-i}/5$ and $\widetilde{RV}_t^m = \sum_{i=0}^{21} RV_{t-i}/22$ and $\mu_0 > 0$. In our notation, $\mathcal{F}_t$ is the smallest σ-algebra generated by the tick–by–tick prices up to time t. Furthermore, L_t represents the leverage effect given by $L_t = I_{(y_t<0)}RV_t$, where $I_{(y_t<0)}$ takes value

[4] A more general extending the HAR with heterogeneous leverage and jumps is proposed in Corsi and Renò (2009).

1 if the log–return at date t is negative and zero otherwise.

(ii) For our option pricing purposes, we require the transition probability density in close form or, equivalently, the (conditional) characteristic function of the joint process of log–returns and RV. Therefore, we preliminary need to specify the conditional distributional properties of the RV. To achieve this goal, we first model the RV as an Auto Regressive Gamma process (ARG), having p-lags, with $p = 22$ as in Corsi (2009). This implies that: $RV_{t+1}|\mathcal{F}_t \sim \Gamma(\delta, \ddot{\beta}'(\mathbf{RV}_t, L_t), c)$ (Gourieroux and Jasiak (2006)), with $\mathbf{RV}_t = (RV_t, RV_{t-1}, ..., RV_{t-22})$ (column vector). In our notation, δ and c describe, respectively, the shape and the scale of the distribution, whereas $\ddot{\beta}'(\mathbf{RV}_t, L_t)$ defines the distribution location:

$$\ddot{\beta}'(\mathbf{RV}_t, L_t) = \ddot{\beta}_1 RV_t + \ddot{\beta}_2 RV_{t-1} + \ddot{\beta}_3 RV_{t-2} + \cdots + \ddot{\beta}_{21} RV_{t-21} + \Lambda L_t. \tag{1.3}$$

(iii) Corsi's specification can be applied to set up an ARG–process. However, the lagged variables in the conditional mean are overlapping, and this is not in line with the ARG(p) representation introduced by Gourieroux and Jasiak (2006). To set up a proper ARG representation, a straightforward manipulation yields to the following:

Lemma 1. *The process $\{RV_t\}_{t\geq 0}$ follows an HARG(3), having parameters δ, c and distribution location:*

$$\beta'(\mathbf{RV}_t, L_t) = \beta_1 RV_t + \frac{\beta_2}{5}(\sum_{i=1}^{4} RV_{t-i}) + \frac{\beta_3}{22}(\sum_{i=5}^{21} RV_{t-i}) + \beta_4 L_t. \tag{1.4}$$

with $\beta_1 = (\ddot{\beta}_1 + \frac{1}{5}\ddot{\beta}_2 + \frac{1}{22}\ddot{\beta}_3)$, $\beta_2 = (\frac{1}{5}\ddot{\beta}_2 + \frac{1}{22}\ddot{\beta}_3)$, $\beta_3 = \frac{1}{22}\ddot{\beta}_3$ and finally $\beta_4 = \Lambda$.

Proof. See Appendix A □

The intuition of the previous Lemma is simply that we collect the lagged terms of an ARG(22) process in three different non–overlapping factors defined as: RV_t, $RV_t^w := \sum_{i=1}^{4} RV_{t-i}$, $RV_t^m := \sum_{i=5}^{21} RV_{t-i}$. The reparametrization in Eq. (1.4) does not imply any loss of information compared to the original one of Corsi (2009), since it relies just on a different rearrangement of the terms.

Thanks to the HARG(3) specification for the RV in (1.5), we can write–down in closed–form its Conditional Moment Generating Function (CMGF) under $\mathbb{P}$. Indeed, the HARG(3), belonging to the ARG class of processes, is a discrete–time affine stochastic process (see Gourieroux and Jasiak (2006)), which thus features a strong analytical tractability. In particular, from computations similar to those in Gagliardini et al. (2005), the CMGF of an

HARG(3) process with conditional mean as in (1.4) is:

$$\varphi^{\mathbb{P}}_{ARG}(\eta) := \mathbb{E}(\exp(-\eta RV_{t+1})|(\mathbf{RV}_t, L_t)) = \exp\left(-\frac{c\eta}{1+c\eta}\left(\beta'(\mathbf{RV}_t, L_t) - \delta\ln(1+c\eta)\right)\right), \tag{1.5}$$

where $\eta \in \mathbb{R}$, $\beta \in \mathbb{R}^4$(row vector) and $\beta'(\mathbf{RV}_t, L_t)$ is as in (1.4). For the sake of brevity, we label

$$b(\eta) = \delta\ln(1+c\eta) \tag{1.6}$$

and

$$a(\eta) = \frac{c\eta}{1+c\eta}. \tag{1.7}$$

We remark that the need of the CMGF of RV under the physical measure is twofold. First, from the probabilistic point of view, the CMGF completely characterizes the distributional features of the RV (e.g. it uniquely defines its conditional moments and its transition density). Second, the CMGF of RV is important for option pricing purposes, as it is the necessary tool to describe the joint behavior of the log–returns process and RV, needed in the change of probability measure.

1.2.3 Joint Conditional Moment Generating Function

In order to complete our probabilistic description of the log returns and RV dynamics, in this Section we study the joint process of $K'_{t+1} := (y_{t+1}, RV_{t+1})$. This is a bi–dimensional real–valued process of log–returns and RV whose state space is $\mathbb{R}\times\mathbb{R}^+$. Thanks to our model set up we can write a closed–form expression for the CMGF of K'_{t+1}. This is an important result since the joint CMGF provides a complete characterization of the joint conditional (i.e. given $\mathcal{F}_t$) transition probability and, more importantly for our purposes, it provides an explicit one–to–one mapping between the parameters of (y_{t+1}, RV_{t+1}), under the measures $\mathbb{P}$ and $\mathbb{Q}$. The possibility of having explicit transformations between the parameters under the two probability measures represents a key theoretical advantage of our approach. Moreover, it is also very useful from the econometric point of view. Indeed, to capture the dynamics of K'_{t+1} under the martingale measure $\mathbb{Q}$, we just need to estimate the parameters under $\mathbb{P}$ and then apply some non–linear transformations. This is particularly convenient in our approach as the estimation of the parameters under $\mathbb{P}$ in Eq. (1.1) and in Eq. (1.4) can be obtained by just using observable realizations (the time–series), for both the log–returns and the RV process. The next Proposition provides the closed–form expression for CMGF of K'_{t+1}.

Proposition 1.1. *If* $RV_{t+1}|\mathcal{F}_t \sim \bar{\Gamma}(\delta, \beta'(\mathbf{RV}_t, L_t), c)$*, then the CMGF of* $K'_{t+1} := (y_{t+1}, RV_{t+1}) \in \mathbb{R} \times \mathbb{R}^+$ *in* $\alpha' = (\alpha_1, \alpha_2)$ *under the physical measure* $(\mathbb{P})$ *is, for* $v \in \mathbb{R}$*:*

$$\varphi_K^{\mathbb{P}}(v) = exp(-b(v) - a(v)\beta(\mathbf{RV}_t, L_t)),$$

with $v = \alpha_2 + \gamma\alpha_1 - \alpha_1^2/2$*. The terms* $b(v)$ *and* $a(v)$ *are given in Eq. (1.6) and (1.7), respectively.*

Proof. See Appendix. A □

1.2.4 Model estimation under $\mathbb{P}$

One of the main advantage of our methodology is related to the estimation of the parameters characterizing the HARG volatility process in Eq.(1.4). More specifically, given the use of RV as proxy for the unobservable volatility, we can directly estimate the parameters of the HARG process simply using observable historical data. This leads to stable estimates of the parameters avoiding the unstable calibration procedure which typically characterizes stochastic volatility models. In our application, in order to gain efficiency in the estimation of the parameter of the conditional distribution of RV, we estimate the parameters of the HARG process in Lemma 1, applying the Simulated Maximum Likelihood (SML), as suggested by Gourieroux and Jasiak (2006).[5] In Table 1.2 we show the estimated parameter (c, δ, β) and their standard deviation.

[Table 1.2 should be here]

Table 1.2 reports the estimation results on our sample. Specifically, we report the means and standard deviations of the estimated parameters obtained estimating the HARG(3) model every Wednesday form January 5th 2000 to December 27th 2006 (358 estimates). The sample period starts on 1st January 1990.

The impact of past lags on the present value of RV is given by the partial autocorrelation coefficient: $c\beta'$. According to our estimates we have that (considering also the leverage effect L_t) the sensitivity of RV_t on the conditional mean of RV_{t+1} is $c(\beta_1 + \beta_4/2) = 0.3782$, whereas the sensitivity of RV_t^w and RV_t^m are respectively $c\beta_2 = 0.2923$ and $c\beta_3 = 0.2189$. In accordance with Corsi's results, the RV coefficients show a decreasing impact of the past lags on the present value of the RV.

[5] Alternative methods for the estimation of all the parameters are the PML or, just for the parameters in the conditional mean, the OLS. For detail see Gourieroux and Jasiak (2006).

Even if the HAR(3) does not formally belong to the class of long–memory processes, Corsi's specification for the conditional mean of the RV is able to reproduce the degree of persistence observed in the empirical data. Furthermore it is also able to provide accurate forecast for RV (see Corsi 2009). In particular, to understand the reliability of our HARG(3) model, we compare some relevant dynamic features of the simulated and observed time–series.

[Figure 1.3 should be here]

In Figure 1.3, we first compare the trajectories of the empirical log–returns and the one simulated according to the HARG(3) model. We clearly see a reasonably good visual agreement, which indicates that our model is able to capture the volatility clustering observed in empirical log–returns.

[Figure 1.4 and 1.5 should be here]

To asses the ability of the model to replicate the memory persistence of the RV process, we compare the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) of data simulated from the HARG(3) model with those of the empirical ones. The results are shown in Figure 1.4 and Figure 1.5. The simulated data are obtained using the parameter estimated at the first date in the sample. Moreover, as in Corsi (2009), in order to give the model the time aggregation necessary to unfold its dynamics at daily level, we simulate the log–returns and the HARG(3) processes at high-frequency[6]. The remarkable similarity between simulated and empirical ACF and PACF shown in Figure 1.4 and Figure 1.5, implies that the HARG(3) is able to capture the strong persistence in the volatility process.

1.3 Risk–neutral dynamics

A proper option pricing model requires the derivation of the dynamics under the martingale measure $\mathbb{Q}$ of both the log–returns and the RV process. Following the direct approach of Bertholon et al. (2008), we combine the affine discrete time HARG(3) model with a discrete time SDF which takes into account a risk–premium on both the log–returns and on the stochastic volatility. Similarly to Gagliardini et al. (2005) and Gourieroux and Monfort (2007), we specify an exponential affine SDF for the time $(t, t+1)$. However, in line with the dynamics of the RV of Section 1.2.2, we consider a richer SDF involving not only the $t+1$ log–returns

[6]For our simulation purposes we use a frequency of 400 minutes. However, changing the simulation frequency does not affect the results.

and RV, but also the three relevant lag components of the RV (short, mid and long term volatility components). More precisely, assuming $r_t = 0$ for computational convenience (but all the results apply for $r_t \neq 0$), we state

Proposition 1.2. *Under the model assumptions (1.1) and (1.4) the following SDF:*

$$M_{t,t+1} = \exp(-\nu_0 - \nu_1 RV_{t+1} - \nu_2 RV_t^d - \nu_3 RV_t^w - \nu_4 RV_t^m - \nu_5 y_{t+1} - \nu_6 L_t) \tag{1.8}$$

where: $RV_t^d = RV_t$, $RV_t^w = \sum_{i=1}^{4} RV_{t-i}$, *and* $RV_t^m = \sum_{i=5}^{21} RV_{t-i}$ *is compatible with the no-arbitrage conditions iff the following implicit parameter-restrictions are satisfied:*

$$\nu_0 = -\delta \log\left(1 + c\left(\frac{1}{2}\gamma^2 + \nu_1 - \frac{1}{8}\right)\right) \tag{1.9}$$

$$\nu_2 = -c\beta_1 \frac{\frac{1}{2}\gamma^2 - \frac{1}{8} + \nu_1}{c\,(\gamma^2/2 - 1/8 - \nu_1) + 1} \tag{1.10}$$

$$\nu_3 = -c\beta_2 \frac{\frac{1}{2}\gamma^2 - \frac{1}{8} + \nu_1}{c\,(\gamma^2/2 - 1/8 - \nu_1) + 1} \tag{1.11}$$

$$\nu_4 = -c\beta_3 \frac{\frac{1}{2}\gamma^2 - \frac{1}{8} + \nu_1}{c\,(\gamma^2/2 - 1/8 - \nu_1) + 1} \tag{1.12}$$

$$\nu_6 = -c\beta_4 \frac{\frac{1}{2}\gamma^2 - \frac{1}{8} + \nu_1}{c\,(\gamma^2/2 - 1/8 - \nu_1) + 1} \tag{1.13}$$

$$\nu_5 = \gamma + \frac{1}{2}, \tag{1.14}$$

where ν_1 *remains a free parameter.*

Proof. See Appendix. A □

The last Proposition shows that the SDF in Eq. (1.8) complies with the no–arbitrage conditions, provided that the restrictions in Eq. (1.9)–(1.13) hold true. Furthermore, we remark that Proposition 1.2 defines all the parameters in the SDF, but ν_1. This is a crucial point of our model. In the estimation of the vector ν, ν_1 is the unique parameter to be calibrated. All the other parameters can be explicitly computed in closed–from, once ν_1 has been calibrated. Moreover, thanks to the SDF specification in Eq. (1.8), it is possible to write–down in closed–form the dynamics of RV under the martingale measure. To this end, we rely on the results in Proposition 1.1 and specify the CMGF under $\mathbb{Q}$ of the joint process K_{t+1}. We then have

Proposition 1.3. *Under the R.N. probability measure $\mathbb{Q}$, the CMGF of $K'_{t+1} := (y_{t+1}, RV_{t+1}) \in \mathbb{R} \times \mathbb{R}^+$, for $\alpha' \in \mathbb{R}^2$ and $\tilde{v} \in \mathbb{R}$ is:*

$$\varphi_K^{\mathbb{Q}}(\tilde{v}) = \exp\left(-a^*\left(\tilde{v}\right)\beta(\mathbf{RV}_t, L_t) - b^*\left(\tilde{v}\right)\right),$$

in which

$$a^*(\zeta)\beta = a(\zeta+\lambda)\beta - a(\lambda)\beta = \frac{c^*\beta\zeta}{1+c^*\zeta},$$
$$b^*(\zeta) = b(\zeta+\lambda) - b(\lambda) = \delta^* log(1+c^*\zeta),$$

with $\lambda = \nu_1 + \frac{\gamma^2}{2} - \frac{1}{8}$, $\tilde{v} = -\frac{1}{2}\alpha_1 + \alpha_2 - \frac{1}{2}\alpha_1^2$ and $\zeta \in \mathbb{R}$. This CMGF has the same functional form as in Eq. (A.2), implying that under the risk–neutral probability measure the RV is still an $HARG(3)$, having parameters

$$\beta^* = \frac{\beta}{(1+c\lambda)}, \tag{1.15}$$
$$\delta^* = \delta, \tag{1.16}$$
$$c^* = \frac{c}{1+c\lambda}. \tag{1.17}$$

Proof. See Appendix A □

The last Proposition provides in Eq. (1.15)–(1.17) the explicit formulas for the one–to–one mapping between the parameters under $\mathbb{Q}$ and $\mathbb{P}$. The availability of such a formulas is a consequence of the affine specification for the SDF and of the affine nature of the HARG(3) process.
From the previous results, we can finally conclude that:

Corollary 2. *Under $\mathbb{Q}$ the log–returns follow a discrete-time stochastic volatility model, with risk premium $\gamma^* = -1/2$. The RV is an $HARG(3)$ process, featuring a transition density given by a non–central gamma, with parameters β^*, δ^*, c^*.*

1.4 Option pricing: Performance assessment

1.4.1 Data description

In this Section we provide a short description of our option data. Our analysis deals with European options, written on $S\&P500$ index. The observations for the option prices range from January 5th 2000 to December 29th 2006 and the data are downloaded from OptionMetrics. Options with time to maturity less than 10 days or more than 360 days, implied volatility

larger than 70%, or prices less than 0.05 dollars are discarded. Moreover, we consider only out-of-the-money (OTM) put and call options for each Wednesday. This procedure yields a total number of $31,365$ observations. The number of put and call options is approximately the same, since we have about 51% put and 49% call options.
To perform our analysis we split the options in different categories, classifying them according to either time to maturity or moneyness. In particular, in the sequel we will use as moneyness measure the following quantity:

$$m = \frac{\ln(K/S_t)}{\sigma\sqrt{\tau}},$$

where σ is the average Implied Volatility (IV) over all sample and τ represents the maturity. This measure can be roughly interpreted as the number of standard deviations expressing the distance of the log strike from the log spot price under the assumption of Gaussian returns (i.e. in the ideal Black-Scholes model). In our classification, a put option is said to be Deep OTM (in the following DOTM) if $m \leq -1.5$ and OTM if $-1.5 < m \leq 0$. A call option is said to be DOTM if $m \geq 1.5$ and OTM if $0 < m \leq 1.5$. According to maturity τ, we classify options as: short maturity ($\tau < 60$ days), medium maturity (τ between $60 - 160$ days) and log maturity ($\tau > 160$ days).

[Table 1.1 should be here]

In Table 1.1, we summarize the information about prices and IV. For a given maturity, looking at the IV values, we notice a strong volatility smile for the short maturity options. Moreover, moving along different maturities and fixing m, we observe the volatility term structure. We notice that as long as $\tau \to 1$ year, the volatility smile becomes a volatility smirk. Specifically for large maturities (e.g. $\tau \in [0.85, 1]$) we observe a strong smirk.

[Figure 1.6 should be here]

To provide a graphical view of this phenomenon, in Figure 1.6 left panel and right panel, we show respectively the average volatility surface and the average volatility curve. From both pictures, we see that when the time to maturity τ increases from 10 days to 1 year, the smile vanishes, while the smirk remains strong also at one year maturity. This implies that the risk–neutral distribution of the log–returns is far away from the Gaussian, even after one year. This feature suggests that a time–varying stochastic volatility model should be applied to describe the strong persistence in the log–returns volatility dynamics.

1.4.2 Estimation of market price of risk

To estimate the market price of risk $\tilde{\gamma}$ in the log–returns equation, we suggest to apply a simple regression model. Specifically, our target is the analysis of the linear dependence between y_{t+1} and RV_{t+1}. To achieve this goal, we consider the dynamics of the returns as given by

$$y_{t+1} = r + \left(\tilde{\gamma} - \frac{1}{2}\right) RV_{t+1} + \sqrt{RV_{t+1}}\epsilon_{t+1} \tag{1.18}$$

$$\epsilon_{t+1} | \, RV_{t+1} \sim N(0,1)$$

Moreover, we notice that, from Eq. (1.1), it follows

$$y_{t+1} - r + \frac{1}{2} RV_{t+1} = \tilde{\gamma} RV_{t+1} + \sqrt{RV_{t+1}}\epsilon_{t+1}$$

that can be re–written as

$$\begin{aligned} \frac{y_{t+1} - r + \frac{1}{2} RV_{t+1}}{\sqrt{RV_{t+1}}} &= \tilde{\gamma}\sqrt{RV_{t+1}} + \epsilon_{t+1}, \\ \tilde{y}_{t+1} &= \tilde{\gamma}\sqrt{RV_{t+1}} + \epsilon_{t+1}. \end{aligned} \tag{1.19}$$

The last equation defines a regression model for the response variable $\tilde{y}_{t+1}$, where the explanatory variable is $\sqrt{RV_{t+1}}$.

Estimation and testing of the parameter $\tilde{\gamma}$ in the linear model of Eq. (1.19), can be achieved by means of a linear regression estimation procedure. Specifically, we consider the daily excess returns of the S&P 500 index w.r.t. the Fed Funds rate[7] from 02-Jul-1990 to 29-Dec-2006. The time series of log returns of S&P 500 is plotted in the top panel of Fig. 1.3. To estimate (and testing the significance of) the parameter $\tilde{\gamma}$ we run a robust OLS regression. The estimated $\tilde{\gamma}$ is 0.522 with a t-statistics of about 26.

1.4.3 Option pricing model

To compute the option prices under the HARG(3) model, we use a Monte Carlo approach in four steps. (i) Estimation of the HARG(3) model under the physical measure $\mathbb{P}$. (ii) Calibration of the parameter ν_1 to specify the dynamics of the RV and the log–returns under the martingale measure $\mathbb{Q}$. For every date t in the sample, this parameter is calibrated taking

[7] In fact, we use as a proxy for the risk–free rate (r) the FED Fund rate.

the most liquid OTM call and put options. This parameter seems stable across different dates, with an average equal to 0.0816 and a standard deviation equal to 0.0525. (iii) Mapping of the parameters of the model estimated under $\mathbb{P}$ into the parameters under $\mathbb{Q}$, using Eq. (1.9)-(1.13). (iv) Then, for every t, we simulate both the RV and log–returns $\mathbb{Q}$–dynamics so that, for each maturity τ and strike K, we can finally compute the prices for call OTM options at t as the average: $(1/N)\sum_{l=1}^{N}\max(S_\tau^{(l)}-K,0)$. In the previous formula, N represents the total number of MC simulations (that we set equal to $20,000$) and $S_\tau^{(l)}$ is the simulated asset price at time τ in the l-th sample path. An analogous formula holds for put options.

To evaluate the performance of our model, we compute the Root Mean Square Error (RMSE) on prices ($RMSE_P$) and on the IV ($RMSE_{IV}$). Those measures of performance are given by: $RMSE_P=\sqrt{\sum_j^{NO_t} err_P(K_j;\tau_j)^2}$ and by $RMSE_{IV}=\sqrt{\sum_j^{NO_t} err_{IV}(K_j;\tau_j)^2}$, where $err_P(K_j;\tau_j)$ is the difference between our model price and the market price for the option j, having strike K_j and maturity τ_j. NO_t represents the number of options on a given day t. Analogously $err_{IV}(K_j;\tau_j)^2$ is the difference between our model IV and the market IV, for the option j. To evaluate the performance, we compute both the $RMSE_P$ and $RMSE_{IV}$ for our option pricing model and for some other competitors. Thanks to this comparison, we are able to asses an absolute and a relative performance evaluation of our option pricing model.

A natural benchmark for our model is represented by the class of GARCH option pricing models. Indeed, both the GARCH and our HARG(3) rely on the assumption of discrete time–varying volatility and they are both based on historical observations. Specifically, we consider the following GARCH option pricing models (see Christoffersen et al. (2008)):

(a) the log–returns dynamic under $\mathbb{P}$ is given by

$$y_{t+1}=r+\lambda\sqrt{h_{t+1}}-(1/2)h_{t+1}+\sqrt{h_{t+1}}z_{t+1}, \tag{1.20}$$

with $z_{t+1}\sim N(0,1)$ and where h_{t+1} follows

$$h_{t+1}=\beta_0+\beta_1 h_t+\beta_2 h_t f(z_t). \tag{1.21}$$

(b) Different GARCH models are characterized by differences in the innovation function $f(\cdot)$. In particular, we consider the following specifications of f:

$$\begin{array}{lrcl} \textit{Simple:} & f(z_t) & = & z_t^2 \\ \textit{Leverage:} & f(z_t) & = & (z_t-\theta)^2 \\ \textit{News:} & f(z_t) & = & (|z_t-\theta|-\kappa(z_t-\theta))^2 \end{array}$$

Exactly as we have done for our option pricing model, to implement these GARCH models, (i) we first need to estimate the GARCH parameters from the observed log–returns. (ii) We perform a change of measure, following Christoffersen and Jacobs (2004). (iii) Finally, we apply MC simulation to price call and put options. In Table 1.3 we show the estimated GARCH parameters and their standard deviations.

1.4.4 Option pricing results

In the upper part of table 1.4, we show the overall performance of our model, relative to different GARCH option pricing models. We notice, that our model over–performs all GARCH models, both in terms of RMSE on price and on IV.

[Table 1.4 about here]

To have a deeper understanding, Table 1.5, Table 1.6 and Table 1.7 report the performance results (both in term of price and in term of IV) of our option pricing model relative to the different GARCH models disaggregated for different maturities and moneyness. A negative value highlights that the RMSE of our model is smaller (in percentage) than the RMSE of the GARCH.

Being able to properly capture the pronounced smile present at short maturity (i.e. $\tau < 60$), our model always outperforms any GARCH model at this horizon. This is a well–known drawback of GARCH models. To circumvent this problem, GARCH option pricing models often includes jump to be able to capture short term smile (e.g. see Christoffersen et al. (2008)). Moreover, our model always prices the at the money options with good precision. We notice, however, that some GARCH models (specifically the Leverage and News GARCH) perform slightly better than (or almost equivalently to) our model, for DOTM and OTM put options having middle and log maturities (see, for instance, Table 1.6). Nevertheless, we remark that for these options our model has a good performance in absolute terms, since the $RMSE_p$ and $RMSE_{IV}$ of the GARCH models is already small. Finally, we notice that all GARCH models have an hard time pricing DOTM and OTM call options. However, our model does not show such difficulties for those options and outperforms GARCH models both in terms of price and IV.

1.5 Conclusion

In this paper, we develop a discrete-time stochastic volatility option pricing model that exploits the historical information contained in the high frequency data. Using the realized volatility as a proxy for the unobservable return volatility we propose a simple (affine) but effective approximate long-memory process, the Heterogeneous Auto-Regressive Gamma (HARG) process. Our model combines the volatility at short, medium, and long horizon, allowing to account for the different pricing impacts of three volatility components. The proposed HARG(3) model, combined with an exponential affine stochastic discount factor, leads to a completely tractable risk- neutral dynamics. The explicit change of probability measure obtained with this framework allows the estimation of the risk neutral parameters directly under the statistical measure, leaving only one free parameters to be calibrated. This characteristic of our option pricing model could be very useful for those options with scarce liquidity for which only few prices are available each day and hence a full calibration procedure of a highly parametrized model would be infeasible.

The extensive empirical analysis on S&P 500 option index shows that the proposed model outperforms competing GARCH models, especially for short term options where our model is able to better capture the smile of the implied volatility curve.

1.6 Figures and Tables

Moneyness		Maturity					
		Less then 60		60 to 160		More then 160	
		Mean	Std	Mean	Std	Mean	Std
≤ -1.5	Put price $	1.0088	1.6356	1.9948	2.6430	3.5977	3.8319
	IV	0.3414	0.1082	0.3209	0.0841	0.2877	0.0546
	Observations	8,500		5,454		4,287	
$-1.5 - 0$	Put price $	9.6620	8.4892	19.0992	13.5132	31.4643	20.0376
	IV	0.1838	0.0611	0.2014	0.0563	0.2126	0.0450
	Observations	8,060		6,025		8,715	
$0 - 1.5$	Call price $	7.5251	8.5565	14.4219	14.1874	25.8457	23.6905
	IV	0.1492	0.0558	0.1581	0.0490	0.1652	0.0412
	Observations	8,108		6,191		8,982	
≥ 1.5	Call price $	0.3506	0.4987	0.4710	0.6540	0.7491	0.9723
	IV	0.3023	0.1257	0.2400	0.0782	0.2010	0.0394
	Observations	4,935		3,353		2.338	

Table 1.1: **Database description.** Mean and standard deviation of prices and implied volatility of S&P 500 index out-of-the-money options on each Wednesday form January 5th 2000 to December 27th 2006 (74,948 observations) sorted by moneyness/maturity categories. IV is the Black-Scholes implied volatility. Moneyness is defined as $m = \ln(K/F)/\sigma\sqrt{\tau}$. Maturity in measured in calendar days.

	c	δ	β_1	β_2	β_3	β_4
Mean	0.2991	0.7734	0.7949	0.9774	0.7319	0.9396
Std	0.0277	0.0666	0.1447	0.1090	0.0751	0.1107

Table 1.2: **HARG(3) estimation.** Mean and standard deviation of weekly parameters estimate form January 5th 2000 to December 27th 2006 (358 estimates). The sample period starts on 1st January 1990.

	Simple	*Leverage*	*News*
λ	-	0.0280	0.0279
		(0.0089)	(0.0090)
β_0	1.1870E-4	1.1870E-6	1.2075E-6
	(1.7356E-5)	(4.6326E-8)	(2.5680E-7)
β_1	0.9432	0.8885	0.8927
	(0.0028)	(0.0086)	(0.0120)
β_2	0.0530	0.0572	0.0512
	(0.0033)	(0.0033)	(0.0062)
θ	-	0.8961	0.8021
		(0.0612)	(0.0904)
κ	-	-	0.0962
			(0.0389)

Table 1.3: **GARCH models estimation.** Mean and standard deviation of weekly parameters estimate form January 5th 2000 to December 27th 2006 (358 estimates). The sample period starts on 1st January 1990. The risk premium parameter, λ, in the *Simple* model is set equal to zero to not induce a leverage effect in the risk neutral dynamics.

		Simple	*Leverage*	*News*
HARG(3)	$RMSE_p$	-39.86 (7.2790)	-18.25 (5.3549)	-17.85 (5.3284)
	$RMSE_{IV}$	-33.20 (7.8111)	-26.70 (7.1182)	-25.45 (6.9991)

Table 1.4: **Option pricing performance.** Option pricing performance of HARG(3) model relative to different GARCH specification as given by $(RMSE^{HARG} - RMSE^{GARCH})/RMSE^{GARCH}$ for price and implied volatility. The $RMSE_{IV}$ and $RMSE_p$ of each GARCH model is reported in brackets. All values are in percentage.

Moneyness m		Maturity		
		Less then 60	60 to 160	More then 160
≤ -1.5	Put price	-6.40	-6.39	-3.32
		(1.38)	(2.36)	(3.56)
	IV	-53.42	-47.41	-13.77
		(14.20)	(10.60)	(6.60)
$-1.5 - 0$	Put price	-25.68	-17.88	3.32
		(4.35)	(6.52)	(9.00)
	IV	-19.62	-8.73	12.65
		(3.83)	(3.67)	(3.43)
$0 - 1.5$	Call price	-38.84	-51.67	-55.29
		(5.54)	(10.20)	(16.57)
	IV	-34.74	-45.73	-51.90
		(4.95)	(5.50)	(5.58)
≥ 1.5	Call price	-60.21	-66.29	-74.69
		(2.19)	(5.10)	(10.31)
	IV	-19.09	-47.78	-56.52
		(8.55)	(8.20)	(10.22)

Table 1.5: **HARG(3) vs *Simple* GARCH.** Option pricing performance of HARG(3) model relative to the *Simple* GARCH as $(RMSE^{HARG} - RMSE^{GARCH})/RMSE^{GARCH}$ for price and implied volatility for each moneyness/maturity option category. The $RMSE_{IV}$ and $RMSE_p$ of the *Simple* GARCH model are reported in brackets.. All values are in percentage.

		Maturity		
Moneyness m		Less then 60	60 to 160	More then 160
≤ -1.5	Put price	-1.58	-7.31	-5.51
		(1.32)	(2.38)	(3.64)
	IV	-20.10	70.53	47.52
		(8.28)	(3.27)	(3.86)
$-1.5 - 0$	Put price	-24.39	-23.64	-14.80
		(4.28)	(7.00)	(10.98)
	IV	-8.48	5.99	22.37
		(3.36)	(3.16)	(3.15)
$0 - 1.5$	Call price	-27.41	-37.01	-44.38
		(4.67)	(7.82)	(13.32)
	IV	-16.69	-19.21	-26.65
		(3.87)	(3.70)	(3.64)
≥ 1.5	Call price	- 27.01	29.50	95.35
		(1.19)	(1.33)	(1.34)
	IV	-49.15	-55.45	-36.31
		(13.61)	(9.61)	(6.97)

Table 1.6: **HARG(3) vs *Leverage* GARCH.** Option pricing performance of HARG(3) model relative to the *Leverage* GARCH. Notation as in Table 1.5.

Moneyness m		Maturity		
		Less then 60	60 to 160	More then 160
≤ -1.5	Put price	0.92	-3.77	-2.56
		(1.29)	(2.30)	(3.53)
	IV	-18.05	73.46	51.47
		(8.07)	(3.21)	(3.76)
$-1.5 - 0$	Put price	-23.28	-21.26	-14.18
		(4.21)	(6.79)	(10.91)
	IV	-7.06	9.16	22.56
		(3.31)	(3.07)	(3.15)
$0 - 1.5$	Call price	-25.95	-34.98	-43.36
		(4.57)	(7.58)	(13.08)
	IV	-15.40	-17.36	-25.30
		(3.82)	(3.61)	(3.60)
≥ 1.5	Call price	-24.20	32.49	89.64
		(1.15)	(1.30)	(1.38)
	IV	-47.90	-58.18	-43.39
		(13.28)	(10.23)	(7.85)

Table 1.7: **HARG(3) vs *News* GARCH.** Option pricing performance of HARG(3) model relative to the *News* GARCH. Notation as in Table 1.5.

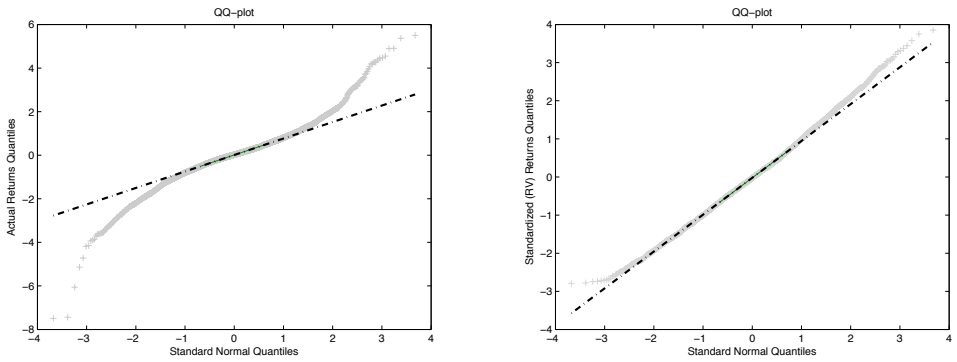

Figure 1.1: **QQ–plot.** Left panel: QQ-plot of actual log-returns of S&P 500 index from January 1st 1990 to December 31th 2007. Right panel: QQ–plot of log-returns of S&P 500 rescaled by contemporaneous realized volatility from January 1st 1990 to December 31th 2007.

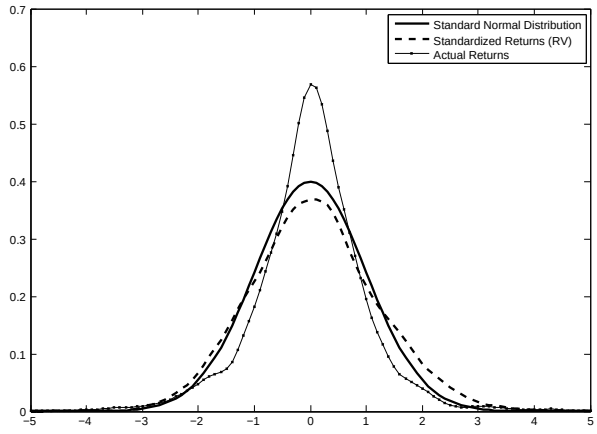

Figure 1.2: **Log-returns distribution.** Comparison of the S&P 500 index log-returns distribution under different re-scaling measures: Standard Normal distribution (solid line), log-returns rescaled by the sample standard deviation (dotted line) and log-returns divided by contemporaneous realized volatility (dashed line).

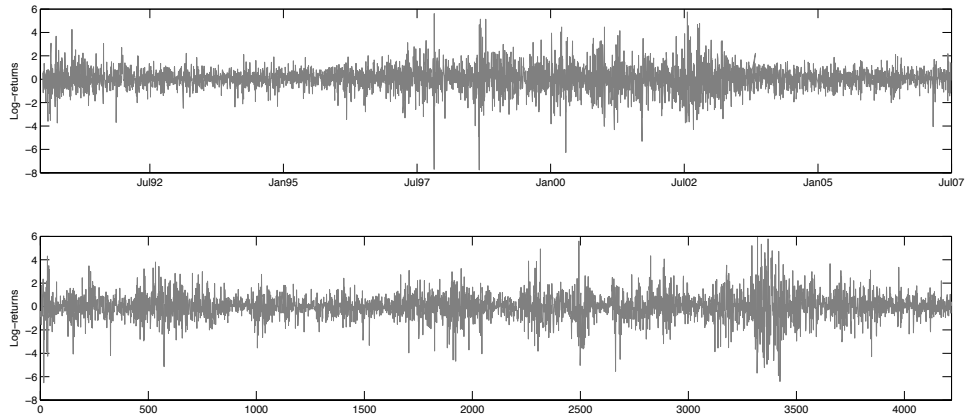

Figure 1.3: **Log-returns.** Top panel: S&P 500 log-returns from January 1st 1990 to December 31th 2007 (4218 observations). Bottom panel: log-returns simulated path from HARG(3) with $c = 0.3261, \delta = 0.8183, \beta_1 = 0.6206, \beta_2 = 0.9152, \beta_3 = 0.7949, \beta_4 = 0.9323$.

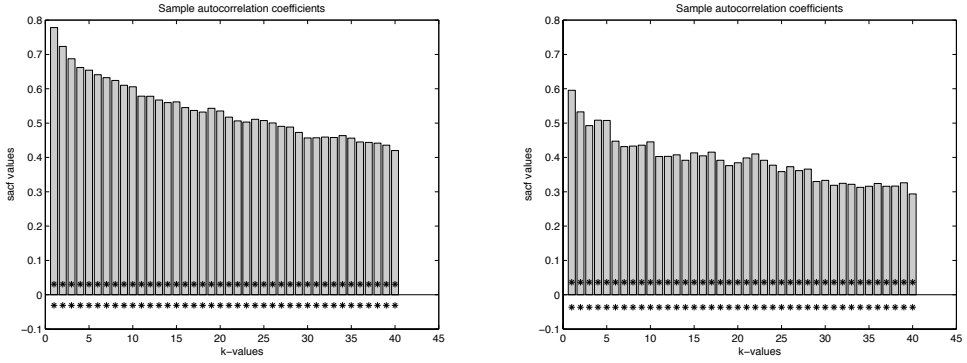

Figure 1.4: **Sample autocorrelation function.** Left panel: sample autocorrelation function of the S&P 500 index realized volatility from January 1st 1990 to December 31th 2007 (4218 observations). Right panel: sample autocorrelation function of a simulated path from HARG(3) model with $c = 0.3261, \delta = 0.8183, \beta_1 = 0.6206, \beta_2 = 0.9152, \beta_3 = 0.7949, \beta_4 = 0.9323$ (parameters estimated on January 2nd 2002).

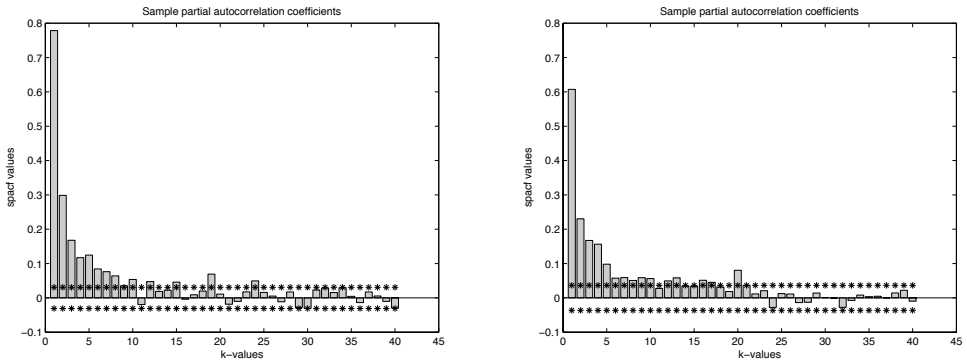

Figure 1.5: **Sample partial autocorrelation function.** Left panel: sample partial autocorrelation function of the S&P 500 index realized volatility from January 1st 1990 to December 31th 2007 (4218 observations). Right panel: sample partial autocorrelation function of a simulated path from HARG(3) model with $c = 0.3261, \delta = 0.8183, \beta_1 = 0.6206, \beta_2 = 0.9152, \beta_3 = 0.7949, \beta_4 = 0.9323$ (parameters estimated on January 2nd 2002).

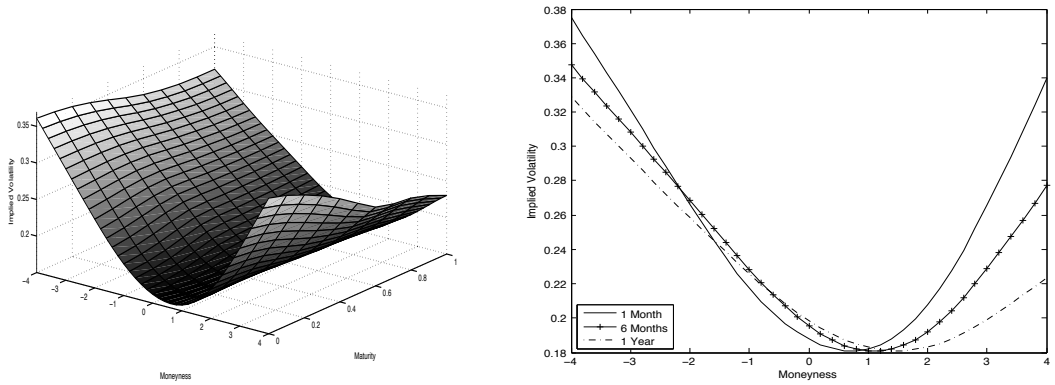

Figure 1.6: **Volatility smile-smirk in S&P 500 index options.** Left panel: implied volatility surface obtained via nonparametric smoothing of daily closing implied volatility quotes from January 5th 2000 to December 27th 2006 (74,948 observations). Independent Gaussian kernels with default bandwidth are used. Moneyness d is defined as $d = \ln K/F/\sigma\sqrt{\tau}$, where $\sigma = 24.38$ percent is the average implied volatility. Right panel: implied volatility curves obtained slicing the implied volatility surface at maturity of 1 month (solid line), 6 months (dashed.line) and 1 year (dotted line).

Chapter 2

Barrier Option Pricing

In the existing literature on barrier options much effort has been exerted to ensure convergence through placing the barrier in close proximity to, or directly onto, the nodes of the tree lattice. For a variety of barrier option types we show that such a procedure may not be a necessary prerequisite to achieving accurate option price approximations. Using the Kamrad and Ritchken (1991) trinomial tree model we show that with a suitable transition probability adjustment our probability adjusted model exhibits convergence to the barrier option price. We study the convergence properties of several option types including exponential barrier options, single linear time-varying barrier options, double linear time-varying barriers options and Bermuda options. For options whose strike price is close to the barrier we are able to obtain numerical results where other models and techniques typically fail. Furthermore, we show that it is possible to calculate accurate option price approximations with minimal effort for options with complicated barriers that defeat standard techniques. In no single case does our method require a repositioning of the pricing lattice nodes.

Keywords: barrier option, binomial tree, convergence rate, lattice models, option pricing, transition probability, trinomial tree.

JEL Classification:

2.1 Introduction

Methods for pricing barrier options consist of two approaches: numerical based methods and theoretical expressions based on continuous-time models. In this paper we use a numerical technique with a suitable modification of the transition probabilities of the Kamrad and Ritchken (1991) trinomial tree to price several types of barrier options. In doing so, we illustrate the convergence of our method to the price, when available from an analytic solution, of the respective options.

For European and American put and call options lattice models are able to yield convergence towards the option price given by a continuous-time model. In the case of a plain-vanilla option, convergence of the standard binomial or trinomial tree to the analytic value generally occurs within a few hundred time steps, yet due to distribution and non-linearity errors a persistent inaccuracy in the price of the option remains. If standard lattice methods are used to price more complex option types such as single constant barrier, multiple barrier and time-varying barrier options, they converge so slowly and exhibit such large bias that their use becomes impractical. This is an important observation because due to their lower cost and popularity in hedging financial and commodities positions, barrier and path-dependent

options are now commonplace across all financial markets. Consequently, a need for improved lattice pricing models exists particularly for the more complex option types.

To explain the upward bias that occurs when pricing options using, for example, the Cox et al. (1979b) (CRR) binomial model, Boyle and Lau (1994) studied what happens to the price of a given option when the distance of the barrier to a layer of nodes in the binomial tree varies. They found that the upward bias problem arises as a consequence of the discretization of the nodes of the binomial lattice. Specifically, the bias occurs when the option barrier passes between two successive layers of nodes comprising the binomial tree without coming close to a node. Numerically this is perceived as a mispricing of the option that takes the form of an upward bias (i.e. convergence to the continuous-time price from above) added on to the option price. In order to reduce this bias, Boyle and Lau (1994) proposed to reposition the nodes in the binomial lattice such that the barrier passes as close as possible to a given layer of nodes in the binomial tree. With this modification, increased convergence rates were achieved; in one case Boyle and Lau report that convergence improved from 800 steps in the binomial tree to just 21 steps. This variation in the distance of the barrier from a layer of nodes manifests itself as an observable pattern in a plot of the option price against the number of time steps of the binomial lattice. One observes a series of alternating crests and troughs in a plot of the option price versus the number of time divisions in the tree. Accurate approximations to the continuous-time price occur at the troughs of the convergence graph where the barrier lies in close proximity to a given layer of the binomial lattice. Conversely, poor price approximations occur at the crests.

It is known that the node-repositioning technique of Boyle and Lau is unable to produce an approximation to the option price when the initial underlying price lies close to the barrier, when there are multiple barriers or when the barrier is time-varying. Ritchken (1995) tries to address these particular problems by employing a trinomial lattice based upon the multinomial model of Kamrad and Ritchken (1991). Ritchken's method is also based upon a repositioning of the lattice nodes, however it differs from the method of Boyle and Lau in that Ritchken positions the lattice using a stretch-factor so that the barrier lies exactly upon a given level of the lattice nodes. Thus, despite increasing the number of time-divisions of the tree, there always exists a layer of nodes that coincides with the barrier level resulting in a rapid convergence to the continuous-time option price. While exhibiting good convergence, Ritchken's method suffers from three drawbacks. The first is that his method encounters difficulty converging to the continuous-time option price (and in some cases fails to converge at all) when the initial underlying price is very close to the barrier. The second drawback is that if a parameter of the option changes (maturity, volatility, etc.) then the entire lattice

must be repositioned before calculating the new option price. Finally, Ritchken's method can only handle barrier options with constant boundaries.

Another efficacious pricing model for pricing barrier options is the adaptive mesh model (AMM) of Figlewski and Gao (1999). This technique works by grafting a fine resolution lattice onto the coarse lattice of the well-known trinomial tree and using smaller time and price increments to compute the finer mesh. In the case of a barrier option, the fine-resolution mesh can be grafted onto the coarse lattice nodes in the vicinity of the barrier to improve the price estimate. This has the effect of dramatically reducing the number of computations required to price the option in comparison to a normal trinomial tree model. It has the additional effect of implicitly determining the number of lattice levels required to achieve an accurate price approximation. Furthermore, layers of finer resolution mesh may be mutually overlaid to improve accuracy. However, owing to the difficulties in constructing a fine-resolution lattice, the AMM model encounters problems with options possessing sloped linear and nonlinear barriers. Consequently, it would be of interest to develop a generalized lattice-pricing model that can be readily applied to options with arbitrarily specified barrier types.

Option pricing using lattice techniques can be thought of as a trade-off between convergence rates and having to reposition the nodes of the lattice in proximity to the barrier. In the following, we propose a simple modification applicable to both the CRR binomial model and the trinomial tree model. Specifically, we propose an adjustment to the transition probabilities that eliminates the need to reposition the nodes of the lattice. Simultaneously, this approach yields significantly increased convergence rates compared to alternative models. To calculate the continuous-time option price we make use of a series of analytical methods and empirical results available in the barrier option pricing literature. We are thus able to compare our calculations to a known value.

The remainder of the paper is organized as follows. Section 2.2 briefly reviews the Kamrad and Ritchken model of the trinomial tree. Section 2.3 explains the adjustment to the transition probabilities of the trinomial tree. In section 2.4 we present the results of our convergence rate analysis for single time-varying barrier options and compare the results of our calculations to other models taken from the barrier option literature. Section 2.5 treats barrier options with multiple time-varying barriers and section 2.6 discusses exponential barriers. In section 2.7 we apply the method to a Bermuda option and compare our model to others. Section 2.8 provides some detail on computational complexity and time. In the final section we conclude.

2.2 Pricing Using a Trinomial Lattice

Appealing to the standard assumptions of lattice-based methods we assume that the price of the underlying asset follows a geometric Brownian motion process. Under this assumption the log-price drift is given by $\mu = r - \sigma^2/2$ where r is the risk-free rate and σ is the instantaneous volatility at time t. The trinomial lattice for the underlying asset is first constructed using the up and down movement probabilities that are given by

$$\begin{aligned} u &= \exp(\sigma\sqrt{\delta t}) \\ d &= \exp(-\sigma\sqrt{\delta t}) = 1/u \end{aligned} \tag{2.1}$$

where σ is the instantaneous volatility and δt is the size of the time-division of the lattice tree. Starting from a known initial asset price, S_0 , the underlying asset prices for the up, middle and down nodes at time $t+1$ are given by uS_0, S_0 and dS_0, respectively. A complete two-step trinomial lattice for the underlying asset is shown in Figure 2.1.

[Figure 2.1 about here]

In the usual trinomial lattice implementation, after the creation of the underlying tree the option tree is constructed as in Figure2.2.

[Figure 2.2 about here]

By working backwards through the option tree and appropriately discounting the calculated option prices one arrives at the lattice approximation to the option price. In such a trinomial tree, the option price evolves according to a set of lattice-specific probabilities such that (working backwards through the tree) the discounted option price, C , at time t is given by

$$C_t = e^{-r\delta t}(p_u C_{t+1,u} + p_m C_{t+1,m} + p_d C_{t+1,d}) \tag{2.2}$$

where p_u, p_m and p_d are the risk-neutral transition probabilities of the trinomial lattice, r is the risk-free rate and $C_{t+1,u}$, $C_{t+1,m}$ and $C_{t+1,d}$ are the call option prices for the up, middle and down nodes of the lattice, respectively. For the purposes of our implementation we use the following representation for the probabilities. Let

$$M = e^{r\delta t} \tag{2.3}$$

$$V = M^2(e^{\sigma^2\delta t} - 1) \tag{2.4}$$

then for the lattice probabilities we have:

$$p_u = \frac{((V + M^2 - M)e^{\lambda\sigma\sqrt{\delta t}} - (M-1))}{((e^{\lambda\sigma\sqrt{\delta t}} - 1)(e^{2\lambda\sigma\sqrt{\delta t}} - 1))} \tag{2.5}$$

$$p_d = \frac{(e^{2\lambda\sigma\sqrt{\delta t}}(V + M^2 - M) - e^{3\lambda\sigma\sqrt{\delta t}}(M - 1))}{((e^{\lambda\sigma\sqrt{\delta t}} - 1)(e^{2\lambda\sigma\sqrt{\delta t}} - 1))} \tag{2.6}$$

$$p_m = 1 - p_u - p_d \tag{2.7}$$

where λ is a trinomial lattice-specific parameter.[1] It is clear that the greater the number of time divisions taken for the tree, the greater the accuracy of the final approximation to the true price. Traditionally, this method of pricing options has been used extensively for pricing European as well American plain-vanilla options. The adjusted transition probability method involves modifying these probabilities in order to produce accurate price approximation for the case of barrier options. Furthermore, using the adjusted probability method it is also relatively simple to extend the procedure to options with time-varying barriers as well as more exotic types like the Bermuda option. We now discuss the probability adjusted extension of the trinomial lattice method as applied to path-dependent barrier options.

2.3 Option with a Linear Time-Varying Barrier

A barrier option is a path-dependent option whose payoff is determined by whether the price of the underlying asset has reached some pre-determined price level negotiated at the time of the contract purchase. For example, in the case of a down-and-out3 barrier option, the option payoff is set to zero when the underlying price falls below the barrier. A barrier option like this can be priced using the same trinomial tree method used to price plain-vanilla options, however, the trinomial tree will converge at an extremely slow rate to the true price of the option. As discussed, it is possible to reposition the nodes of the lattice to increase convergence but this becomes difficult for curved barriers. Compared to repositioning techniques, the transition probability adjustment method provides a more intuitive way to increase the convergence rate. The basic methodology has been used previously to increase the convergence rates of Monte Carlo option pricing algorithms. It is known that, due to the discretized path along which the asset price evolves, it is possible for the underlying asset price to breach the option barrier without being detected by the Monte Carlo simulation as discussed in Geman and Yor (1996). One way to alleviate this problem is to use the supremum of a Brownian bridge to calculate the probability that the underlying asset price touches the barrier for any given step of the simulation. However, this method is not without

[1] Although we implement the transition probability adjusted model on a trinomial tree, the model can easily be implemented on the standard binomial lattice with slightly less effort. In Table 2.1 and Table 2.2 we show results for both models.

its limitations. As noted in Baldi et al. (1999) this technique cannot be effectively used to price multiple barrier and time-varying barrier options. Consequently, Baldi derived a series of approximations for the exit probability of a Brownian bridge that can be used to price multiple and time-varying barrier options. Although he used these probabilities to improve upon Monte Carlo calculations, our contribution is to demonstrate that these probability approximations can be used to price options with single and multiple time-varying barriers on a simple trinomial lattice.

We describe our technique in the case of a down-and-out call option with a single linear time varying barrier. The same procedure applies to every type of barrier option, provided one knows the appropriate probability adjustment. Consider an option with a time-dependent, linear barrier. We define the barrier level L_t (lower barrier at time t) using the equation of a line and choosing a slope, m, and an intercept, b, as in Equation (2.8)

$$L_t = mt + b. \tag{2.8}$$

Note that the special case of a knock-out option with a constant barrier can be recovered by setting $m = 0$ in Equation (2.8). The situation for the time-varying barrier is illustrated in Figure 2.3.

[Figuret 2.3 about here]

The lattice of Figure 2.3 is indexed by the time division $\delta t = T/N$ where T is the option maturity and N is the number of divisions of the tree. Figure 2.4 shows a trinomial lattice with a time-varying linear barrier overlaid with two possible price paths labeled Diffusion Path 1 (DP1) and Diffusion Path 2 (DP2). These paths represent possible underlying price trajectories between times t and $t + \delta t$ on the lattice. It can be seen that both DP1 and DP2 breach the barrier between adjacent node layers of the lattice and thus represent situations in which the option payoff should be set to zero.

[Figure 2.4 about here]

Paths such as these are analogous to the Monte Carlo paths that break the barrier yet remain undetected by the calculation. Because the trinomial lattice only approximates a continuous-time diffusion, diffusion paths 1 and 2 are possible in continuous time, but are not well approximated by the discretized nature of the lattice. Similarly, the trinomial lattice will possess paths like those indicated yet the stock price can end up above the barrier at the next time step of the tree. In such cases the crossing of the barrier is not detected. To account for this effect it is necessary to modify the option price calculation in order to reduce the

expected payoff of the option by the total probability of all such possible diffusion paths. The probabilities of these diffusion paths are related to the exit probabilities calculated by Baldi et al. (1999).

We model a diffusion path of the logarithm of the stock price between lattice time divisions as a Brownian bridge. A Brownian bridge is a stochastic process in continuous-time whose probability distribution is given by the conditional distribution of a Weiner process. In particular, the starting and ending points of the process are known quantities. Mathematically, a Brownian bridge is an R^n-valued Gaussian process $(S_t)_{0\leq t\leq T}$ for which we can define an exit probability, ρ. Let $\rho_i^{\delta t}$ be the probability that S_t breaches the linear barrier, i (with $i = L$ for the lower barrier and $i = U$ for the upper barrier), in the lattice time interval $[t, t + \delta t)$. More formally, it is the probability that the barrier is breached by a Brownian bridge that starts at $\log S_t$ at time t and is conditional on it reaching $\log S_{t+\delta t}$ at time $t + \delta t$. In order to correct for the possibility that the diffusion path breaches the barrier we subtract this probability from unity so that we obtain the probability that the diffusion path does not breach the barrier in the interval $[t, t+\delta t)$. The correction is thereby obtained by multiplying the usual trinomial lattice probabilities by $(1 - \rho_i^{\delta t})$. This is what we imply by the phrase transition probability adjustment.

We illustrate the method using a barrier option with a barrier defined as in Equation (2.8). Using the subscript to denote a lower (i.e. down-and-out) barrier, the exit probability for this type of barrier is given by Baldi et al. (1999) as:

$$\begin{aligned}&\rho_L^{\delta t}(T_0, S_{T_0}, S_{T_0+\delta t}) = \\ &\exp\Big[-\frac{2}{\sigma^2}\log\Big(\frac{S_{T_0}}{L_{T_0}}\Big)\Big(\frac{1}{\delta t}\log\Big(\frac{S_{T_0+\delta t}}{L_{T_0}}\Big) + \frac{m}{b + m\delta t}\Big)\Big]\end{aligned} \tag{2.9}$$

where S_{T_0} is the asset price at the current lattice node, $S_{T_0+\delta t}$ is the asset price at the next lattice node and L_{T_0} is the barrier level at the current node. The numerical procedure involves correcting the lattice probabilities for the diffusion paths in the following manner. First we price the underlying tree using the normal trinomial tree method. At each node of the lattice we also calculate the level of the barrier associated with that particular node using the equation for the barrier, in this case Equation (2.8). Subsequently, we work backwards through the option tree correcting for the exit probabilities, or rather we adjust the transition probabilities. At each node in the trinomial lattice there are three adjustments to perform for the up, mid and down transitions. The calculation of the option price at a given node is

shown in Equation (2.10).

$$\begin{aligned} C(S_{T_0}) = \exp -r\delta t(p_u(1-\rho^{\delta t}_{L,up})C(S^{up}_{T_0+\delta t}) \\ +p_m(1-\rho^{\delta t}_{L,mid})C(S^{mid}_{T_0+\delta t}) \\ +p_d(1-\rho^{\delta t}_{L,down})C(S^{down}_{T_0+\delta t})) \end{aligned} \tag{2.10}$$

where $C(S_{T_0})$ is the call option price corresponding to the node at which the barrier level is L_{T_0}. After iterating through the tree in this manner and performing the necessary adjustments we arrive at the final approximation to the option price.

2.4 Options with a Single Linear Time-Varying Barrier

To start with, we consider the special case of a single constant barrier option (by setting $m = 0$ in Equation (2.8)). We study the behaviour of the probability adjustment method when the stock price approaches the barrier. Under these circumstances many lattice-based option valuation techniques have difficulty producing an approximation to the option price. In the case of the standard binomial tree, convergence is so slow that even after 5000 time divisions, there is significant difference between the approximate and analytical values. Even Ritchkens stretched trinomial tree method encounters difficulty in pricing a down-and-out call when the stock price approaches the barrier. For example, see Table 2.1. Ritchkens method first encounters difficulty at 500 time-divisions and a stock price of 91.0. Furthermore, when the initial stock price is very close to the barrier, even as many as 5000 iterations are unable to provide an approximation to the option price.

[Table 2.1 about here]

Conversely, because the probability adjustment technique is based on the lattice method with no node repositioning, we are always able to produce an approximation to the option price regardless of the stock price to barrier distance. Table 2.1 presents our option price approximations in comparison to those of Ritchkens as the stock price approaches the barrier. At a stock value of 90.3, Ritchkens method is unable to produce an approximation to the price in 5000 divisions or less. Conversely, we obtain an approximation with 500 time- divisions. Furthermore, we are able to obtain price approximations when the stock price is extremely close to the barrier. Additionally, the quality of the approximation improves with decreasing distance between the stock price and barrier and, in the extreme case of a stock price of 90.01, we have very good agreement between our approximation and the analytical price for the adjusted trinomial model.

We now turn to the general case of a linear time varying barrier. To provide a numerical example we select some values for the slope and intercept of the barrier Equation (2.8). We set $S_0 = 100, K = 100, \sigma = 25\%, r = 10\%, T = 1, m = 10$ and $b = 95$. In Figure 2.5 we plot the convergence curve for the linear barrier option with the parameters specified above.

[Figure 2.5 about here]

It is clear that the convergence of the adjusted tree is considerably faster than that of the normal (unadjusted) trinomial tree. Furthermore, the small fluctuations visible in the crests of the unadjusted curve are almost non-existent in the adjusted price curve. Unfortunately, for this case we do not possess an analytical solution to which we can compare our results.

2.5 Options with Two Linear Time-Dependent Barriers

Extending the example above, we now add an additional upper barrier, resulting in an option with two linear time-dependent barriers. Unlike the single barrier case, the probability adjustment must now account for both the upper and lower barriers. The appropriate exit probabilities for the upper, U_t, and lower, L_t, barrier were calculated by Baldi et al. (1999) and are given in Equation (2.11):

$$\begin{aligned}
&\rho_L^{\delta t}(T_0, S_{T_0}, S_{T_0+\delta t}) = \\
&\exp\Big[-\frac{2}{\sigma^2} \log\Big(\frac{U_{T_0}}{S_{T_0}}\Big)\Big(\frac{1}{\delta t}\log\Big(\frac{U_{T_0}}{S_{T_0+\delta t}}\Big) + \frac{m_{upper}}{b_{upper} + m_{upper}\delta t}\Big)\Big], \\
&\rho_L^{\delta t}(T_0, S_{T_0}, S_{T_0+\delta t}) = \\
&\exp\Big[-\frac{2}{\sigma^2} \log\Big(\frac{S_{T_0}}{L_{T_0}}\Big)\Big(\frac{1}{\delta t}\log\Big(\frac{S_{T_0+\delta t}}{L_{T_0}}\Big) + \frac{m_{lower}}{b_{lower} + m_{lower}\delta t}\Big)\Big],
\end{aligned} \tag{2.11}$$

where the previous variable definitions still apply and U and L are the values of the upper and lower barrier at time δt, respectively. To price the double barrier option the underlying tree is first constructed. For each time division of the lattice we calculate the level of both the upper and lower barriers using the equations that define the barrier. Once this is completed we can continue to price the option by calculating the option payoffs at maturity. These are calculated at the last time division of the lattice. We then work backwards through the tree calculating the option price at each given node using a slightly modified version of Equation (2.10) for the price of the option. If, at a given node, the underlying price is either above U or below L we set the option value to zero, since the asset price has breached the barrier. Equation (2.12) gives the probability adjustment, ρ, for the double barrier option case. The

adjustment is a linear combination of the exit probabilities defined in Equation (2.11) for the upper and lower transitions:

$$\rho = 1 - \rho_U^{\delta t} = \rho_L^{\delta t}. \tag{2.12}$$

For the double time-varying barrier option it suffices to multiply the trinomial lattice probability by Equation (2.12) in order to effect the probability adjustment. To provide a numerical example, we choose the following set of parameters to describe the option: $S_0 = 100, K = 100, \sigma = 25\%, r = 10\%, T = 1, m_{lower} = -22, b_{lower} = 92, m_{upper} = 35$ and $b_{upper} = 105$. Figure 2.6 shows the convergence curves for the double linear time-varying barrier option.

[Figure 2.6 about here]

We do not have an analytical value for the double time-varying barrier option either, yet it is clear that the plot converges to a constant value. The convergence is relatively stable after 1000 time divisions of the lattice and convergence is near-monotonic. The plot exhibits almost no periodical fluctuations in approaching a constant value for the option price. Notably, the unadjusted curve converges much more slowly to, presumably, the same value. However, an analytical or independent empirical result is needed to confirm the accuracy of our calculation. To judge the goodness of our method we can consider the special case of a knock-out option with double constant barriers. Although there is no analytic formula available, this type of option has been widely investigated in the option pricing literature. Therefore, we can compare our results with other numerical approximations. The data presented in Table 2.2 are taken from Pelsser (2000). It is clear from the table the probability-adjusted method is in good agreement with the values obtained using other methods in the existing barrier option pricing literature. Specifically we consider the Kunitomo and Ikeda (1992), finite difference and Pelsser (2000) methods. We conclude that for short-term maturity options, the model demonstrates accurate approximations to the option price within 2000 time-divisions of the lattice. Furthermore, these results do not differ significantly from those calculated using 1000 divisions of the tree.

[Table 2.2 about here]

2.6 Options with Exponential Barriers

We now turn our attention to the case of a barrier option with a time-varying exponential barrier. The probability adjustment is again given by Baldi et al. (1999) and is defined in Equation (2.13). This gives the exit probability in terms of the option parameters. As in the

linear case, the barrier is characterized by its slope, m, intercept, b and current lattice time-step, T_0:

$$L_t = \exp mt + b.$$

The probability correction is given by:

$$\begin{aligned} &\rho_L^{\delta t}(T_0, S_{T_0}, S_{T_0+\delta t}) = \\ &\exp\Big[-\frac{2}{\sigma^2}(\log(S_{T_0}) - b - mT_0)\Big(\frac{\log(S_{T_0+\delta t}) - b - mT_0}{\delta t} - m\Big)\Big] \end{aligned} \tag{2.13}$$

The parameters for the calculation are: $S0 = 95, K = 100, = 25\%, r = 10\%, T = 1$ year, exponential barrier slope = 0.05, exponential barrier intercept = 90. The analytical value is 5.4861.

Having specified the barrier and the parameters of the option we calculated the option price approximation for all time divisions from 1 to 1000 using the aforementioned procedure. The results are plotted in Figure 2.7.

[Figure 2.7 about here]

Because the exponential barrier option is priced using the same numerical technique that was applied to the single barrier option calculation times remain consistent and take approximately 4 seconds. After approximately 600 time-divisions of the tree, we note the decreased oscillations of the probability-adjusted model compared to that of the trinomial tree model. Convergence towards the analytical value of 5.4861 is also considerably faster. To judge the performance of our model for the exponential barrier option against another model, we compare our results to the empirical results published by Costabile (2002). The results are shown in Table 2.3.

[Table 2.3 about here]

Usually, Costabile' s method converges slightly faster than the adjusted probability technique due to the fact that he is repositioning the nodes of his lattice whereas our method does not. As mentioned, we know that methods based on lattice manipulation exhibit faster convergence. Nonetheless, Costabiles method is specific to single exponential barrier options only yet the adjusted method can handle a multitude of barrier types and is not restricted to a particular option type.

2.7 Application to Bermuda Options

A Bermuda option is a discretely monitored barrier option whose payoff is zero if the underlying price is below the barrier at some discretely monitored time, t^*. Figure 2.8 shows the

diffusion paths that must be considered in order to apply the transition probability adjustment method.

[Figure 2.8 about here]

We can see two possible types of diffusion paths for the underlying price; a safe path and a knock-out path. Along the safe diffusion paths, the underlying price (even though discretely monitored) never falls below the barrier in continuous-time, yet in discrete time it appears to be a knock-out path. Looking now at the discretely monitored knock-out paths, the price of the underlying asset actually does fall below the barrier for some monitored time t^*. At this particular time instant the option does, in fact, knock-out which results in a payoff of zero. In applying the transition probability method the situations for both safe type paths and knock- out type paths must be handled in order to achieve convergence to a fixed price. This is the basis of our method in the Bermuda option case.

The method for pricing a Bermuda option under transition probability adjustment proceeds as follows. First the underlying tree is constructed in the usual manner. We then determine the monitoring frequency of the option. This monitoring frequency provides us with a discrete set of times, $\{t^*\}$, at which we determine whether the barrier has or has not been breached. When these monitoring dates lie between the time partitions of the lattice we adjust the lattice probabilities. This situation is illustrated in Figure 2.9.

[Figure 2.9 about here]

In the downward transition from S_t to $S_{t+1,d}$ the probability that the option expires is equal to the probability that a Brownian bridge between S_t and $S_{t+1,d}$ falls, at time t^*, below the barrier. A complication arises when the barrier coincides with a time partition of the lattice as shown in Figure 2.10.

[Figure 2.10 about here]

In this case the exit probabilities cannot be calculated and the method suffers from discretization error. For the purpose of illustration, let the barrier level of 2.8 be 95, the initial asset price, $S_t = 100$ and $S_{t+1,d} = 90$. In this scenario we would have to calculate the probability that the diffusion process falls below 95 at time t^* having started at S_t and ended at $S_{t+1,d}$. However, this probability equals unity when the barrier exceeds 90. The underlying problem is that the option payoff on the trinomial tree is unaffected by the level of the barrier. Using the above example, the payoff of the option would be calculated as:

$$C_t = e^{r\delta t}(p_u C_{t+1,u} + p_m C_{t+1,m} + p_d 0). \tag{2.14}$$

Equation (2.14) produces the same result for the price of the option, C, even if the barrier was placed at 90.01 since $C_{t+1,d}$ would still be zero. This result manifests itself as a large bias in the approximation to the option price.

To avoid this discretization problem we can restrict the computation of the trinomial tree, only to a number of steps such that the lattice time division does not coincide with barrier monitoring times. As a general guideline, in practice it is best to avoid choosing multiples of the prime factors of the monitoring frequency in order to eliminate this problem. Thus, if the monitoring frequency, , is selected to be weekly and we assume 50 weeks in a year, it suffices to select a number of steps such as $\{51, 53, 57, 59, \cdots\}$. Since the prime factors of 50 are 2 and 5, we do not compute the trinomial tree for steps that are multiples of 2 and 5.

The pricing of the Bermuda option is not significantly different from the single linear barrier case, the main difference being that a slight amount of additional effort is required to account for the discrete monitoring times as previously discussed. The probability adjustment is given by the cumulative normal distribution which is the probability distribution of a Brownian

bridge between two successive points of the lattice. For the Weiner process $\{W_t\}$ that starts at $W(t_1) = \log(S_t) = a$ and ends at $W(t_2) = \log(S_{t+\delta t}) = b$, its distribution in time $t \in (t_{1,t_2})$ is normal with mean:

$$\mu = a + \frac{t - t_1}{t_2 - t}(b - a) \tag{2.15}$$

and variance

$$\sigma^2 = \frac{(t - t_1)(t_2 - t)}{t_2 - t_1}. \tag{2.16}$$

The exit probability adjustment for the Bermuda option is therefore the cumulative normal distribution with mean, μ, and variance, σ^2, as defined in Equations (2.15) and (2.16). The distribution function is evaluated at the log price of the underlying asset at the current lattice node. The option price is then calculated using the trinomial lattice and the procedure described for the previous option types. To illustrate, we provide several numerical examples and compare the results to existing techniques for Bermuda options in the literature.

Broadie et al. (1997) derive a continuity correction for discretely monitored barrier options using the Reimann zeta function. This allows them to price a Bermuda option accurately, yet their method breaks down when the underlying price is close to the barrier or the barrier is time varying. A comparison between the adjusted transition probability method with the continuity correction method is given in Table 2.4.

[Table 2.4 about here]

The table shows the results of the adjusted probability method compared to the values published in Broadie et al. (1997) for the corrected continuous pricing method. The option parameters are $S_0 = 100, K = 100, \sigma = 30\%, r = 10\%$ and $T = 1$. The number of monitoring times for the corrected continuous method is 50 (with $T = 0.2$) and for the adjusted method this translates into 250 annual monitoring points. These values correspond to daily monitoring of the option by assuming 250 trading dates per year. Values of the true price are taken from Broadie et al. (1997). and are calculated using a specialized trinomial tree as described in Broadie et al. (1999). Both pricing methods perform well, but when the underlying price is close to the barrier, the probability-adjusted method has a much smaller value for the relative error illustrating its ability to accurately price options for which the initial underlying asset price is close to the option barrier.

Naturally, we can reduce the monitoring frequency and therefore reduce the number of probability adjustments in the trinomial lattice. In doing so we can compare the adjusted model to additional results by Broadie et al. (1997). Table 2.5 shows the comparison of the model results for reduced monitoring frequency (the other parameters are equal to the previous example).

[Table 2.5 about here]

The increased accuracy of the transition probability adjusted trinomial tree over the corrected-continuous method is immediately evident when the underlying asset price is close to the barrier. The increase in the relative error of the corrected-continuous model in the case is pronounced even several price units away from the barrier.

To provide a graphical depiction of the convergence properties of the probability adjusted method applied to a Bermuda option, in Figure 2.11 we provide a series of convergence plots for various barrier levels.

[Figure 2.11 about here]

Panels (a) through (c) show convergence graphs for barrier levels, L, of 95, 99.5 and 99.9, respectively with weekly barrier monitoring. The remaining parameters are taken from Table 2 in Duan et al. (2003): $S_0 = 100, K = 100, \sigma = 20\%, r = 10\%, T = 0.5$ and $\lambda = 1.2$. Panels (a)-(c) clearly show the convergence of the probability adjusted method to the analytic price which is shown on the plots as a straight line. Fluctuations about the true price are small and in most cases are within less than one-tenth of a unit of the option price. Some oscillations of increasing period do remain, however. Panel (d) shows the convergence plot for Example 1 as published in Ahn et al. (1999). The parameters for this calculation are taken to be

$S_0 = 40, K = 40, \sigma = 40\%, r = 4.88\%, T = 1, L = 35$ and a single monitoring time at $T = 0.5$. The convergence to the true price in this case is near- monotonic and occurs rapidly. There are no noticeable oscillations in this particular case. Combined, panels (a)-(d) illustrate the flexibility of the method for pricing options of various maturities and various barrier levels; including barrier levels that are very close to the initial underlying asset price.

2.8 Numerical Performance

All calculations were implemented in the C++ programming language. Figure 2.12 shows the CPU time in seconds spent on the adjusted trinomial option tree and the CPU time in microseconds spent on the adjusted trinomial underlying tree. Also shown is the total combined CPU time of the calculation. All computations were performed on a 1.83 GHz CPU with 2 GB of RAM.

[Figure 2.12 about here]

The CPU time spent on the underlying tree is negligible and on the order of microseconds. Consequently, it is evident that the bulk of the CPU time is spent on generating the option tree. In fact, total calculation time and CPU option tree time are indistinguishable on the figure. Calculation time increases rapidly after 4000 lattice levels and at 10000 lattice levels, the calculation time is approximately 47 seconds. A typical calculation for 1000 lattice divisions and a time-varying barrier requires approximately 4 seconds.

2.9 Conclusion

In this paper we have presented a generalized method for pricing numerous types of barrier options that is based on a simple modification of the trinomial tree model. With minimal effort, the calculations can readily be implemented on a binomial lattice with little to no change in accuracy. This also has the effect of decreasing calculation time owing to the reduced number of lattice paths. Additionally, the model can be applied to any option as long as the exit probability can be calculated (or approximated) in a closed-form. Our results for time-varying barrier options are promising and demonstrate good convergence properties towards the continuous-time price of the option. More specifically, we have shown that it is possible to produce option price approximations even when the initial underlying price is very close to the barrier, a result that is often difficult to obtain using alternative models in the

literature. Furthermore, the model produces accurate price approximations for options with short maturity.

We have applied the transition probability method to several types of time-varying barrier options including those with exponential, single linear and double linear barriers. A lack of closed-form option valuation equations for some option types renders it difficult to precisely gauge the accuracy of our approximations in the case of the single linear and double linear time-varying barriers. Nonetheless, the price appears to converge to a fixed value based on the convergence plots. The accuracy of the option price approximations produced by the model dominates that of the Kamrad and Ritchken model while avoiding any repositioning of the lattice nodes. This makes it expedient and simple to implement even for complex option types.

Application of the model to discretely monitored Bermuda options also produces good results with no loss in pricing accuracy when the barrier level is close to the underlying asset price. Given its straightforward implementation, the model outperforms more complicated pricing models and convergence to the true price is near to monotonic, a desirable property for an asset pricing model.

While no single lattice based pricing model can be used to estimate option prices with arbitrary accuracy, a given model selected from the literature can be used either as a general purpose pricing model or as a specialized pricing model tailored to a specific option. What is important is that the practitioner understands the advantages and shortcomings of any particular option pricing model.

2.10 Figures and Tables

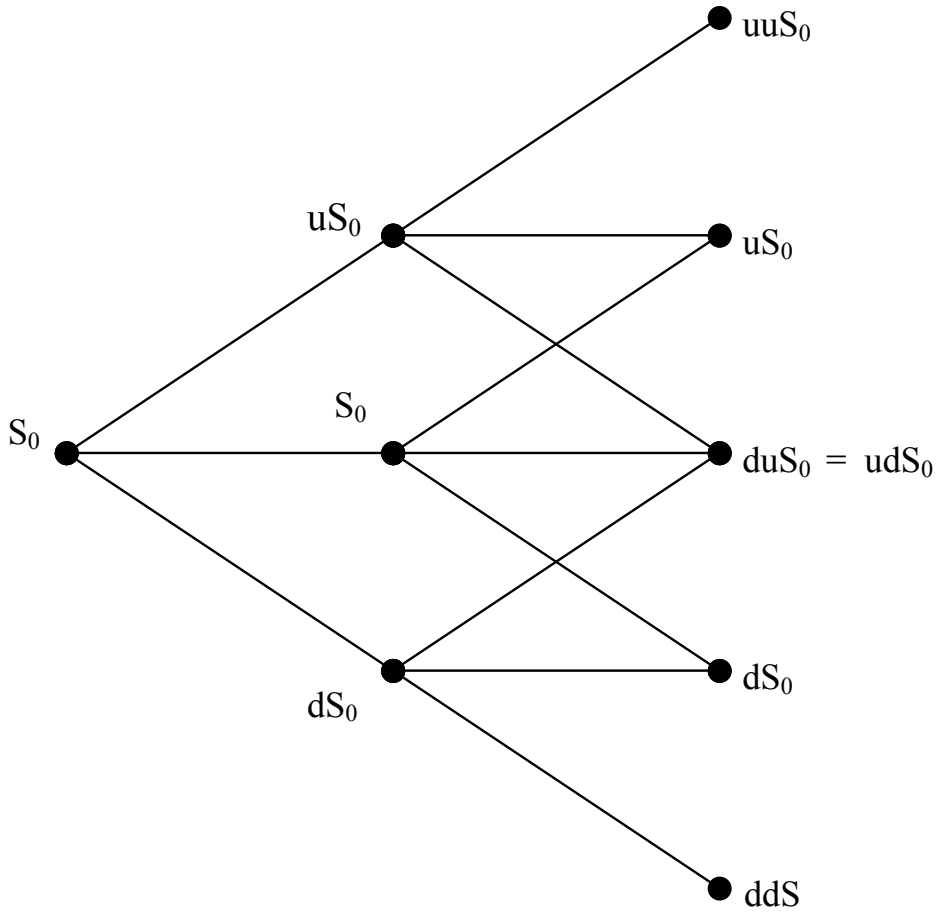

Figure 2.1: **Evolution of the underlying trinomial tree.**

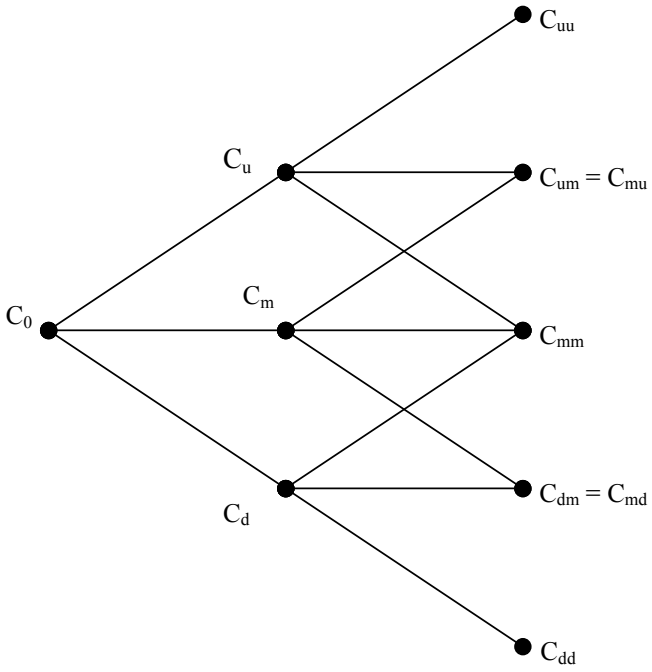

Figure 2.2: **The option price trinomial tree lattice associated with the underlying assets tree.**

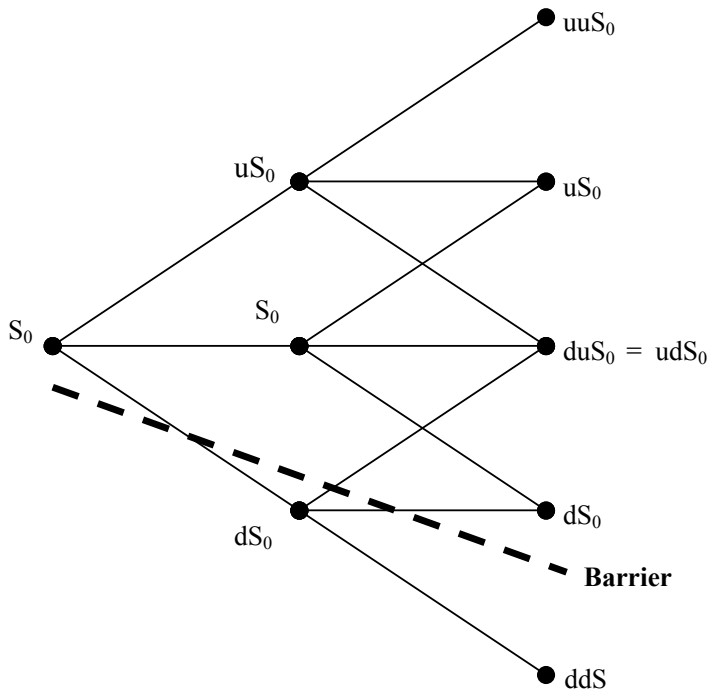

Figure 2.3: **The trinomial lattice for an option with a linear time-varying barrier.**

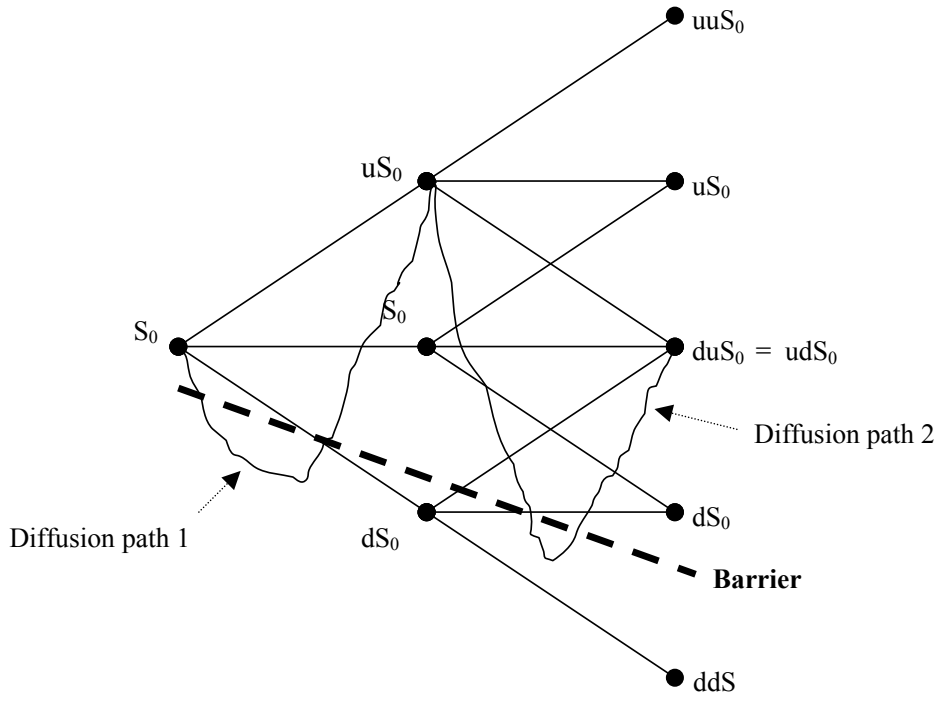

Figure 2.4: **Trinomial lattice with diffusion paths that traverse the barrier between lattice nodes.**

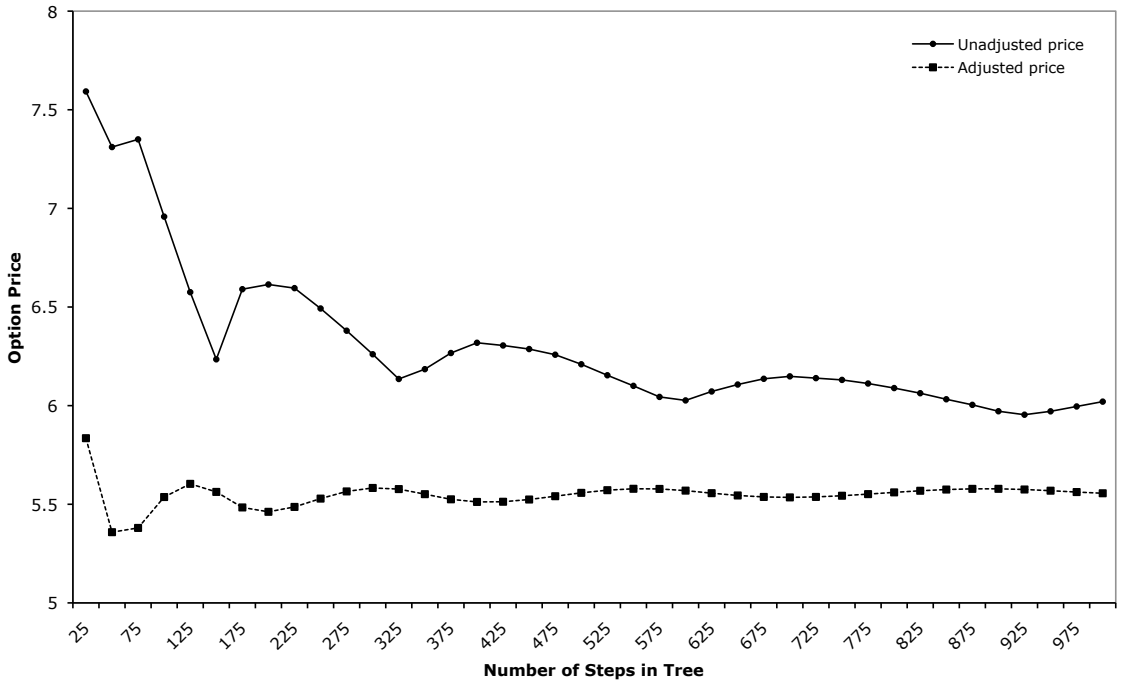

Figure 2.5: **Convergence plot for the single linear time-varying barrier option.**

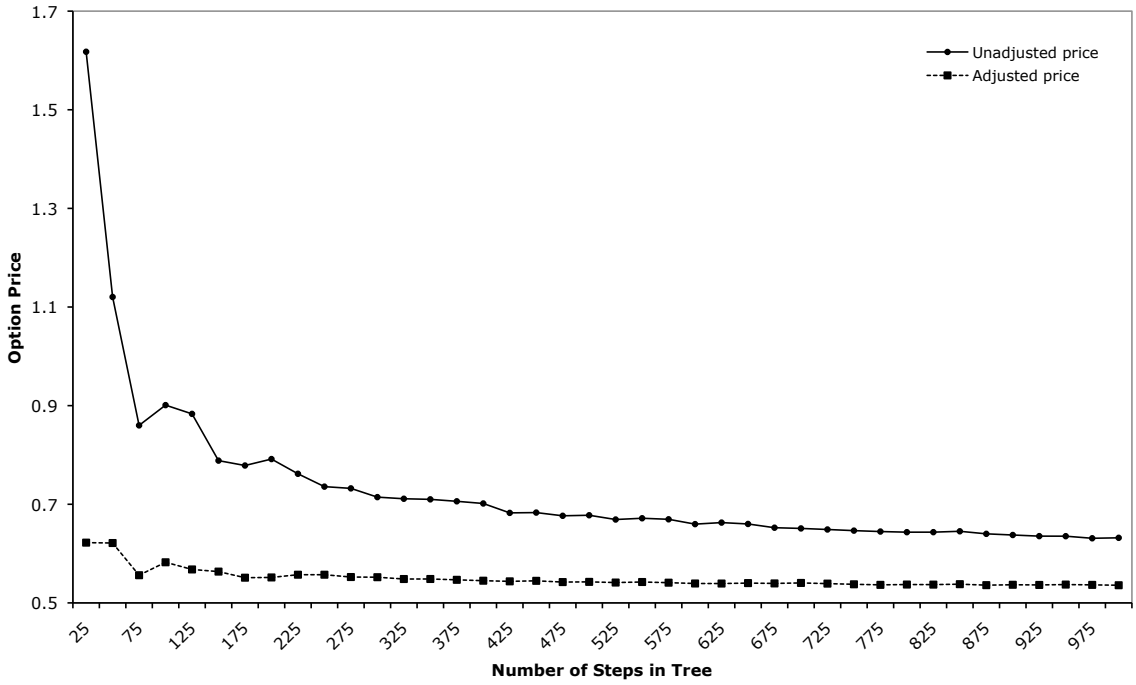

Figure 2.6: **Convergence plot for the double linear barrier option.**

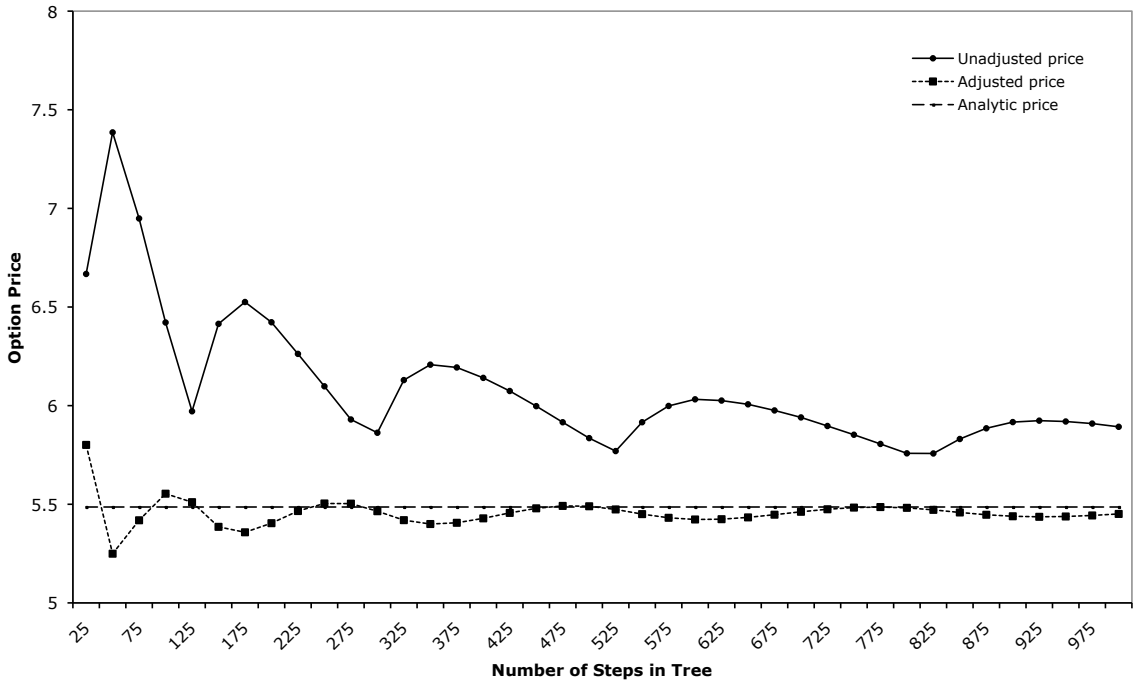

Figure 2.7: **Results of the convergence calculation for an exponential barrier option.**

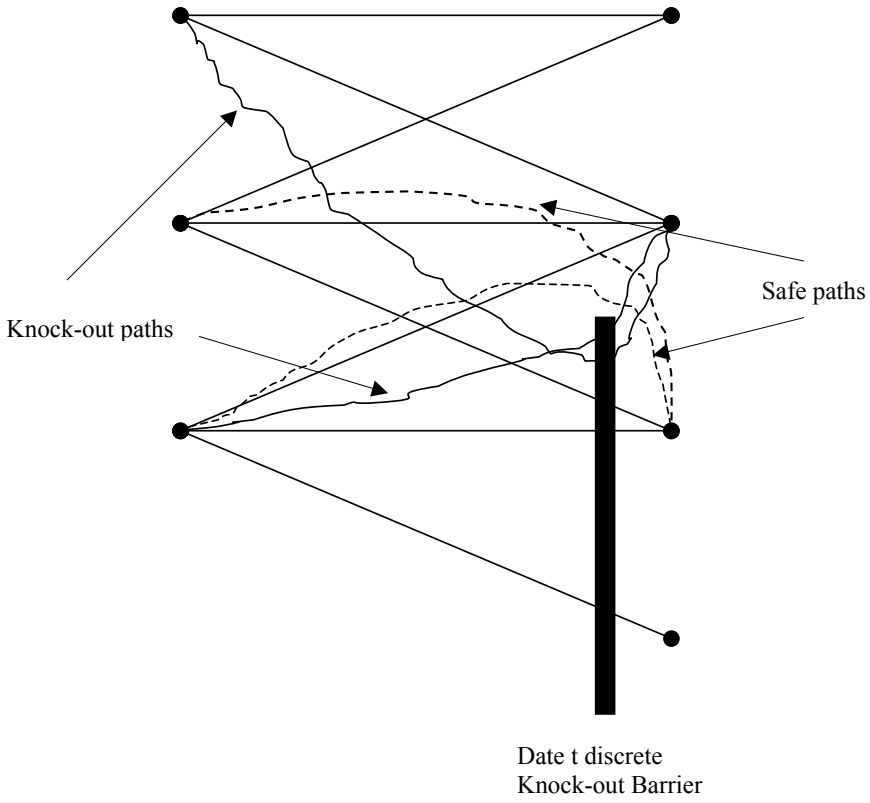

Figure 2.8: **The two types of possible diffusion paths.**

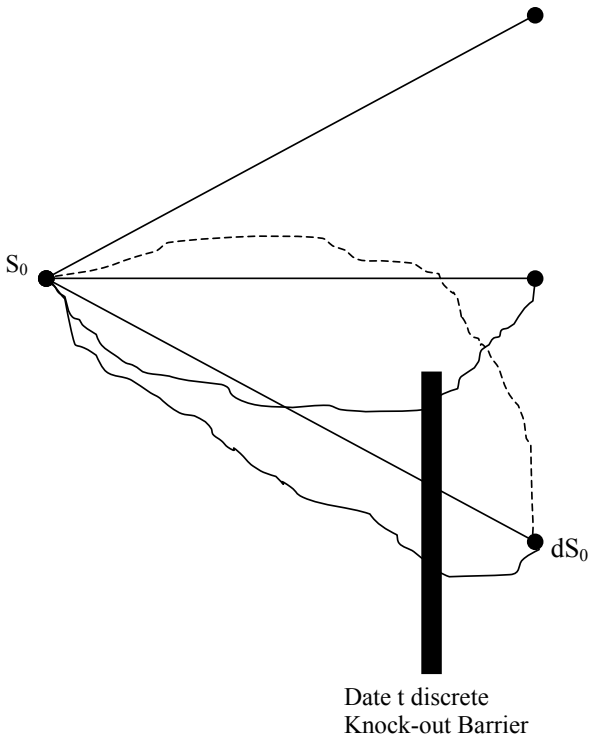

Figure 2.9: **Bermuda options.**When the discrete monitoring time, t^*, lies between the lattice nodes, the probability correction is applied.

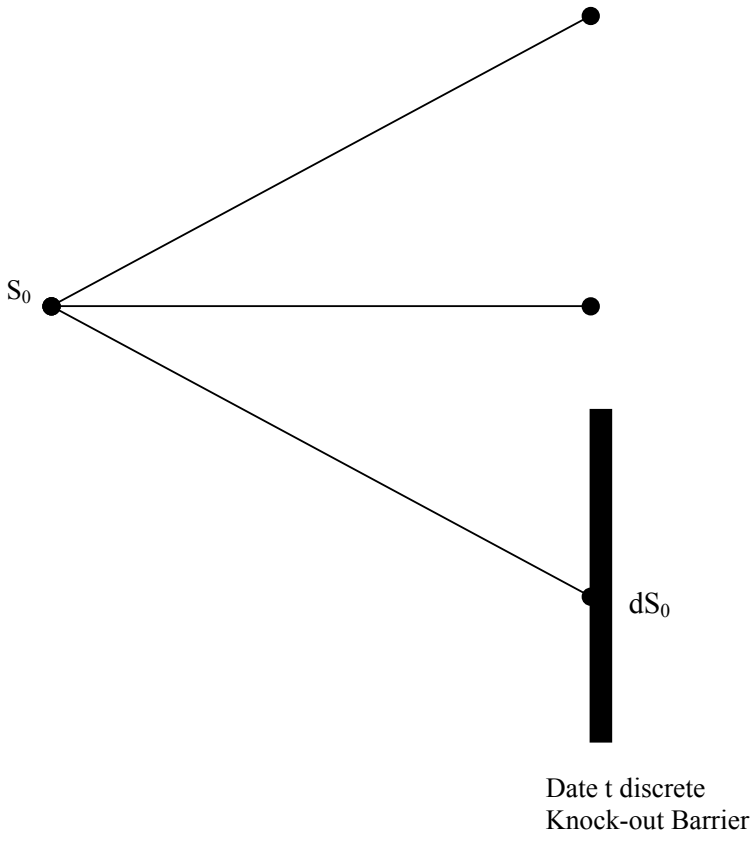

Figure 2.10: **Bermuda options.**A complication arises when the barrier and a layer of lattice nodes coincide at a discrete monitoring time, t^*.

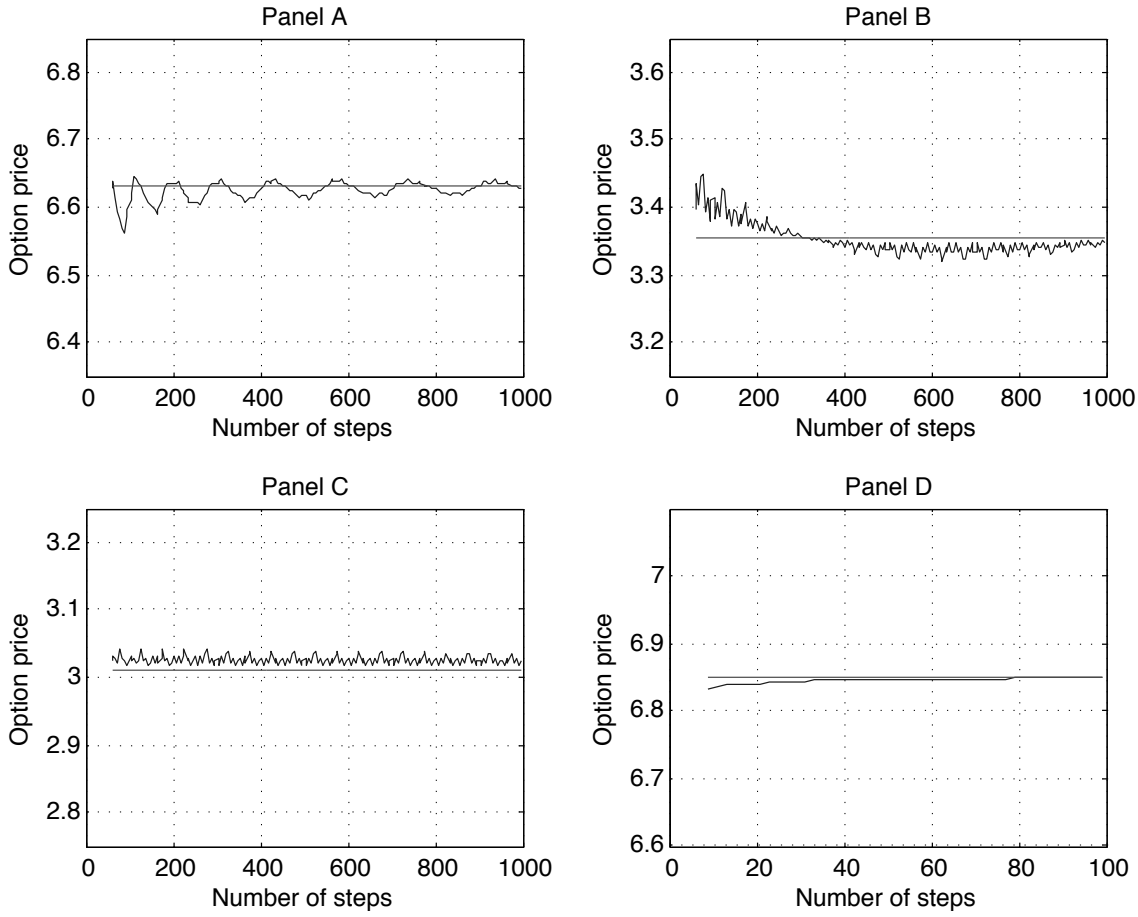

Figure 2.11: **Convergence plot.** Panels A through C show convergence plots for various starting barrier levels with weekly monitoring. The parameters are taken from Exhibit 2 in Duan et al. (2003): $S_0 = 100, K = 100, \sigma = 20\%, r = 10\%, T = 0.5, \lambda = 1.2$. Panel D shows a comparison with the method a Ahn et al. (1999). The parameters for this calculation are taken to be $S_0 = 40, K = 40, \sigma = 40\%, r = 4.88\%, T = 1, L = 35$ and a single monitoring time at $T = 0.5$.

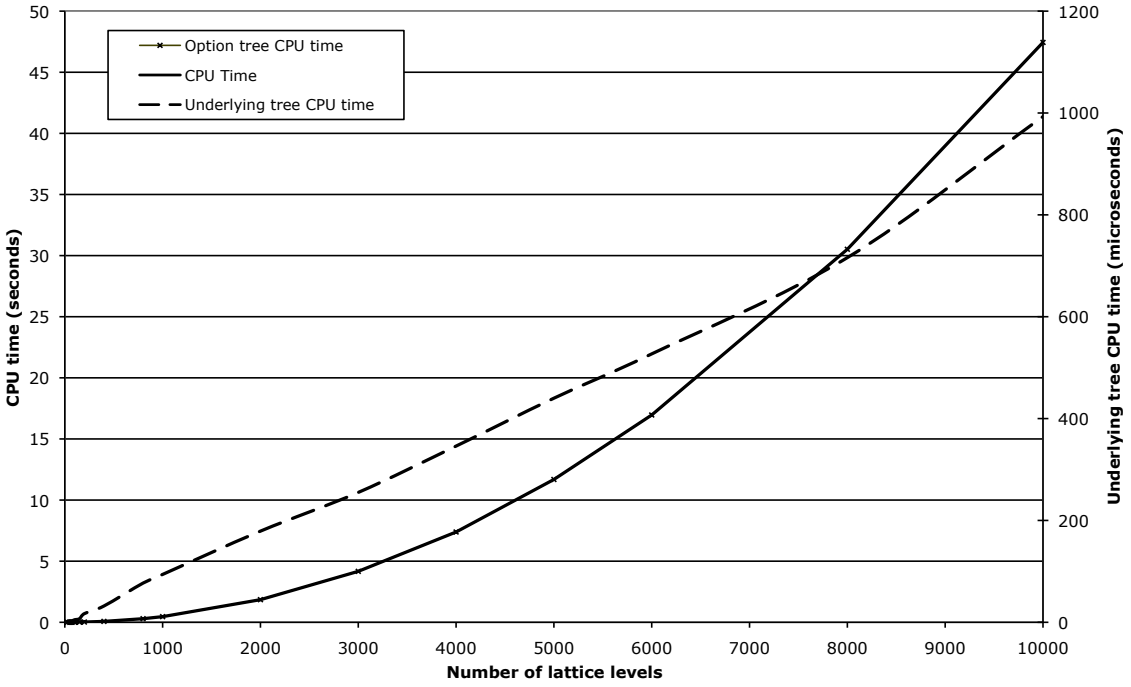

Figure 2.12: **CPU times for the underlying asset lattice and the option price tree.**

	Number of Time Steps in the Tree						
Stock Price	500	1000	2000	3000	4000	5000	Analytic Price
94.0	4.849/ 4.723 (4.863)	4.814 /4.839 (4.864)	4.866 /4.811 (4.864)	4.840 /4.880 (4.864)	4.839 / 4.845 (4.864)	4.82 / 4.833 (4.864)	4.864
93.0	3.667/3.610 (3.700)	3.601 / 3.634 (3.701)	3.697 / 3.722 (3.702)	3.641 / 3.708 (3.701)	3.649 / 3.717 (3.701)	3.654 / 3.703 (3.702)	3.702
92.0	2.45 / 2.526 (2.504)	2.509 / 2.398 (2.506)	2.487 / 2.425 (2.506)	2.507 / 2.497 (2.506)	2.488 / 2.502 (2.506)	2.454 / 2.464 (2.506)	2.506
91.5	1.8 / 1.726 (1.894)	1.799 / 1.930 (1.894)	1.868 / 1.850 (1.895)	1.885 / 1.877 (1.895)	1.836 / 1.863 (1.895)	1.893 / 1.907 (1.895)	1.895
91.0	1.222 / 1.333 -	1.176 / 1.155 (1.274)	1.239 / 1.286 (1.274)	1.221 / 1.248 (1.275)	1.275 / 1.217 (1.275)	1.224 / 1.272 (1.274)	1.274
90.5	0.716 / 0.669 -	0.707 / 0.695 -	0.614 / 0.672 -	0.57 / 0.612 -	0.593 / 0.580 (0.642)	0.626 / 0.590 (0.642)	0.642
90.4	0.561 / 0.521 -	0.576 / 0.544 -	0.549 / 0.558 -	0.498 / 0.542 -	0.462 / 0.512 -	0.458 / 0.480 (0.515)	0.515
90.3	0.409 / 0.380 -	0.424 / 0.393 -	0.433 / 0.412 -	0.423 / 0.419 -	0.403 / 0.418 -	0.381 / 0.410 -	0.387
90.2	0.263 / 0.248 -	0.271 / 0.252 -	0.281 / 0.260 -	0.287 / 0.267 -	0.289 / 0.273 -	0.288 / 0.276 -	0.258
90.1	0.127 / 0.122 -	0.129 / 0.122 -	0.131 / 0.123 -	0.129 / 0.124 -	0.135 / 0.126 -	0.137 / 0.127 -	0.129
90.05	0.063 / 0.061 -	0.063 / 0.061 -	0.064 / 0.061 -	0.064 / 0.061 -	0.064 / 0.061 -	0.065 / 0.061 -	0.065
90.01	0.012 / 0.012 -	0.012 / 0.012 -	0.012 / 0.012 -	0.012 / 0.012 -	0.012 / 0.012 -	0.012 / 0.012 -	0.013

Table 2.1: The last column contains the price of the down-and-out call option using an analytica solution. The prices are displayed as adjusted binomial price / adjusted trinomial price. Prices in brackets are those given in Ritchken (1995). Prices indicatd by "-" were unable to be computed using the node repositioning method of Ritchken; that is, the number of partitions used were insufficient to produce an approsimation to the analytical price.

Vol	U	L	KI	Ana	FD	Approx 1	Approx 2
	1500	500	25.12	25.12	24.57	25.12 / 25.12	25.12 / 25.12
$\sigma = 0.2$	1200	800	24.76	24.76	24.69	24.77 / 24.76	24.76 / 24.76
	1050	950	2.15	2.15	2.15	2.14 / 2.15	2.15 / 2.15
	1500	500	36.58	36.58	36.04	36.59 / 36.58	36.59 / 36.58
$\sigma = 0.3$	1200	800	29.45	29.45	29.40	29.42 / 29.40	29.42 / 29.46
	1050	950	0.27	0.27	0.27	0.27 / 0.26	0.26 / 0.27
	1500	500	47.85	47.85	47.31	47.86 / 47.85	47.85 / 47.84
$\sigma = 0.3$	1200	800	25.84	25.84	25.82	25.85 / 25.88	25.83 / 25.86
	1050	950	0.02	0.02	0.01	0.01 / 0.01	0.01 / 0.01

Table 2.2: The option parameter used are $S_0 = 1000, K = 1000, r = 5\%$ and $T = 1/12$. "Approx1" is calcutated for 1000 time-divisions of the adjusted tree, and "Approx2" is calculated using 2000 time-divisions of the adjustd tree. Column "U" gives the upper barrier while column "L" gives the lower barrier. "KI" is calculated using the method of Kunitomo and Ikeda (1992); "Ana" is the results of Pelsser (2000); "FD" is the finite difference calculation based on a 1000 by 1000 grid.

Slope = m = -0.1			Slope = m = 0.1		
Tree Lvls.	Costabile	Adjusted	Tree Lvls	. Costabile	Adjusted
17	7.002	6.631	24	5.020	5.400
77	6.958	6.753	92	4.949	4.985
181	6.920	6.787	203	4.934	4.876
327	6.910	6.786	356	4.934	4.877
515	6.912	6.802	552	4.932	4.912
2100	6.902	6.868	2174	4.929	4.915
4754	6.900	6.889	4865	4.929	4.918
Analytic	6.896		Analytic	4.928	

Table 2.3: The adjusted-probability method compared with the extended Cox-Ross-Rubinstein method of Costabile (2002).

Barrier	Continuous Barrier	Corrected Continuous	Probability Adjusted	True Price	Broadie Rel. Err. (%)	Adj. Rel. Err. (%)
85	6.308	6.322	6.322	6.322	0	0
86	6.283	6.306	6.306	6.306	0	0
87	6.244	6.281	6.281	6.281	0	0
88	6.185	6.242	6.242	6.242	0	0
89	6.099	6.184	6.184	6.184	0	0
90	5.977	6.098	6.097	6.098	0	-0.02
91	5.808	5.977	5.976	5.977	0	-0.02
92	5.579	5.810	5.811	5.810	0	0.02
93	5.277	5.585	5.583	5.584	0.02	-0.02
94	4.888	5.288	5.283	5.288	0	-0.09
95	4.398	4.907	4.905	4.907	0	-0.04
96	3.792	4.428	4.430	4.427	0.02	0.07
97	3.060	3.836	3.841	3.834	0.05	0.18
98	2.189	3.121	3.131	3.126	-0.16	0.16
99	1.171	2.271	2.337	2.337	-2.82	0

Table 2.4: Broadie et al. (1997) versus the adjusted transition probability method for a down-and-out call bermuda-style option.

	Barrier	Corrected Continuous	Adjusted Tree	True	Cor. Cont. Rel. Err. (%)	Adj. Tri. Rel. Err. (%)
m=125	85	6.327	6.326	6.326	0.02	0.01
	87	6.293	6.292	6.292	0.02	0.01
	89	6.210	6.210	6.210	0.00	0.00
	91	6.033	6.031	6.032	0.02	-0.01
	93	5.688	5.686	5.688	0.00	-0.03
	95	5.084	5.080	5.081	0.06	-0.02
	97	4.113	4.119	4.116	-0.07	0.09
	99	2.673	2.813	2.813	-4.98	0.00
m=25	85	6.337	6.336	6.337	0.00	0.00
	87	6.323	6.321	6.321	0.03	0.01
	89	6.284	6.280	6.281	0.05	0.00
	91	6.194	6.187	6.187	0.11	0.00
	93	6.004	5.999	6.000	0.07	-0.01
	95	5.646	5.670	5.671	-0.44	0.00
	97	5.028	5.167	5.167	-2.69	0.01
	99	4.050	4.489	4.489	-9.78	0.01

Table 2.5: The option parameter are $S_0 = 100, K = 100, \sigma = 30\%, r = 10\%$ and $T = 0.2$.

Chapter 3

Valuing Modularity as a Real Option

3.1 Introduction

The concept of modularity in design was rigorously introduced in business economics by Baldwin and Clark (2000).[1] They propose a quantitative model to describe the economic forces that push a design towards modularization and the consequences of modularity on the business environment.

Value creation is the goal of the modularization process and real options theory offers a natural framework to evaluate a modular design.[2] Baldwin and Clark (2000) pointed out six operators describing the structure of a modular system, or alternatively its evolution from a non–modular (or interconnected) design to a modular design: *splitting*, *substitution*, *augmenting*, *excluding*, *inversion*, and *porting*. These operators can be thought of as *options* in the designer's palette and Baldwin and Clark propose to link the six operators to real options theory.

The idea of modularity has been known among real options professionals for a long time but it has not attracted enough attention.[3] On the other hand, Baldwin and Clark's work has been widely discussed in information systems and product design literatures, but only in a very qualitative way.[4] So, there is a need to bridge modularity and real options theory and practice, and at the same time, to bring the real options approach closer to system designers. In this article we propose a way to quantify the value contribution of the modularization process and of modules themselves, so to have a practically useful method to apply real options to design decisions.

We extend Baldwin and Clark's model to a stochastic dynamic framework by allowing the state variables of the modularization decisions to follow realistic stochastic processes and by accounting for the dynamic nature of the decision process. We can think of a modular operator as an (generally compound) option to improve the value of a design. Hence, we introduce an

---

[1] Baldwin and Clark (2000) describe three types of modularity: *modularity in design*, *modularity in production* and *modularity in use*. The first refers to the creation of a modular system, the second is related to the simplification of the production process (i.e. dividing complex production tasks into smaller processes); the third concerns the possibility for the consumer to arrange elements in order to obtain a design configuration that reflects his needs. In this work we focus on the modularization of a design and on the related modularization of the production process.

[2] See Dixit and Pindyck (1994) or Trigeorgis (1996) as general references on real options.

[3] Exceptions include Bonaccorsi and Rossetto (1998), Gollier et al. (2005) and Rodrigues and Armada (2007).

[4] See for example Sullivan et al. (2001) and Cai et al. (2007)

approach for valuing the modular operators while accounting for their interactions.[5] Since the number of state variables and of operators simultaneously involved can be significant, we propose a numerical implementation of our valuation approach based on Monte Carlo methods, and in particular on the versatile Least–Squares Monte Carlo (LSM) method by Longstaff and Schwartz (2001).

The paper is organized as follows. In Section 3.2, we describe the basic aspects of modularity as introduced by Baldwin and Clark (2000) and the six main modular operators. We also describe the evolution of a non–modular design into a modular system, by means of the six operators. In Section 3.3 we describe how a non–modular system can be valued. This will permit us to address also some issues related to financial valuation of the individual modules. The non–modular design is the status quo and the benchmark for modularization decisions. In Section 3.4, we provide an approach for valuing the six modular operators. Finally, in Section 3.5, we describe the numerical implementation based on LSM method and offer some numerical results to test the accuracy of the numerical implementation and to show how our approach can be used in realistic contexts.

3.2 Modular designs

A *design* is a detailed description of a product. It is completely determined by a number of parameters and their interconnections. These parameters are related to one another if there is a physical or a logical connection or dependence among them. A *module* is defined by a cluster of strongly interconnected parameters which are almost independent from the parameters of other modules. A *modular design* is a hierarchical set of modules tied through specific *design rules*, which are imperative principles of composition that each module must respect to maintain the compatibility with the other modules and the entire project.[6] The hierarchical structure, which is the framework for individual modules, assigns them different structural functions according to their position. *Hierarchical* modules are placed at the highest level and pose a set of design rules, or *visible information*, for the *hidden* modules, which are dependent and connected modules placed at a lower hierarchical level (see Figure 3.1).

[Figure 3.1 about here]

[5] Along the same line, although for a different purpose, Gamba (2008) provides a way to decompose complex capital budgeting problems with multiple options into a set of simple options.

[6] See for example Steward (1981a), Steward (1981b), Eppinger (1991), Eppinger et al. (1994), Baldwin and Clark (2004).

The modularization process of a design implies many consequences both on the design and the managerial side. As for the design, modularity creates development options: each module gains functional independence within the overall design rules, to which it must adhere. From a managerial viewpoint, the biggest effect of modularization is decentralization: each module will be designed, made and eventually implemented by a specific unit. Hence, modularity improves specialization in the design process. Actually, each module may evolve freely (independently of the other modules) within given design rules and each individual unit can work at its module with no worry of damaging the whole project. Generally speaking, modularization permits to manage complexity because it splits a system into a set of independent elements of smaller size and the design rules tie the modules up into a hierarchical structure.

Baldwin and Clark (2000) explain the dynamic of a modular design through a set of operators, which are also standard design structures. A possibly iterative and simultaneous application of these operators permits to obtain any modular structure from a non–modular, or *interconnected*, one. They are denominated: *splitting*, *substitution*, *augmenting*, *excluding*, *inversion*, *porting*.[7]

The splitting operator is at the core of the modularization process because it permits to generate a set of independent modules from an interconnected design/module. The substitution operator allows to change an existing module (or an interconnected design) with a new one. These two operators can be applied both to modular and non–modular designs. The remaining operators can only be applied to a modular system. The augmenting operator either creates a new hierarchical level or increments an existing layer of modules. By excluding we create a minimal system that can be incremented later on. The inversion operator creates a new source of visible information (design rules) isolating the common features embedded in different modules. Finally, the porting operator allows to make a module component compatible with other designs. Once a modular system is obtained, it can be improved upon using the same six operators. Moreover each operator can be applied locally to the system without interfering with the rest of the structure.

An important consequence of modularity is that, if we have a consistent valuation approach for each of the above operators, we have also a valuation approach for the modularization process and for modular structures. Since the application of these operators is meant to create value, the natural valuation approach is contingent claim analysis applied to discretionary

[7] As acknowledged by Baldwin and Clark (2000), although this set of operators is not exhaustive, it is the smallest one that permits to obtain a modular system from a non–modular design.

investment decisions, that is real options theory. This is what we describe in the next sections.

3.3 Financial valuation of a design

Before describing how to value the modular operators, we discuss the issues related to the valuation of design decisions and in particular, those regarding the primitives of financial valuation of an interconnected design.

A system is interconnected when the parameters describing it are (or seem to be) strongly linked because the designer has a limited knowledge of the relations among them. Hence, the interconnected structure can be thought of as a single module project in which it is difficult to change even a single parameter without affecting all the others. Therefore, absent an analysis of the potential modular structure, an interconnected project can be only improved upon as a whole. Such a redesign is worth doing if it produces a positive net value (namely, the total new value less the research and implementation costs) greater than the value of the initial (*status quo* or existing) design.

We assume a given interconnected design and we model the variation of its value both as a consequence of a change of market conditions and of research and development activity.[8] For simplicity, we assume also that there is no time-to-build the new design, so that the research activity and the creation of the new structure are simultaneous.[9]

Let W_t denote the gross value, at time t, of the future cash flows from the current (interconnected) design assuming no change of configuration of the system. W_t may be calculated using standard capital budgeting techniques. To clarify, the designer may be able to determine W_t from the forecast of revenues, production costs, and any other drivers of the cash flows of the current system. This permits to determine the sequence of expected future cash flows and from this, using a suited cost of capital incorporating the market price of risk, W_t is computed. The parameters of the stochastic process of W_t, and in particular the volatility necessary to value the real options, are derived (possibly using Monte Carlo simulation) from the processes of the value drivers from which W_t is computed.[10] For definiteness we make

[8] We will consider economic and technical uncertainty altogether even if we account only for market uncertainty in determining risk premia.

[9] This is equivalent to the case the design decision is not changed during the time needed to build the new design. We will make the same assumption also when valuing the modular operators.

[10] Kogut and Kulatilaka (2001) discuss the issues related to the use of observed assets/factors prices to determine the underlying asset for real options valuation. In particular, even if we can decompose the gross

the standard assumptions on real options valuation as described by Copeland et al. (2005).[11] According to Samuelson (1965, 1973), since W_t is a present value of future cash flows, it behaves like the price of a traded security and so its process is driven by a Brownian motion, no matter what the actual stochastic processes of the drivers underlying W_t are.

W_t is the first state variable underlying the decision to improve the current interconnected design. The second state variable is the present value of expected future cash flows under the new design and is driven by a different (possibly correlated) Brownian motion, denoted W_t^*, whose parameters are derived using the same approach we discussed above for W_t. To summarize, both W_t and W_t^* are Markov stochastic processes driven both by market (systematic) and technical (non-systematic) uncertainty. Although the actual drivers of the decision are W_t and W_t^*, sometimes we will assume $V_t = W_t^* - W_t$, the incremental value of the design, as the state variable of the valuation problem. This is done with the sole purpose of keeping the notation compact.

Given our previous assumptions that the risk of W_t is spanned, a risk–neutral probability or equivalent martingale measure (EMM) is determined. With reference to the standard valuation approach described above, such that W_t is the expected discounted value of future cash flows, the risk premium needed to change the probability from the historical measure to the EMM is the one used to determine the cost of capital.[12]

---

value into a set of spanning factors that are priced in financial markets, it may not be so obvious to determine the drift of W_t, due to implicit convenience yields and costs of carry that are specific of the real factors, but not of W_t. An alternative approach may be to consider W_t as an unobservable variable underlying the decisions made by firms whose business risk is comparable to the current system's one. In this case, to estimate the parameters characterizing the stochastic process of W_t we must assume that we have a publicly traded stock whose dynamic depends on the same factors affecting W_t. Since the equity is a derivative security of W_t, we can use structural estimation techniques to determine the parameters of the model. An example of this approach, although applied to a different type of real option, is proposed by Gamba and Tesser (2009). However, also in this case we may need to adjust the drift of the stochastic process for whatever convenience yields and carrying costs are incurred by the owner of the "comparable" asset, but not by the holder of the real option we are valuing.

[11] In Copeland et al. (2005), Chapter 9C, the three basic assumptions are: (*i*) the "marketed asset disclaimer" for W_t; (*ii*) absence of arbitrage opportunities in financial markets; (*iii*) the present value of future cash flows, W_t, fluctuates randomly as a Brownian motion.

[12] Gamba et al. (2008) provide a general justification for risk–neutral valuation in capital budgeting in case the firm is levered (and there are personal and corporate taxes). With reference to the alternative approach mentioned in Footnote 10, when W_t is unobservable and is estimated using a structural approach, it is estimated under the EMM. So there is no need of a risk–adjustment in this case.

If we consider a given maturity T and assume that the option to redesign the interconnected structure can be exercised at any time $t \leq T$, the value of this option is

$$F(t, V_t) = \sup_{\tau \in H(t,T)} \left\{ \mathbb{E}_t \left[e^{-r(\tau - t)} \Pi(\tau, V_\tau) \right] \right\}, \tag{3.1}$$

where $\Pi(t, V_t) = \max\{V_t - C, 0\}$ is the net benefit from the redesign, C is the research and implementation costs of the new design, $H(t, T)$ is the set of stopping times in $[t, T]$ and $\mathbb{E}_t[\cdot]$ is the expectation, under the EMM, conditional on the information available at time t. Interpreting equation (3.1), the system is redesigned when $W_t^* > W_t + C$, so that W_t can be seen as an opportunity cost of changing the structure.[13]

3.4 Valuing modular designs

In this section we describe a valuation approach for modularity in design. Since any modular design can be obtained using the operators introduced by Baldwin and Clark (2000), our goal is to propose a suited valuation approach for each operator allowing to capture the interactions among them.[14]

3.4.1 Splitting

Under the condition that some parameters defining a module/system are independent, the module/system can be split into two or more modules and suitable design rules are provided. This operator has a dual function. When applied to an interconnected system, it produces a modular design. In case of a modular design, in which each individual module is independent of the others, this operator splits an existing module into two or more modules. For definiteness, we will assume that the modularization process takes place from an interconnected system. In order to dictate the entire set of design rules, the designer must know all the dependencies among the parameters. Such a knowledge is obtained as a result of a research effort.

[Figure 3.2 about here]

[13] Since the system can be changed at any time before T, a second opportunity cost is related to the value of the option to defer.

[14] Option interaction effects have been studied by Trigeorgis (1993).

At this stage of our analysis, we can distinguish between at least two different types of splitting.[15] The first applies when the initial design is completely interconnected and its independent components are difficult to isolate and be turned into modules. In this case a bigger effort is required to identify all the links among the parameters and the system can be split when all the modules are ready. The second case refers to a design in which the structure is interconnected but the links among the parameters are clearly understood by the designer or, as referred to by Baldwin and Clark (2000), the design has been already *rationalized*. In this case, the value of the current design can be used as a benchmark for the decision on the new versions of the module. Here we focus on the first specification, postponing the latter until Section 3.4.3.

An example of splitting can be found in organizational theory. A bank is considering to specialize its business activity. One way to achieve this result is to split its business, currently interconnected as a single module, into a set of main functions (private investments, retail, small businesses, large businesses etc...). That requires the choice of the target market and the creation of a set of independent modules/divisions based on the main functions. These modules are linked to a central decisional unit which dictates the global business plan (design rules). The benefit of this structure is that each business unit is free to evolve, within the design rules, independently of what happens to the rest of the system. This can be made formal and effective by creating a pyramidal group, where the parent company (the central decision unit) controls the subsidiaries (divisions).

In general, assume that the designer wants to split the system into J modules (as an example but with no limitation to generality, Figure 3.2 shows a split into two modules), and that the decision to split and the option to implement the new modules have maturity T. When the structure is strongly interconnected, the valuation problem comprises only two state variables: W_t, the gross value of the system before the redesign, and W_t^*, the gross value after the redesign. By splitting, we produce a set of modules that by definition are functionally independent (in the sense that they can evolve freely). Hence, W_t^* is the sum of the values of the individual modules W_t^{*j}, for $j = 1, \cdots, J$. For brevity, we denote $V_t = W_t^* - W_t$ the

[15] Rodrigues and Armada (2007) developed a model for valuing the splitting operator, which accounts for the three typical steps of the modularization process, as pointed out by Baldwin and Clark (2000). The first step is the decision to split the interconnected structure; the second is the research activity, and the third step is implementation. Here we focus on the basic features of each operator. Obviously all the valuation formulae presented below can be extended to consider the above three steps.

incremental value from redesigning of the system as a whole.[16]

The decision is made if the value of the modularized design, minus the cost for implementing new modules and design rules, is greater than the value of the old interconnected project. The value of the splitting operator is

$$F_{\text{spl}}(t, V_t, J) = \sup_{\tau_s \in H(t,T)} \left\{ \mathbb{E}_t \left[e^{-r(\tau_s - t)} \Pi_{\text{spl}}(\tau_s, V_{\tau_s}, J) \right] \right\} \tag{3.2}$$

with payoff function

$$\Pi_{\text{spl}}(t, V_t, J) = \max \left\{ V_t - \sum_{j=1}^{J} C_j - C_s(J), 0 \right\}, \tag{3.3}$$

where C_j is the research and realization cost of the j-th module, and $C_s(J)$ is the cost to determine the design rules of the modularized design, which depends on J, the number of modules in the new design. Usually the splitting costs are positively related to the number of modules involved in the modularization. As the number of modules increases, a larger set of design rules is needed in order to create a coherent architecture and a set of interfaces to make all the modules to work in one system.

Two important remarks are in order. The first is that the splitting operator can be exercised only when the new modular system is ready. If at least one of the new modules has not been implemented yet, the decision to split is postponed. The second is that, when $J = 1$, the interconnected design is simply replaced by a new interconnected system, and consequently $C_s(1) = 0$, and equation (3.2) collapses into (3.1), because no design rules are required.

To map the organizational example into equations (3.2) and (3.3), W_t^{*j} is the gross value of the new j-th unit; W_t is the gross value of the current structure; C_j is the cost to create the new j-th business unit; $C_s(J)$ is the cost to create the organizational structures (e.g., the IT infrastructure) to make the business units to work together.

---

[16] Differently from the static framework based on Normal distributions used by Baldwin and Clark (2000), our distributional assumptions on the values of the modules may not permit to split a module/system while remaining in the same class of distributions. For instance, if we assume that the value of the interconnected system evolves according to a Geometric Brownian Motion (GBM), the values of the constituent modules cannot be a GBM. Importantly, this is not an issue in our valuation approach. Our assumption is weaker, because we require that, for any proposed modular architecture, the requisite W-processes exist – one for each stochastically evolving development process. However, we address the issue of "conservation of variance" described by Baldwin and Clark (2000) in a numerical fashion in Section 3.5.3.

Equation (3.3) is similar to the payoff of a call option on V_t. This is because here we are valuing a version of the splitting operator that is simplified in two fundamental aspects: the current system has not been *rationalized* yet; only one new version for each module is considered. Later on, we will remove these two assumptions, and generalize the structure of the operator.

3.4.2 Substitution

Substitution of a module is aimed at improving the system. The substitution operator can be applied to both interconnected and modular structures. In the first case it creates a complete new (interconnected) project. In the second case it only affects individual modules so that each module can evolve independently of the others (see Figure 3.3). Either case, and following Baldwin and Clark (2000), p. 264, we assume that the improvement is the outcome of K alternative attempts (experiments) in a given time period, whereby only the best outcome is the candidate for replacing the current version of the module (or interconnected design).

To properly evaluate the substitution operator, we have to consider the hierarchical level of the module under consideration. Modules placed at a lower hierarchical level, or *hidden* modules, must respect only the design rules which are given by the preceding (upper) and connected modules. Changes in the internal structure of hidden modules do not influence the other components of the project. Modules placed at the highest level, or *hierarchical* modules, in addition to their specific functions, pose a set of design rules for the dependent and connected modules (see Figure 3.1). When a hierarchical (i.e., upper–level) module is replaced, a cost for defining the new interface for the lower-level linked modules, the so called *visibility cost*, denoted Q, is incurred. On the other hand, when a hidden module is replaced, $Q = 0$. That implies also that we should observe more design activity on lower level than on upper level modules, because their substitution has a limited impact on the design structure and a slower rate of change for modules at a higher level.

[Figure 3.3 about here]

The flexibility to improve an existing module with no need of redesigning the entire structure is perhaps the most important motivation to modularize a system. Usually, when a module is replaced it is often the case that the new version maintains all the features of the previous one, while expanding its functionality. Computers are a classic example of modular structure and the computer design is suited to show how the application of the substitution operator works. The CPU is a hierarchical module: when it is changed, we usually have to

change also the motherboard, which is a lower-connected module. On the other hand, it is possible to improve (applying the substitution operator) the video performance of a computer using a new graphics card, without affecting the global design structure. Hence, the graphic card can be considered a hidden module in the broad computer structure.

Assume that a time horizon T is given for replacing a given module (or an interconnected design), and to start with, say that the number of trials, K, is decided in advance. In our notation, for the selected module, W_t is the gross value before the redesign, and W_t^{*k} is the gross value after the redesign if the outcome of the k-th experiment is implemented, with $k \in \{1, \ldots, K\}$. To keep the notation compact, we denote $V_t^k = W_t^{*k} - W_t$ the incremental value of the module from the k-th attempt. Let $\mathbf{V}_t = \left(V_t^1, \ldots, V_t^K\right)$ be the resulting vector of incremental values for the module. Hence, the value of the substitution operator (applied to the module) is

$$F_{\text{sub}}(t, \mathbf{V}_t, K) = \sup_{(k,\tau)} \left\{ \mathbb{E}_t \left[e^{-r(\tau - t)} \Pi_{\text{sub}}(\tau, V_\tau^k, K) \right] \right\}, \tag{3.4}$$

where (k, τ) is the control, with k denoting the selected trial version, and $\tau \in H(t, T)$ the stopping time for this decision. The payoff is

$$\Pi_{\text{sub}}(t, V_t^k, K) = \max \left\{ V_t^k - Q - \sum_{k=1}^{K} I_k - C_k, 0 \right\}, \tag{3.5}$$

where Q is the visibility cost (which may be zero in case the module at hand is hidden), C_k is the incremental implementation cost of the selected new version of the module and I_k is the cost to run experiments on the k-th version of the module. Notice that the value of the substitution operator reduces to the value of a simple call option when $K = 1$.

If the number of trials, K, can be optimally chosen *ex ante*, the value of the substitution operator is

$$F_{\text{sub}}(t, \mathbf{V}_t) = \max_{K \in \mathbb{N}} \left\{ \sup_{(k,\tau)} \left\{ \mathbb{E}_t \left[e^{-r(\tau - t)} \Pi_{\text{sub}}(\tau, V_\tau^k, K) \right] \right\} \right\}, \tag{3.6}$$

with the same notation used above.

We can map the example of the replacement of the CPU unit in a PC design in the previous equations. Assuming the other modules remain unchanged, the incremental value of the module for each possible candidate of the new CPU, V_t^k for $k = 1, \ldots, K$, is estimated as the value increase of the PC design due to the new CPU. The visibility cost, Q, refers to the need of re-designing the lower connected modules (for example the motherboard) whenever a hierarchical module (as the CPU) is substituted. C_k is the production cost of the selected

version of CPU. Finally, the cost of the k-th trial is the sum of the direct costs to prepare the different versions of the CPU and to test them.

3.4.3 The splitting operator revisited

The above analysis permits to extend the scope of the splitting operator. As we said in Section 3.4.1, we have at least two different types of splitting. Here, in line with Baldwin and Clark (2000), we describe the modularization process for a design that has already been rationalized, in the sense that the potential units (i.e. the future modules) has been already identified in the current system.

Given (by definition) the functional independence of the J modules, each individual module resulting from the split can be valued in isolation. Therefore, the incremental value of a modular project is the sum of the incremental values of its modules. Namely, using the notation introduced before, for a given module j, if W_t^j is the gross value before the redesign, and W_t^{*j} is the gross value after the redesign, we denote $V_t^j = W_t^{*j} - W_t^j$ the incremental value from redesigning, for $j = 1, \cdots, J$, and finally, the total marginal contribution of splitting is captured by $V_t = \sum_{j=1}^{J} V_t^j$, where for some j we can have $W_t^{*j} = W_t^j$, in the sense that module j is not changed. In this setting, there is no point in splitting a rationalized system at a positive cost if the resulting modules are not improved, because the (positive) cost $C_s(J)$ would be paid, but $V_t^j = 0$ for all j. So, to exercise the option to split an improved version must be proposed for at least one module.

The above is true if, as we did for simplicity in Section 3.4.1, we assume that the designer implements one version for each of the required modules. Instead, it is often the case that more than one version is proposed for a new module. In this case, there are two distinct steps in the splitting process: the decision to split the interconnected design, and the selection and implementation of the best version for each module (see Figure 3.4).

[Figure 3.4 about here]

So, the value of the option to improve the modules must be incorporated, along the lines of what we did in Section 3.4.2. The payoff of the splitting operator in (3.3) becomes

$$\Pi_{\text{spl}}(t, \mathbf{V}_t, J) = \max \left\{ \sum_{j=1}^{J} F_{\text{sub}}(t, \mathbf{V}_t^j) - C_s(J), 0 \right\}, \tag{3.7}$$

where F_{sub} is defined in (3.4) (or alternatively in (3.6), if the number of trials can be optimally chosen) and $\mathbf{V}_t = (\mathbf{V}_t^1, \ldots, \mathbf{V}_t^J)$ is such that $\mathbf{V}_t^j$ is a vector, whose components are the

incremental values corresponding to the multiple independent research activities engaged for the j-th module. Interpreting equation (3.7), when the designer decides to split, she dictates the entire set of design rules, at a cost $C_s(J)$. For each module, she has the opportunity to select the best version out of many experiments, but the best version does not need to be ready at the time of the splitting decision. To exercise the option to split, assuming $C_s(J)$ is positive, at least one version must be available for each of the J modules, otherwise $V_t^j = 0$ for all j.

Comparing (3.7) to equation (3.3), F_{sub} incorporates the value of the opportunity to select the best version for each module after the split. Importantly, while in (3.3) the splitting decision is made *ex-post*, when the new version of the modules is valuable enough, in (3.7) the decision is made *ex-ante*, based on the option value of the substitution operator. This can be done because the system has been already rationalized. While the old version of the modules are used, the designer starts a research activity on each unit. If this is successful, she implements the new version of the module. Otherwise, she keeps the old version of the module.

Hence, F_{spl} can be thought of as a complex compound American option on the max of several call options, each of which has the gross value increment for the individual module as underlying asset. I.e., F_{spl} is the value of an option on a portfolio of options.[17]

3.4.4 Augmenting

The augmenting operator improves a design by adding one or more modules (see Figure 3.5). The opportunity to improve a design with no change in the rest of the structure is another important motivation for its modularization. The augmenting and the excluding operator, described later on, are frequently used together. For this reason, an example involving the augmenting operator is presented in Section 3.4.5, together with the excluding operator. However, when we add more modules to the current system with the augmenting operator, we do not change the existing design rules of the structure.[18]

[Figure 3.5 about here]

---

[17] Valuation formulae for compound options have been provided by Geske (1977, 1979) and Carr (1988). Compound options have been extensively used to model real options. Examples are Kemna (1993) and Martzoukos and Trigeorgis (2002).

[18] This is an important difference with respect to the excluding operator, which instead involves the provision of new design rules.

Let assume a finite horizon T for adding a module, and that there is only one version for the $(J+1)$-th module. When information about the new module becomes available, the incremental value of the whole system from the $(J+1)$-th module is $V_t^{J+1} = W_t^{J+1} - W_t^J$, where W_t^J and W_t^{J+1} are the gross values of the design before and after adding the new module, respectively. The value of the augmenting operator for one additional module is

$$F_{\text{aug}}(t, V_t^{J+1}) = \sup_{\tau \in H(t,T)} \left\{ \mathbb{E}_t \left[e^{-r(\tau - t)} \Pi_{\text{aug}}(\tau, V_\tau^{J+1}) \right] \right\}, \tag{3.8}$$

where $\Pi_{\text{aug}}(\tau, V_\tau^{J+1}) = \max\left\{V_\tau^{J+1} - C_{J+1}, 0\right\}$ and C_{J+1} is the research and development cost of the $(J+1)$-th module. This is the payoff of a simple call option on V_t^{J+1} with strike price C_{J+1}. The extension to the case the firm can evaluate several potential candidates for the $(J+1)$-th module using the substitution operator is done using the same logic as in Section 3.4.3.

3.4.5 Excluding

The excluding operator permits to create a minimal design with the opportunity to increase (using the augmenting operator) its size, scope and depth later on, if the initial design is successful. Importantly, the whole structure and the suited design rules to incorporate the additional modules later on must be set since the beginning.

This approach can have both strategic and financial motivations. Strategically, the initial exclusion of a module reduces the impact of potential failure of the whole design. On the other hand, the initial exclusion of a module from the system may allow to fund the subsequent expansion with the cash flows generated by the reduced (but operating) initial design.

Applications of this operator are usually found in the valuation of large and irreversible investments, such as power plants and oil wells. As an example, Gollier et al. (2005) describe how the flexibility provided by the excluding operator can generate value and hence anticipate the optimal timing of the investment decision. They describe the following situation. An electricity company is planning to expand its production capacity by building a new nuclear power plant. It can follows two alternative approaches: in the first, one large production unit is built; the second approach is modular, because it comprises the construction of a series of lower size power production units over time. Assume that the electricity price is the main driver of the decision. When the electricity price is highly volatile, the modular approach allows to reduce risk and to shorten the time of the initial investment. This approach corresponds to applying the exclusion operator to the initial design and then to use the augmenting operator within the design rules set at the beginning. An initial power plant of reduced size

is constructed, with the option to expand (i.e. to augment) its capacity later, should the economic conditions turn favorable. As an example, Figure 3.6 shows a design comprising two modules, which is initially realized with only one module, and the second module is added in a second step.

[Figure 3.6 about here]

To value this operator, let T_α be the time horizon for the decision to dictate the entire design rules and to introduce the reduced design, and T_ω the time horizon to complete the design.[19] Conveniently enough, the problem can be thought of as one of valuing a compound American option. So we begin from the last option (assuming the first has been already exercised) and then work backwards to value the excluding operator. Let $V_t = W_t^* - W_t$ be the incremental value of the additional module to complete the design, where W_t^* and W_t are the gross values of the complete and of the initial design, respectively. The option to add the second module is simply an application of the augmenting operator and its value is

$$F_{\text{aug}}(t, V_t) = \sup_{\tau \in H(t,T_\omega)} \left\{ \mathbb{E}_t \left[e^{-r(\tau - t)} \Pi(\tau, V_\tau) \right] \right\},$$

with $\Pi(\tau, V_\tau) = \max\{V_\tau - C, 0\}$ and C is the realization cost for the additional module. Hence, the value of the excluding operator is

$$F_{\text{excl}}(t, W_t, V_t) = \sup_{\tau \in H(t,T_\alpha)} \left\{ \mathbb{E}_t \left[e^{-r(\tau - t)} \Pi_{\text{excl}}(\tau, W_\tau, V_\tau) \right] \right\}, \tag{3.9}$$

where

$$\Pi_{\text{excl}}(\tau, W_\tau, V_\tau) = \max\{W_t - C_W - C_s + F_{\text{aug}}(\tau, V_\tau), 0\},$$

and C_W is the research and realization cost of the minimal system and C_s is the cost of the design rules for the complete system.

In the nuclear power plant example, C_W and W_t are respectively the cost and the value of the initial reduced size plant. C_s is the cost to set up the entire design rules, that is the cost of providing the complete architecture and interfaces (e.g., the connection with the network) in which the initial power plant and the potential expansion production units have to work.

Finally, to decide if the initial exclusion is worth it, $F_{\text{excl}}(t, W_t, V_t)$ is to be compared to the value of introducing the whole design in one step with no staging, in order to determine the net gain from the application of the excluding operator.

---

[19] For simplicity, but with no restriction, we assume that the initial system consists of one module and the complete design is obtained by adding one module. This can be easily extended to any number of modules. The difference in equation (3.9) is that we would have many compound options.

3.4.6 Inversion

The life of a modular design can be divided into two typical phases: in a first phase, an interconnected design is turned into a modular one by splitting it; in a second phase, the design can be improved upon by further splitting, augmenting, replacing, porting and excluding the existing modules. However, additional changes can be made to increase the value of the system. Among these, there is the possibility to improve the design by grouping similar or common functions that are spread across the structure into a single module. This module is then connected to all the other modules where the common function was present. As Baldwin and Clark (2000), p. 323, pointed out:

> "Immediately after the split, there will be a rush of experimentation, but sooner or later such experimentation displays diminishing returns, and give rise to unmanageable amounts of variety. When the benefit of further experimentation no longer justify the cost, additional design rules are called for. These new design rules are created via the inversion operator."

Typically, this operator involves three steps, as depicted in Figure 3.7 for a paradigmatic case. We have to:

1. find similarities in the modules;
2. split the relevant modules in order to single out the similar components;
3. create a new module from the similar components and place it at a higher hierarchical level (i.e., by *inverting* its ranking in the hierarchy of the original design).

Two are the most important consequences. First, the structure becomes less flexible because we add a new hierarchical level, and generally speaking this reduces the value of the system, because the more levels we have the more rigid is the system. A second effect is that scale economies are obtained in the research and development activity, and this increases the value of the system.

[Figure 3.7 about here]

In what follows, we describe how these two offsetting incentives affect the valuation of the inversion operator. Assuming a broader modular structure, we focus only on the sub-

set of modules involved in the inversion; i.e., those modules sharing the common or similar function.[20]

As an illustrative example, we can think of a merger between two auto manufacturers.[21] Since the two companies belong to the same business area, their internal structures can potentially have some similar functional units or modules (e.g., the administrative department, the research department, or the production line of some common part of the vehicles). In the left part of Figure 3.7, Module 1 and Module 2 represents the production systems of the two companies involved in the merger. They have a common unit and so the owner can apply the inversion operator to benefit from scale economies. First she has to isolate (i.e. split, as in the second step in Figure 3.7) the common components I in each firm, say, two lines that produce the same part for the vehicles of the merged companies, assuming the original brands are maintained. Next, a new module $\hat{I}$ (i.e. a new production line) is designed, which can work with both original systems. Finally, the new module is placed on top of them by inverting its ranking in the hierarchy of the original design, as in the third step of Figure 3.7. The resulting company has only one production module that provides it services to the remaining modules of the two merged companies (i.e. Module 1 and Module 2 specific components).

The value of the inversion operator is

$$F_{\text{inv}}(t, \mathbf{V}_t) = \sup_{\tau \in H(t,T)} \left\{ \mathbb{E}_t \left[e^{-r(\tau - t)} \Pi_{\text{inv}} (\tau, \mathbf{V}_\tau) \right] \right\} \tag{3.10}$$

where

$$\Pi_{\text{inv}}(t, \mathbf{V}_t) = \max \left\{ V_t^{\widehat{I}} - C_{\widehat{I}} - Q - \sum_{j=1}^{J} F_{\text{sub}}(t, V_t^j), 0 \right\}. \tag{3.11}$$

For definiteness, we assume that the common component is in J modules of the current design, that no other change takes place for them, and that the function of the inverted module is unchanged. In equation (3.11), $V_t^{\hat{I}}$ is the incremental value of the new upper-level module and is estimated as the difference between the value of the new design, $W_t^{\hat{I}}$, and the value of the original structure, $W_t = \sum_{j=1}^{J} W_t^j$, where W_t^j is the value of the old version of module j, which includes the common component. In (3.11), $C_{\hat{I}}$ is the research and realization cost of the new module; $F_{\text{sub}}(t, V_t^j)$ is the value of the option to pursue an independent improvement

[20] The possibility to focus on the local effects (instead of taking care of the broader effects) of a change in the structure is actually one of the benefits of considering modular designs.

[21] See for example Rudholm (2007) for an analysis of the motivations behind the mergers between Volkswagen with SEAT and Skoda.

for the j-th module, from (3.4) or (3.6). Hence, $-\sum_{j=1}^{J} F_{\text{sub}}(t, V_t^j)$ is the opportunity cost to reduce the flexibility of the design. Finally, Q is the visibility cost that must be paid because, when the inversion operator is applied, the interfaces of the lower connected modules with the new upper-level module, $\hat{I}$, may need to be changed.

In the example we are considering, $C_{\hat{I}}$ is the cost of creating a new centralized production unit for the common components, $-\sum_{j=1}^{J} F_{\text{sub}}(t, V_t^j)$ is the overall opportunity cost due to the restricted flexibility of the new firm to design (and produce) different types of vehicles under the different brands. Lastly, the visibility cost Q is the cost of the communication procedure between the common production line and the production units of the merged companies.

Equation (3.11) summarizes the basic features of the inversion operator. Importantly, if the new inverted module has a value $V_t^{\widehat{I}}$ lower than the other costs, no inversion takes place. Not differently from what we did with the other operators, also this one can be used jointly with the other modular operators described above. For example, when the designer chooses to invert a module in an already rationalized design structure, she might attempt to improve upon that module. This would change the payoff of the inversion operator in a way that should be clear at this point: we just need to put the value of the substitution operator for that module, $F_{\text{sub}}(\mathbf{V}_t^{\widehat{I}}, t)$ from (3.4) or (3.6), in place of $V_t^{\widehat{I}} - C_{\widehat{I}} - Q$ in equation (3.11). The consequence is that, as we noted in Section 3.4.3, the designer decides to invert *ex-ante*, before knowing the actual outcome of the substitution process.

3.4.7 Porting

The porting operator is applied when a module has functions that can be used also in a different structure. That is, the module has an independent set of parameters that can work well also out of the current design rules, and consequently can be connected with other designs. As Baldwin and Clark (2000), p. 343, says:

> "Porting is like inversion in that it promotes a common solution in a wide range of contexts – this reduces the costs of design (the design does not have to be redesigned from scratch in every system), but may diminish gains from subsequent experimentation. Portable modules and subsystems also are not 'trapped' by the design rules of a particular system; in a sense they are 'free to roam' from system to system."

The steps for porting modules in other structures are four:

1. a potential portable unit is found;

2. we split the initial module into two sub-modules, one independent from the design and the other deeply connected to the current design;

3. the portable sub-module is isolated from the structure. As a result, new design rules are created;

4. we link the portable module with all the other designs where it can be used. This is done through the creation of specific translator modules to connect each structure with the portable unit.

[Figure 3.8 about here]

The use of this operator may be described by the following case.[22] iPod, the now famous Apple Mp3 player, was compatible only with Macintosh Operating Systems (OS) when it was first released (October 2001). So, at that stage the iPod was a lower–level module in the broader Apple system. A possible strategy to enlarge the client basis was to make the iPod compatible with other OS's (and specifically, with the Microsoft Windows family of OS's) by *porting* the module outside the current design structure to an upper hierarchical level, making it a new source of visible information. This was exactly what Apple did. This came at a cost: to make iPod to work on a PC, Apple had to develop (2003) a Windows software (Musicmatch Jukebox). After few months, Apple replaced Musicmatch with a PC version of iTunes, the software that allows (also) to manage the iPod.[23] The Windows version of iTunes (and Musicmatch) is an example of translator module, which is a design structure allowing the ported module to work inside others designs.

In general, two offsetting incentives motivate the porting process. Scale economies (i.e., fixed cost savings) are the major incentives in favor of the creation of a ported module. However, the resulting system becomes less flexible because a new source of visible information (i.e., a new set of design rules) is imposed. To value the porting operator we will refer to the case in Figure 3.8, with no restriction of generality. As we did with the inversion operator, we assume that the initial design has already been split. Moreover, we assume that the portable module, $\hat{P}$, can potentially work inside M different systems. The value of the operator depends on the ownership structure of the systems involved in the porting process.

---

[22] For a complete discussion see Ina Fried, "Will iTunes Make Apple Shine?" CNET News.com, October 16, 2003.

[23] We are not considering the additional value of iTunes coming from the fact that Windows users can actually buy music and other contents from the on-line store.

Let assume first that all these systems belong to the same owner. In this case, she has to realize the translator module for each system where the module is ported. On the other hand, she can save the cost of redesigning M different versions of the same module. Therefore, the value of the porting operator, considering the incremental value of all the systems where the portable unit can be used, is:

$$F_{\text{port}}(t, \mathbf{V}_t) = \sup_{\tau_p \in H(t,T)} \left\{ \mathbb{E}_t \left[e^{-r(\tau_p - t)} \Pi_{\text{port}}(\tau_p, \mathbf{V}_{\tau_p}) \right] \right\}$$

where

$$\Pi_{\text{port}}(t, \mathbf{V}_t) = \max \left\{ V_t^{\widehat{P}} - C_{\widehat{P}} - Q - \sum_{i=1}^{M} R_i - \sum_{i=1}^{M} F_{\text{sub}}(t, V_t^i), 0 \right\}. \tag{3.12}$$

As we did with the inversion operator, here we are assuming that the function of the ported module remains unchanged. In equation (3.12), $V_t^{\hat{P}}$ is the incremental value due to the realization of the new ported module and it is the difference between the value of the new design, $W_t^{\hat{P}}$, and the initial value of the modules involved in the porting process, $W_t = \sum_{i=1}^{M} W_t^i$. $C_{\widehat{P}}$ is the related research and realization cost. $-\sum_{i=1}^{M} F_{\text{sub}}(t, V_t^i)$ is the opportunity cost due to the decision to avoid to pursue independent improvement for each system, reducing the flexibility of the overall modular structure. $\sum_{i=1}^{M} R_i$ is the cost related to the creation of the translator modules for the external systems. Finally, Q is the visibility cost that arises from re-designing the interface of the internal translator module.

In case the target systems, where the module is ported, are not controlled by the owner of the original system, the above general formula in (3.12) is simplified because $F_{\text{sub}}(t, V_t^i) = 0$ for $i = 1, \ldots, M$, as the opportunity costs (if any) are attributable to other agents. Instead, the cost to create the translator module, $\sum_{i=1}^{M} R_i$, may or may not be paid by the owner of the ported module.

Getting back to the iPod case, $V_t^{\hat{P}}$ is the present value of the cash flows generated by selling the iPod to Windows users. $C_{\widehat{P}}$ is the realization cost of the new ported module, while Q represents the cost of finding a configuration of the ported module that is independent of the involved systems. Since Apple does not own the target systems (Windows), $F_{\text{sub}}(t, V_t^i) = 0$, as noted above. Finally, R_i is the cost to develop the Windows versions of iTunes. In this case, since Windows can accept modules developed by third parties,[24] the designer of the ported module must realize also the translator modules. In other cases, the cost of the translator

[24]The software industry offers many examples of open systems; i.e., systems that can accept modules developed by external entities. This fact usually simplifies the porting process.

modules ($\sum_{i=1}^{M} R_t^i$) can be drop from (3.12).[25] In this case, the *leverage* offered by the porting operator becomes significant, or as Baldwin and Clark (2000), p. 344, say

> "[...] the value of this option goes up dramatically if the system to be ported and the host systems are owned by different enterprises."

The above specification of the porting operator can be extended as in the previous sections, so that this operator can interact with the other ones. For example, in case the initial design has already been rationalized, the substitution operator with many potential candidates can be applied to the portable module by putting $F_{\text{sub}}(\mathbf{V}_t^{\widehat{P}})$ in place of $V_t^{\widehat{P}} - C_{\widehat{P}} - Q$ in (3.12). Again, in this situation, the porting operator is exercised ex-ante because it gives the designer the option to start a parallel research activity on the portable module while the old module can be used until it is replaced by the new version.

3.5 Numerical valuation

3.5.1 The LSM method

Given the optimal stopping nature of the valuation problems we described above, and the complexity that a modular design can have, the valuation of the modular operators must rely on numerical methods.

In all the valuation problems of Section 3.4, we need to solve a stochastic optimal control problem of the form

$$F(t, \mathbf{V}_t) = \sup_{(\tau,u)} \left\{ \mathbb{E}_t \left[e^{-r(\tau-t)} \Pi(\tau, \mathbf{V}_\tau, u) \right] \right\}, \tag{3.13}$$

where $\mathbf{V}_t$ is of dimension n, $\Pi(t, \mathbf{V}_t)$ is the payoff from immediate exercise of the operator, $\tau \in H(t,T)$, with $H(t,T)$ denoting the set of stopping times in $[t,T]$, and $u \in U(\tau, \mathbf{V}_\tau)$ is a given control to be chosen at τ.

This kind of problems are solved using dynamic programming, starting from the final date T and then working backwards to the current date by solving the associated Bellman equation at all possible decision dates. To estimate F, we divide the time interval $[t,T]$ in a given number of steps of equal length dt. The related Bellman equation is

$$F(t, \mathbf{V}_t) = \max \left\{ \max_{u \in U(t,\mathbf{V}_t)} \Pi(t, \mathbf{V}_t, u), \Phi(t, \mathbf{V}_t) \right\},$$

[25] A notable example is offered in the same industry: on most recent cars it is possible to connect the iPod to the car computer. In this case, the auto manufacturer (as opposed to Apple) creates the connections (translator module).

where $\Phi(t, V_t)$ is the value of continuation, which is equal to the conditional expectation (under the EMM) of the value of the operator in the subsequent step, discounted to time t at the risk-free rate:

$$\Phi(t, \mathbf{V}_t) = \mathbb{E}_t \left[e^{-rdt} F(t + dt, \mathbf{V}_{t+dt}) \right].$$

The continuation value must be computed using some numerical methods. We will use the Least–Squares Monte Carlo (LSM) method proposed by Longstaff and Schwartz (2001) because it is a versatile technique that allows to manage multivariate state variables, ameliorating (although not avoiding) the curse of dimensionality affecting other numerical procedures, like the lattice methods.

The procedure is based on Monte Carlo simulation to generate the paths of the relevant state variables and on estimating, at all possible decision dates, the continuation value by a least–squares regression of the discounted value of the payoff at future dates over a linear combination of a set of basis functions of the simulated state variables at time t:

$$\Phi(t, \mathbf{V}_t) \approx \sum_{\ell=1}^{L} \hat{\beta}_\ell \varphi_\ell(\mathbf{V}_t),$$

where L is the number of basis functions used in the regression, $\hat{\beta}_\ell$ is the estimated coefficient relative to the ℓ-th function and φ_ℓ is a specific function of the state variables.[26] To mitigate the curse of dimensionality we use a complete set of polynomials of total degree p in n variables to define the functions $\varphi_\ell(\cdot)$, so that the number of coefficients β_ℓ we need to estimate grows polynomially with the state space dimension, n.[27]

3.5.2 Testing the LSM method for individual modular operators

In our numerical experiments, we use alternatively power, Chebyshev, Hermite and Legendre polynomials up to degree $p = 3$, with no substantial difference in the numerical results obtained with each choice. Moreover, to keep dimensionality low and to be able to assess the reliability of the algorithm (i.e., to assess the converge to numerical results obtained using binomial lattice), in this first set of experiments we will adopt the simplifying assumption

[26] Typical choices of the basis functions are: power, Legendre, Chebyshev, Laguerre, Hermite. See Moreno and Navas (2003) for a study on the effect of using different types of basis functions. See also Stentoft (2004) for a study of the convergence properties of the LSM method.

[27] For example, if the dimension of the state space is n, the tensor product of total degree equal to 3 is made of 3^n terms while the corresponding complete polynomial comprises only $1 + n + n(n + 1)/2$ terms. For a reference, see Judd (1998).

that the state variables of the valuation problem are the incremental values, V_t, instead of the actual state variables W_t and W_t^*, according to the notation we introduced in Section 3.2.

The benchmark estimates of the values are obtained using the Cox et al. (1979a) approximation for the valuation of the augmenting operator (a one-dimensional problem), while for the remaining operators (essentially multi-variate problems) we adopt the Adjusted Generalized Log-Transformed (AGLT) binomial method by Gamba and Trigeorgis (2007).[28] This forces us to assume that the state variable V_t (instead of W_t^* and W_t) follows correlated Geometric Brownian Motion (GBM)[29]

$$dV_t^j = \alpha_j V_t^j dt + \sigma_j V_t^j dB_t^j,$$

where α_j is the risk-neutral drift of the process, σ_j is the standard deviation and dB_t^j is the increment of a Brownian motion under the EMM. Later on we will propose other applications based on more realistic assumptions on the structure of the problem.

Monte Carlo sample paths are generated using an Euler discretization of the stochastic equation defining the GBM.[30] These experiments are meant to show the consistency of the proposed numerical methods with an accurate (but slow) valuation method based on binomial lattices in conjunction with a two point Richardson extrapolation,[31] an accurate although slow method, as pointed out by Broadie and Detemple (1996).[32]

We analyze five paradigmatic problems related to *splitting*, *substitution*, *augmenting*, *exclusion*, and *porting*. In what follows, we will skip the inversion operator because the mathematical structure of the problem is very similar to the one of the porting operator (and

---

[28] The latter methods has been proved to be more efficient than other lattice methods in a multi-dimensional setting.

[29] We are aware that this is not a realistic assumption, because if W_t^* and W_t behave like an asset and so follow a GBM, as clarified in Section 3.3, $V_t = W_t^* - W_t$ cannot be a GBM. Moreover, V_t cannot be negative (or $W_t^* > W_t$ is always true) under the current assumption. We accept this simplification at this stage because it is just a way to have a feasible and reliable benchmark using a binomial lattice method for a valuation based on Monte Carlo simulation in a multi-dimensional setting.

[30] See Kloeden and Platen (1999) or Glasserman (2004) as excellent references on Monte Carlo methods in finance.

[31] Geske and Johnson (1984) and then also Boyle et al. (1989) and Breen (1991) suggested to use Richardson extrapolation as a practical method to obtain accurate approximations of exact values while saving on computing time.

[32] All the routines we used are available on request.

because it will be the focus of a more realistic problem, later on). Table 3.1 collects the parameters we used for this set of experiments.

[Table 3.1 about here]

The splitting operator allows to create a project made of two modules, like the one in Figure 3.2. There are two state variables with initial values $\mathbf{V}_0 = (7, 9)$, drifts $\alpha = (0.04, 0.06)$, volatilities $\sigma = (0.25, 0.17)$ and correlation $\rho = 0.3$ that must be realized subject to their research and realization cost, $\mathbf{C} = (5, 8)$, and to the cost of splitting $C_s = 3$.

The substitution operator gives the owner the option to select the best out of two competing versions of the module before $T = 1$, as depicted in Figure 3.3. Therefore, there are two state variables with current values $\mathbf{V}_0 = (6, 10)$, and the parameters of their processes are $\alpha = (0.04, 0.02)$, $\sigma = (0.25, 0.17)$, and correlation $\rho = 0.3$. The (incremental) realization costs are $\mathbf{C} = (5, 8)$ and we assume zero research and development and visibility costs (i.e. $I_{j,k} = 0$ for all $k = 1, ..., K$ and $Q_j = 0$ in the equation of substitution payoff).

For the augmenting operator we consider a situation in which the designer has the option to expand the existing system by adding a new module, as in Figure 3.5. The parameters of the incremental value process are: initial value $V_0 = 10$, drift $\alpha = 0.03$ and volatility $\sigma = 0.15$. The realization cost is equal to 8.

As for the excluding operator, we examine the opportunity to launch a minimal system (initial value $W_0 = 7$, drift $\alpha = 0.04$, volatility $\sigma = 0.25$, research and realization cost $C_W = 5$ and maturity $T_\alpha = 0.5$) with the subsequent option to expand it adding a new module (initial value $V_0 = 9$, drift $\alpha = 0.06$, volatility $\sigma = 0.17$ research and realization cost $C = 8$ and maturity $T_\omega = 1$) to the design, as described in Figure 3.6. For simplicity, but not reducing generality, we assume zero cost for dictating the design rules. So, the problem has a two-dimensional state space.

Finally, for the porting operator, we consider the case in which we can realize a portable module that fits well within three external different systems. This entails also the realization of the translator modules. Therefore the valuation problem has four state variables: one for the portable module (initial value $V_0^{\widehat{P}} = 13.3$, drift $\alpha = 0.07$, volatility $\sigma = 0.1$ and realization cost $C^{\widehat{P}} = 9$) and three to model the dynamics of the potential values from the research activity in the three different systems (initial values $\mathbf{V}_0 = (10, 4, 4)$, drifts $\alpha = (0.04, 0.04, 0.02)$, volatilities $\sigma = (0.03, 0.04, 0.02)$ and realization costs $\mathbf{C} = (11, 5, 4)$). The realization of the translator modules implies costs $\mathbf{R} = (1, 1, 1)$ and the visibility cost is set $Q = 1$.

[Table 3.2 about here]

Table 3.2 presents the estimates obtained from the Least–Square Monte Carlo method and the accurate values from a lattice method. In particular, "*LSM*" is the sample mean value obtained from 30 experiments, for the splitting, substitution, augmenting and excluding operators; from 10 experiments for the splitting operator; from 20 experiments for porting. "*s.d.*" is the corresponding standard deviation of the sample mean which is between 0.002 and 0.013. Each of the Monte Carlo experiments is based on 8000 paths and 100 steps for substitution, 8000 paths and 120 steps for porting, 10000 and 100 for splitting, 15000 and 100 for augmenting, 10000 and 400 for excluding. The simulated paths are obtained using the Euler discretization of the continuous time dynamics and the antithetic variates technique. These results are compared to the accurate values obtained using a lattice method together with a two point Richardson extrapolation. For the augmenting operator we use the approximation by Cox et al. (1979a) with $(100, 200)$ steps respectively; for the remaining operators we use the AGLT approximation by Gamba and Trigeorgis (2007) with $(75, 150)$ steps for the substitution, excluding and splitting operators, and with $(10, 20)$ steps for porting operators.

From inspection of Table 3.2, we can see that the employed numerical method is fairly accurate for the purposes of capital budgeting. While these examples are basic to allow a comparison to numerical solution based on binomial lattices, the Monte Carlo simulation method permits easily to generalize on the number of modules and state variables involved in the decision process. This would be unfeasible using a binomial lattice technique. In this respect, the algorithm we propose is more general and flexible. In the next section we apply the approach to a more realistic valuation problem.

3.5.3 Application to a complex design

As an illustrative example of valuation of a modular project involving several operators, consider the case described in Section 3.4.6.

Two auto manufacturers (A and B), who produce similar types of vehicle, create a merged company. While keeping the original brands alive, to improve efficiency, they consider centralizing the design and production of some components, namely the car frame. This requires two steps (see also Figure 3.7). First, to split the current production processes in order to isolate the production line of the car frame and make it independent of the rest of the productions. Second, to apply the inverting operator to centralize the production of the common component.

In order to apply the splitting operator (see equation (3.2)) we estimate the value of

the initial system $W_t = W_t^A + W_t^B$, where W_t^A and W_t^B are respectively the value of the current productions of the two brands considered in isolation. After the splitting, the system will comprise four modules: the specific ones (which produce all the parts that are brand–specific), whose values are $W_t^{A,s}$, $W_t^{B,s}$, respectively, and the car frame production modules, whose values are $W_t^{A,c}$ and $W_t^{B,c}$. We denote $W_t^* = W_t^{A,c} + W_t^{A,s} + W_t^{B,c} + W_t^{B,s}$ the value of the system after the split.

The application of the inversion operator requires the creation of the new centralized module, I. Our hypothesis is that the original design is rationalized so that, as pointed out in Section 3.4.3, the inversion can take place beginning the experimentation activity on the inverted module, while using the old version. We assume that three experiments are conducted on the new version, $\widehat{I}$, of the inverted module, and the best outcome is chosen at the end of the test. The value of the version from the k-th experiment is $W_t^{I,k}$ while the associated cost is $K_{\widehat{I},k} = C_k + \sum_{\kappa=1}^{3} I_\kappa$, for $k = 1, 2, 3$, using the notation in equation (3.5). Q is the visibility costs to make the specific components of each vehicles compatible with the new common car frame. Therefore, the value of the inverted module is $F_{\text{sub}}(\mathbf{V}_t^I)$ from equation (3.4), where $\mathbf{V}_t^I$ is a vector with components, $V_t^{I,k} = W_t^{I,k} - (W_t^{A,c} + W_t^{B,c})$, for $k = 1, 2, 3$. Hopefully, but not necessarily, $V_t^{I,k} > 0$ as a result of the scale economies generated by the inversion.[33]

The inverting decision should be based not only on the direct costs, but also on the opportunity costs the designer has by avoiding an independent upgrade of the individual common modules. Assuming (for simplicity) that just one new version is considered, we denote $\widehat{W}_t^{A,c}$ and $\widehat{W}_t^{B,c}$ the values of the new versions for the two modules. We denote $C_{A,c}$ and $C_{B,c}$ the related costs of improving the two common production lines. Using the valuation formula for the substitution operator (see equation (3.4)), the opportunity cost are $F_{\text{sub}}(V_t^{A,c}) \geq 0$, where $V_t^{A,c} = \widehat{W}_t^{A,c} - W_t^{A,c}$, and $F_{\text{sub}}(V_t^{B,c}) \geq 0$ where $V_t^{B,c} = \widehat{W}_t^{B,c} - W_t^{B,c}$. We assume that no improvements are planned for the brand specific production lines.

Given our assumptions, the payoff of the inversion operator is

$$\Pi_{\text{inv}}(t, \mathbf{V}_t^I, V_t^A, V_t^B) = \max\left\{F_{\text{sub}}(\mathbf{V}_t^I) - F_{\text{sub}}(V_t^A) - F_{\text{sub}}(V_t^B), 0\right\}, \tag{3.14}$$

[33]With the inversion operator, the firm replaces two production lines with one common manufacturing center, which has to guarantee the same output, assuming the overall production is unchanged. This decision generates scale economies by reducing the total production costs and in the end it increases the overall cash flow. For clarification, assume that the total revenues of the two brands remain unchanged after the inversion and that the decision affects only the production side. In this case, denoting PC^j the production costs in the j-th unit, $V_t^{I,k} = PC_t^{A,c} + PC_t^{B,c} - PC_t^{\widehat{I}}$, as the production costs of the brand specific production lines do not influence the inversion decision.

and the value is $F_{\text{inv}}(t, \mathbf{V}_t^I, V_t^A, V_t^B)$.

As for the splitting operator, its payoff, which includes the value of the inversion operator, is (dropping some arguments, for brevity)

$$\Pi_{\text{spl}}(t) = \max\left\{W_t^* - W_t - K_{A,c} - K_{A,s} - K_{B,c} - K_{B,s} - C_s + F_{\text{inv}}(t), 0\right\}, \tag{3.15}$$

where $W_t^* - W_t$ is the incremental value of the split design, K_j is the R&D and realization costs of each module, C_s is the splitting cost and $F_{\text{inv}}(t) \geq 0$ is the value of the option to invert the common components.

The solution of the problem involves the simulation of the 11-dimensional Markov process

$$\mathbf{W}_t = \left(W_t^A, W_t^B, W_t^{A,s}, W_t^{B,s}, W_t^{A,c}, W_t^{B,c}, W_t^{I,1}, W_t^{I,2}, W_t^{I,3}, \widehat{W}_t^{A,c}, \widehat{W}_t^{B,c}\right).$$

For valuation purposes and given the discussion in Section 3.3, we assume the values of the modules are correlated GBM under the EMM. The parameters describing the stochastic process of the state variable and the costs are reported in Table 3.3.

[Table 3.3 about here]

The choice of the base case parameters can be motivated as follows. The initial values of the two current systems is such that $W_0^A = W_0^{A,s} + W_0^{A,c}$, and $W_0^B = W_0^{B,s} + W_0^{B,c}$. Namely, we assume that splitting them does not change their current value. The current value of the inverted module is assumed higher than the sum of the current values of the two common modules, for all the three possible versions of the inverted module: $W_0^{I,k} > W_0^{A,c} + W_0^{B,c}$, for $k = 1, 2, 3$. The new versions of the common modules, assuming they are independently pursued, improve upon the current versions, but not upon the inverted module: $W_0^{A,c} + W_0^{B,c} < \widehat{W}_t^{A,c} + \widehat{W}_t^{B,c} < W_0^{I,k}$, for $k = 1, 2, 3$.

As for costs, we assume that the cost to create the new modules is the same for all. The only exception is for the inverted module, as the design, test and realization cost are lumped in one figure, $K_{\widehat{I},k}$. For simplicity, we assume that the cost of the inverted module is the same for all possible versions. The visibility cost, Q, is relatively higher than the other costs, to account for the many changes that the brand–specific modules may require to make them compatible with the new inverted module.[34] Lastly, the splitting cost, C_s, is relatively low because the initial system is almost modular.

[34]These changes do not affect the value since they do not influence the cash flows. They are needed just to make the modules to work together.

As for the stochastic processes of the values of the modules, we assume that their drift coincide with the risk–free rate, or equivalently, the values of the modules do not account for any convenience yield. The variance of W_t^A, W_t^B is (approximately) preserved by the processes $W_t^{A,s} + W_t^{A,c}$, and $W_t^{B,s} + W_t^{B,c}$, respectively. This is to show that the value of splitting is positive also in a situation where the overall uncertainty of the system is unchanged, as suggested by Baldwin and Clark (2000), pag. 259. As for the correlations, we just notice the following. We assume that the values of the common components are strongly and positively correlated, and that they are also positively correlated with the values of the three possible versions of the inverted module. They are positively correlated also with the new versions of the common modules, in case the inversion does not take place. The values of the existing system are assumed to be weakly correlated with all the other modules. Finally, the two brand specific modules are almost uncorrelated with the other modules. The time horizon for the option to split is set at $T_1 = 1$ year, and for the option to invert at $T_2 = 2$ years.

The numerical analysis of this complex modularization decision is reported in Table 3.5. We provide the value of the inverting operator, estimated as the sample average of the values of 40 independent simulations using the LSM method. Each experiment is based on 10,000 paths and 100 time steps. We compute also an estimate of the probability of the application of the inversion operator, as the sample average of the exercise probability. For a specific Monte Carlo experiment, this is the number of paths such that the operator is exercised, over the total number of paths. To capture the timing of the inversion decision, we estimate also the average exercise time, conditional on the fact that such decision is made. For each Monte Carlo experiment, this is the average of the exercise time, for the paths where a decision is made.[35] The exercise probability and the average time are useful to analyze the impact of a greater or smaller flexibility on the decision policy. In Table 3.5 we report in parentheses the standard deviation of the sample estimates.

[Table 3.5 about here]

The analysis of the value and of the optimal policy of the project is based on the break-down of the many sources of flexibility. For this reason, we solve also other sub-problems, where some of the features of the base case model are excluded.

As a first sub-case, we value the same problem considered above, but omitting the inversion operator (see Table 3.5, *Splitting only*). This is equivalent to dropping $F_{\text{inv}}(t)$ from equation

[35] These statistics are based on the EMM. So, if empirical data were available, they could not be compared to the corresponding empirical statistics. Yet, we can legitimately compare them across the different versions of the valuation model.

(3.15), and it permits to determine the incremental value of the inverting operator on top of the value of splitting.[36]

A second interesting variation of the base case is given by the restriction of the inversion operator to the case only one version of the inverted module is tested (see Table 3.5, *Splitting & inversion, with a single test and with opportunity costs*). This changes the payoff in (3.14) to

$$\Pi_{\text{inv}}(t) = \max\left\{F_{sub}(V_t^I) - F_{\text{sub}}(V_t^A) - F_{\text{sub}}(V_t^B), 0\right\},$$

where V_t^I is one-dimensional, and it permits to determine, when compared to the base case, the value of testing several versions of the inverted module.[37]

A third sub-case (see Table 3.5, *Splitting & inversion, with a triple test and no opportunity costs*) entails the inversion of the common module, but not accounting for the opportunity cost of an independent development of the two common modules, $F_{\text{sub}}(V_t^A) = 0 = F_{\text{sub}}(V_t^B)$ in equation (3.14), which reduces to $\Pi_{\text{inv}}(t) = F_{\text{sub}}(\mathbf{V}_t^I)$. This case permits to capture the role of the independent substitution opportunities and how their value change the exercise policy of the inversion operator.[38]

The last variation (see Table 3.5, *Splitting & inversion, with a single test and no opportunity costs*) is given by considering at the same time the restriction to one version of the inverted module and the absence of opportunity costs. This changes the payoff of the inverting operator to $\Pi_{\text{inv}}(t) = F_{\text{sub}}(V_t^I)$ and provides the value of the opportunity to invert the module, with no other form of flexibility derived from this.

The results in Table 3.5 permit to break down the value of the different operators involved in the design problem. By comparing the case with only the splitting operator to the base case, we can see that a significant value is given by the possibility to invert the module ($3.99 - 0.28 = 3.71$). The value of the splitting operator is made of at least two components: the value of the substitution operator on the inverted module, and the opportunity cost due to the reduced flexibility. Unfortunately, there is no easy way to decompose the values of these two operators, as they tend to interact. As a first approximation, the possibility to

---

[36] In this case, the dimension of the state space of the problem is reduced to 6, as the state variable is $\mathbf{W}_t = \left(W_t^A, W_t^B, W_t^{A,s}, W_t^{B,s}, W_t^{A,c}, W_t^{B,c}\right)$.

[37] In this case, the dimension of the state space of the problem is 9, because the state variable is $\mathbf{W}_t = \left(W_t^A, W_t^B, W_t^{A,s}, W_t^{B,s}, W_t^{A,c}, W_t^{B,c}, W_t^I, \widehat{W}_t^{A,c}, \widehat{W}_t^{B,c}\right)$.

[38] The state variable in this case is $\mathbf{W}_t = \left(W_t^A, W_t^B, W_t^{A,s}, W_t^{B,s}, W_t^{A,c}, W_t^{B,c}, W_t^{I,1}, W_t^{I,2}, W_t^{I,3}\right)$, and the dimension is 9.

conduct three tests (as opposed to only one) has a positive value of $3.99 - 2.34 = 1.65$, and the impact of opportunity costs is $9.05 - 3.99 = 5.06$. Yet, when we compare the base case to the case with only one test and no opportunity costs, we see that the combined effect is significantly lower: $5.83 - 3.99 = 1.84$.

A second important aspect of the design problem is the optimal exercise policy for the inversion operator. For this reason, Table 3.5 reports the exercise probability and the average time of exercise (in case inversion actually takes place) under the EMM. It is quite obvious that positive opportunity costs reduce the exercise probability and increase the average time of exercise of the inversion operator, regardless the number of tests on the inverted module. Somehow less obvious is the effect of a higher number of tests. While it is indisputable that this increases the value, it reduces the probability of inverting, as we can see if we compare (taking aside the interaction effect with the opportunity costs) the case with three tests (about 11.9%) to the case with just one test (about 31.1%). And the average time of inversion is longer with three tests (1.54 year) than with one (0.77 year). The above holds true (although at a reduced size) also if we consider the interaction with the opportunity cost. This surprising effect is due to the fact that the growth rate of the substitution operator for the inverted module with three tests is (given the current parameters) significantly higher than the one with just one test. Since the option to invert is American, this induces an optimal delay due to a reduced "convenience yield" for the case with three tests.

3.6 Conclusions

There has been a significant set of contributions on modularity over the last decade. On the managerial side, the conditions and the consequences of the modularization process have been extensively investigated. Much less effort has been devoted to the issues that the modularization process poses in terms of financial valuation for capital budgeting purposes.

In this work we provide a valuation approach based on real options theory, which allows to tackle those issues. We are able to describe the six modular operators proposed by Baldwin and Clark (2000) in a stochastic optimal control framework. Moreover, we show how we can combine the individual operators, thus allowing to evaluate (at least in principle) any modular design. Our approach is implemented numerically using Monte Carlo simulation, with the Least–Squares Monte Carlo method by Longstaff and Schwartz (2001) to cope with the dynamic programming feature of the valuation problems. We show in a set of experiments that the numerical method based on Monte Carlo simulation can be as accurate as binomial lattices.

Although these numerical experiments are very simple, because they involve one operator at the time, the approach we propose is very general, because it can tackle multi-dimensional decision problems and any combination of the modular operators. In the last part of the work, we present a worked out valuation problem involving many operators and addressing the main issues of valuation of modular designs.

3.7 Figures and Tables

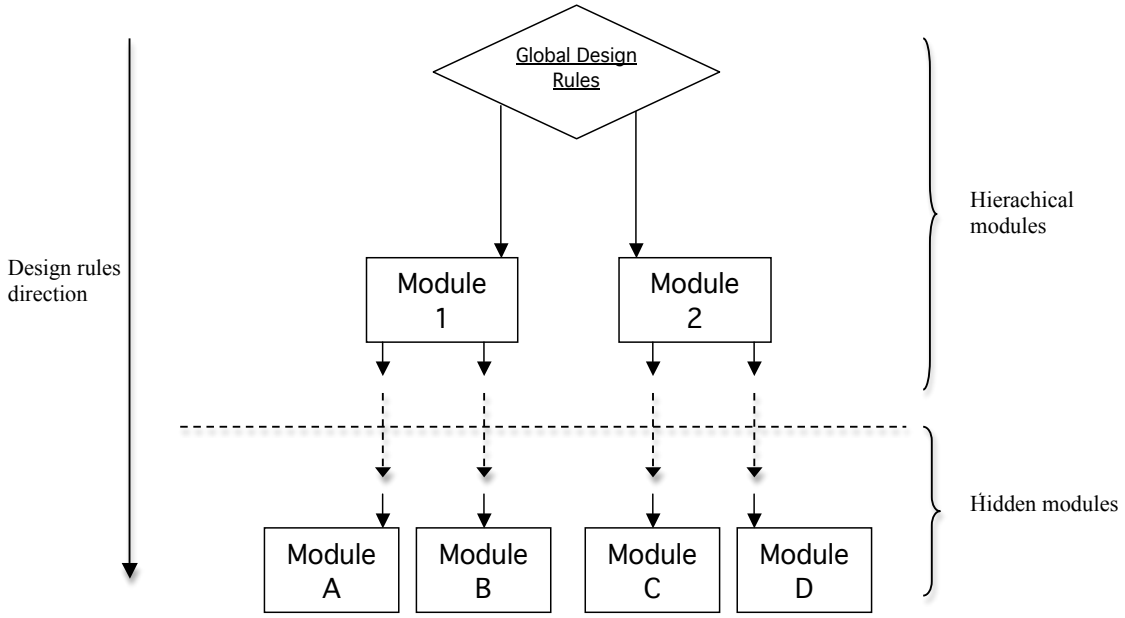

Figure 3.1: **A modular design**. This represents a simplified modular design. Modules placed at the top of the structure are called hierarchical, because they pose a set of design constraints (rules) to the lower-connected modules. The higher the hierarchical position of a module in the modular structure is, the larger is the set of implicit design rules it poses to its lower connected modules. At the bottom of the structure are placed the so called hidden modules. They are free to change as long they obey to their specific design rules posed by upper-connected modules. In a complex design there can be more than one hierarchical layer of modules.

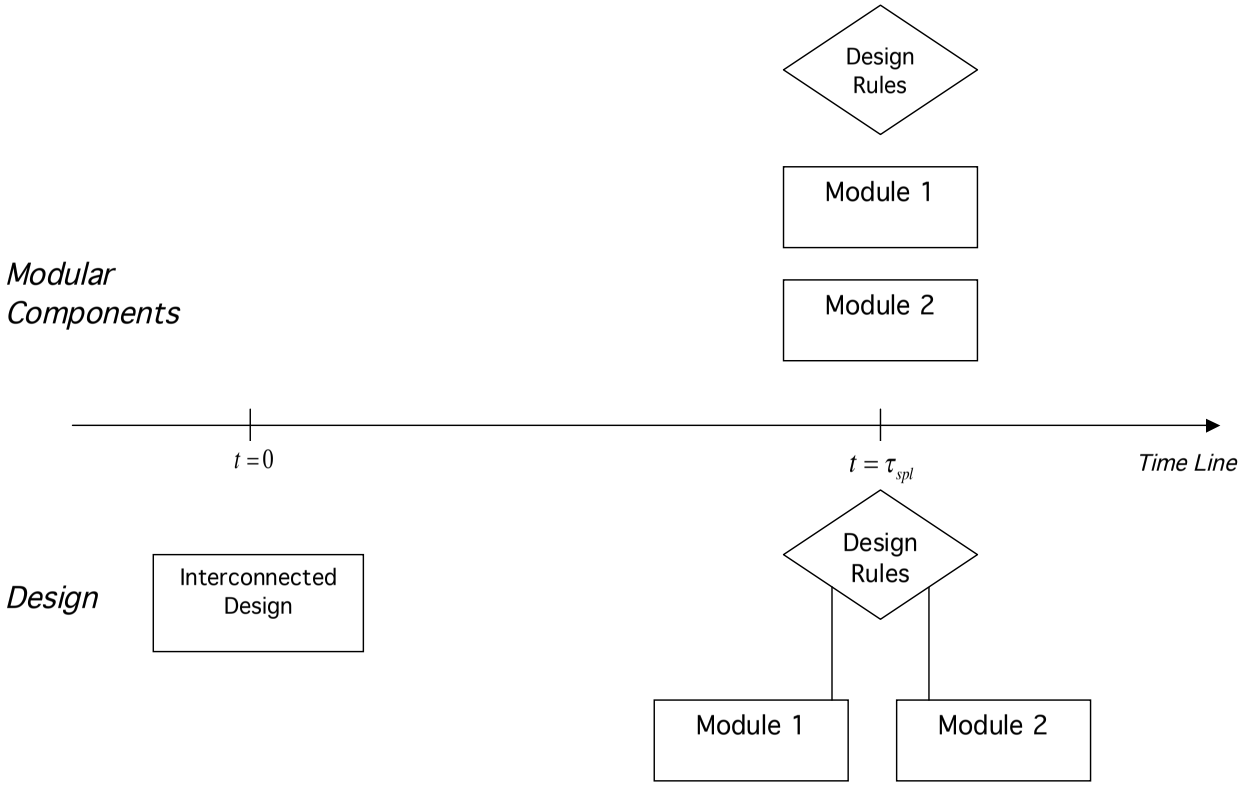

Figure 3.2: **Splitting Operator**. The figure describes an interconnected design split into two modules. When the design is split (at $t = \tau_{spl}$, which is a stopping time), the designer must dictate the design rules and create the modules to be inserted in the modular structure. In the upper part of the figure, above the time line, we show the involved components and the timing of their creation. In the lower part, we show the evolution of the design.

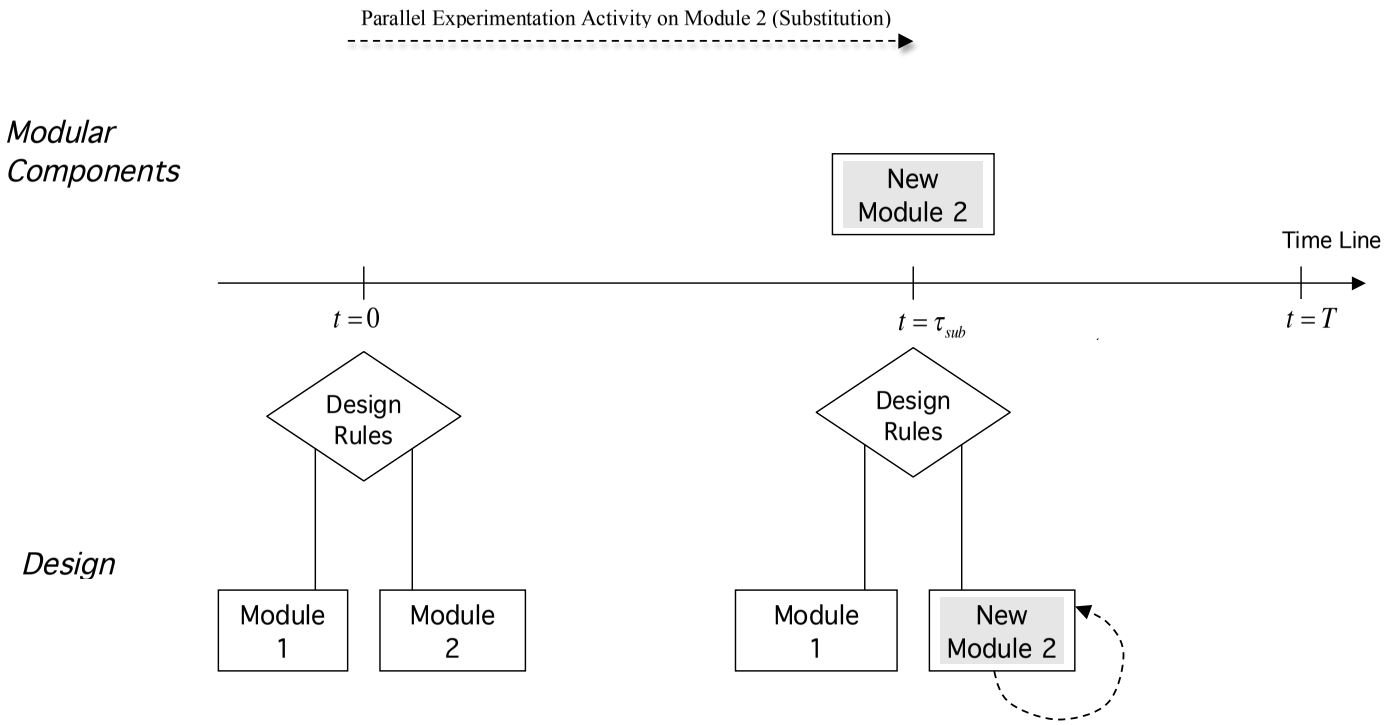

Figure 3.3: **Substitution Operator**. This operator allows to replace an existing module with a new one as a result of research and development activity. The designer starts at $t = 0$ a research on several possible versions a given module. At at $t = \tau_{sub}$, which is a stopping time, she selects the best version among the competitive alternatives. In the upper part of the figure, above the time line, we show the involved components and the timing of their creation. In the lower part, we show the evolution of the design.

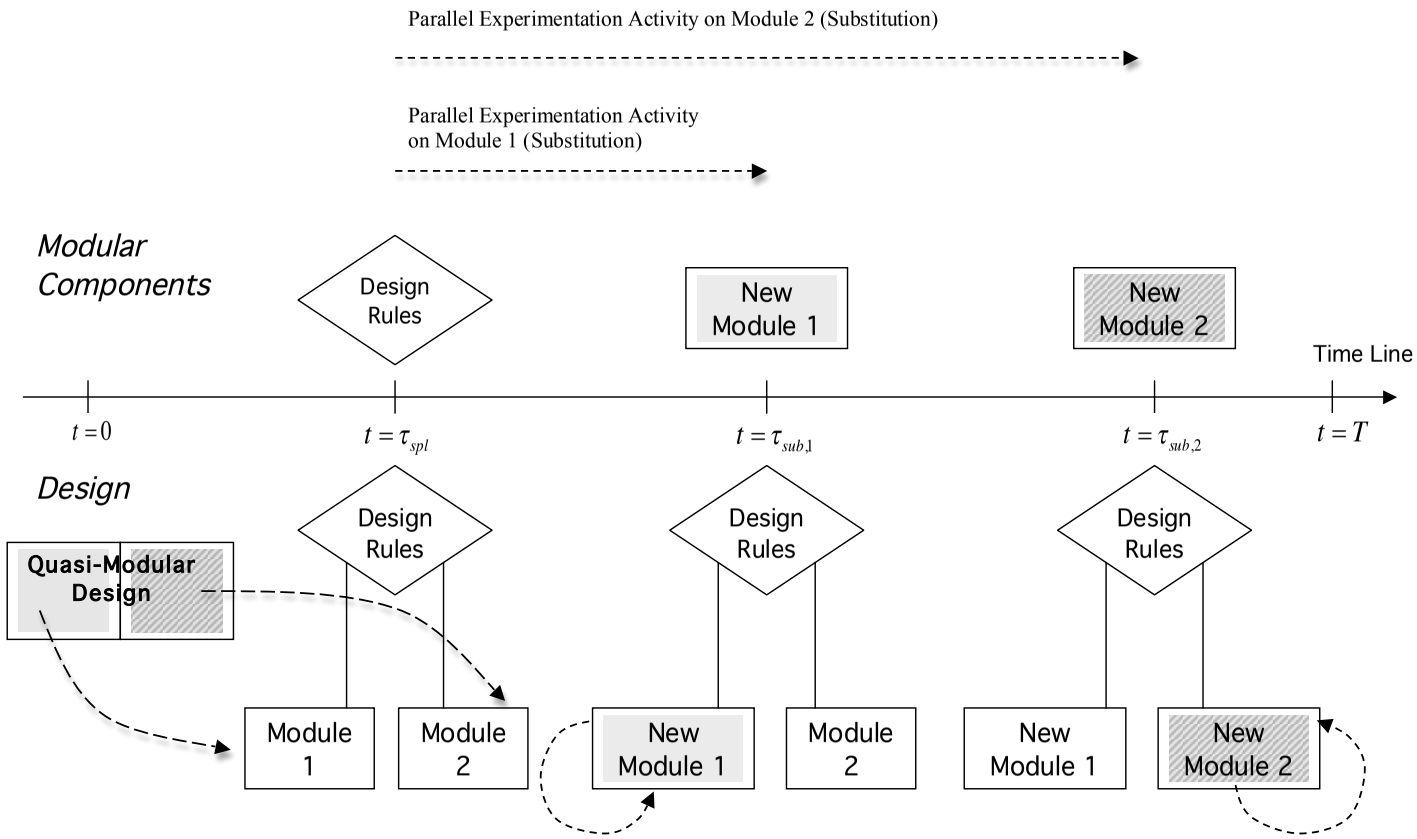

Figure 3.4: **Splitting Operator (revisited)**. This figure shows the splitting operator applied to a rationalized design in which the potential future modules can be defined at the beginning of the modularization process. The decision to split a rationalized design implies the creation of the global design rules at (the stopping time) $t = \tau_{spl}$. Next, research is started on each unit in order to improve the system by implementing the best version of each module, respectively at (the stopping times) $t = \tau_{sub,1}$, and $\tau_{sub,1}$. In the upper part of the figure, above the time line, we show the involved components and the timing of their creation. In the lower part, we show the evolution of the design.

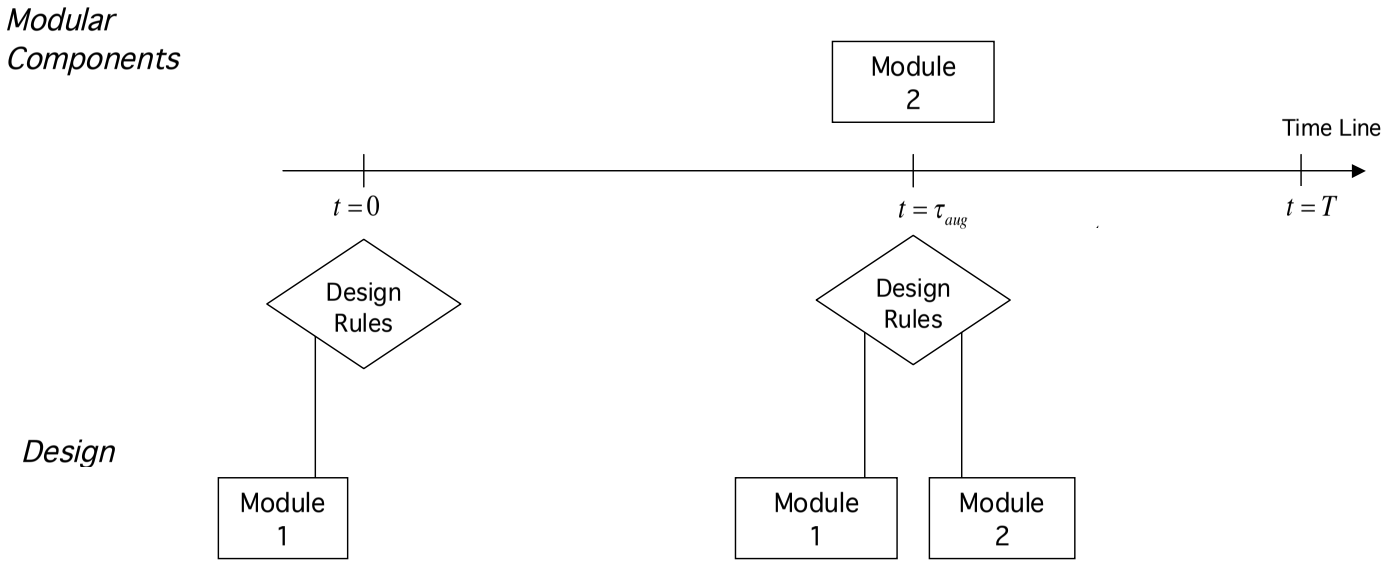

Figure 3.5: **Augmenting Operator**. This operator permits to increase the size of a given design. Specifically, the picture shows a modular design that is augmented at $t = \tau_{aug}$ (a stopping time) with the creation of a new module (i.e. Module 2) which expands the functionality of the whole design. In the upper part of the figure, above the time line, we show the involved components and the timing of their creation. In the lower part, we show the evolution of the design.

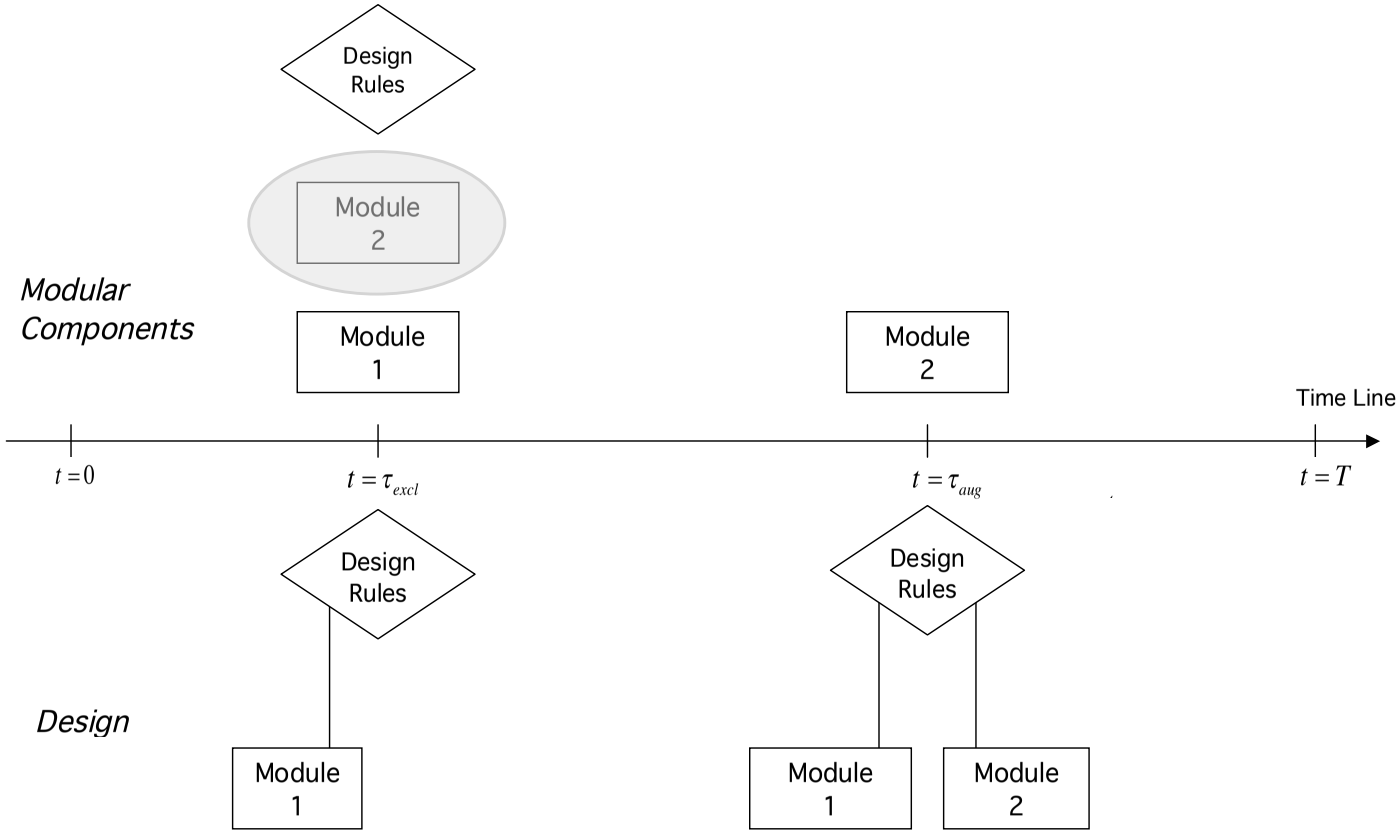

Figure 3.6: **Excluding Operator**. This operator allows to implement (at the stopping time $t = \tau_{excl}$) an initial minimal design with the option to augment it later on if the market conditions turn favorable. The entire modular structure must be determined, although non implemented, at the beginning of the process. I.e., Module 2 is designed at $t = 0$ but its implementation into the modular structure is postponed until $t = \tau_{aug}$, which is a stopping time. In the upper part of the figure, above the time line, we show the involved components and the timing of their creation. In the lower part, we show the evolution of the design.

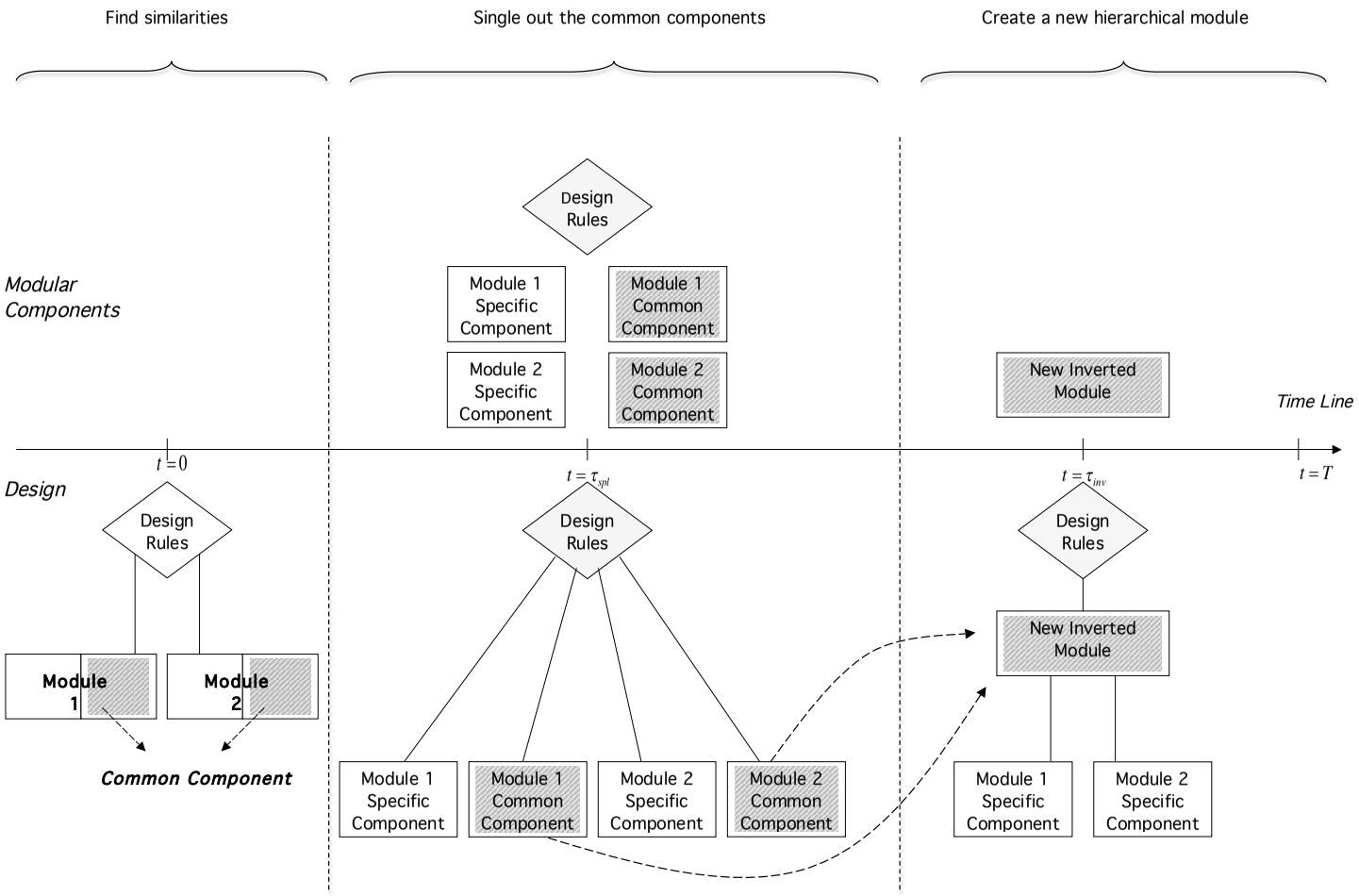

Figure 3.7: **Inversion Operator**. This operator is used to rationalize the design by grouping similar or common functions, which are spread across the structure, into a single module. In the first step, the designer identifies the similar components (left panel). In the second step, she isolates them using the splitting operator (intermediate panel). Finally, she creates a new module ($\widehat{I}$) and connects it to the other modules where the common function was present (right panel). τ_{spl} and τ_{inv} are stopping times. In the upper part of the figure, above the time line, we show the involved components and the timing of their creation. In the lower part, we show the evolution of the design.

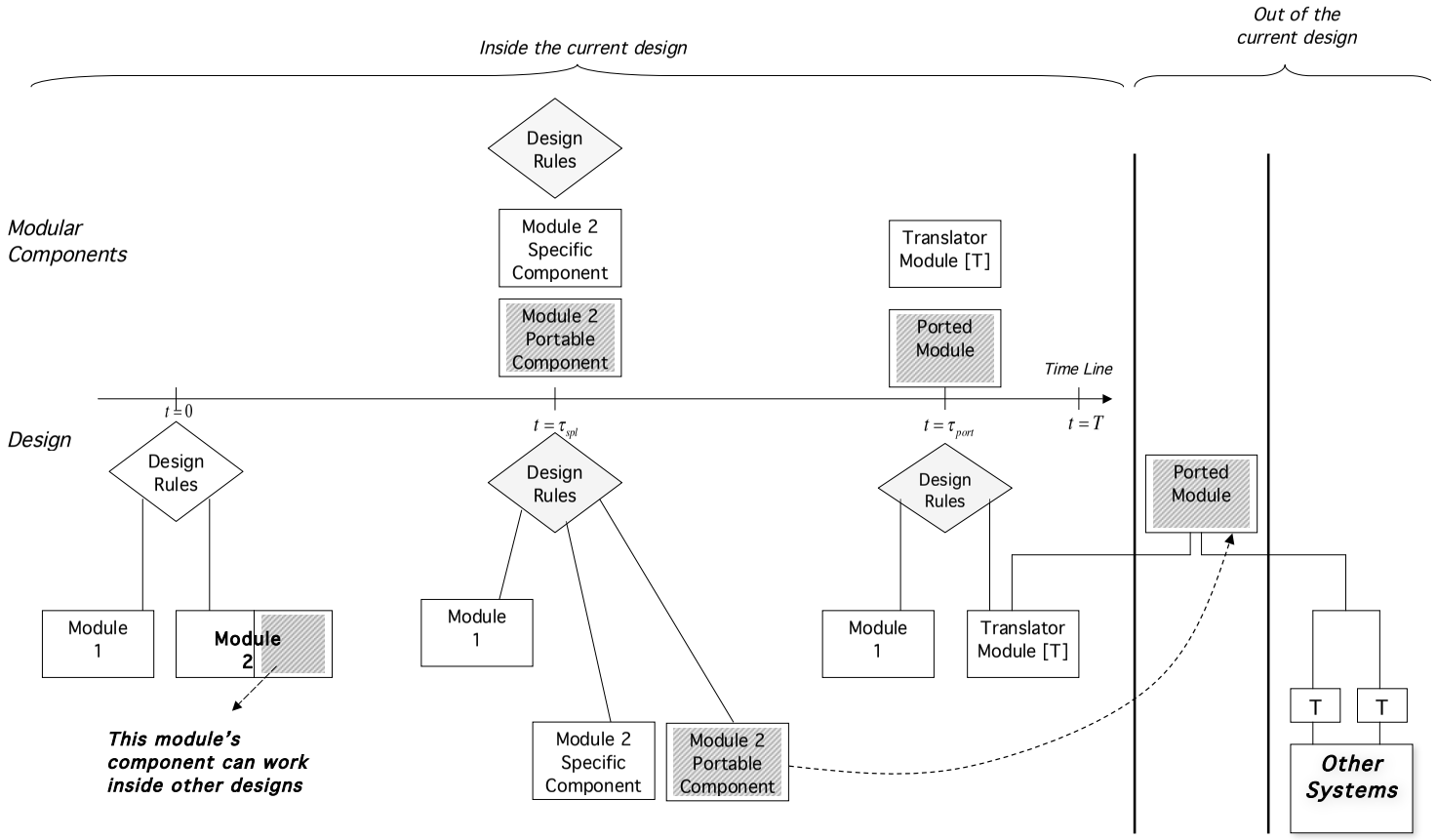

Figure 3.8: **Porting Operator**. This operator is used to port a module from a system to other designs. As with the inversion operator, three steps are required. Identification of the portable unit, splitting of the component, and porting of the module ($\widehat{P}$) outside of the original system (right panel). T denotes the translator module that makes the portable module compatible with the other systems. In the upper part of the figure, above the time line, we show the involved components and the timing of their creation. In the lower part, we show the evolution of the design.

Parameter	Splitting	Substitution	Augmenting	Excluding	Porting
$\mathbf{V}_0$	(7,9)	(6, 10)	10	(7,9)	(13.3, 10, 4, 4)
$\mathbf{C}$	(5, 8)	(5, 8)	(5, 8)	(5,8)	(9, 11, 5, 4)
α	(0.04, 0.06)	(0.04, 0.02)	0.03	(0.04, 0.06)	(0.07, 0.04, 0.04, 0.02)
σ	(0.25, 0.17)	(0.25, 0.17)	0.15	(0.25, 0.17)	(0.1, 0.03, 0.04, 0.02)
$\rho_{i,j}$	0.3	0.3		0.3	(-0.1, 0.3, 0.5, 0.1, 0.6, 0.2)
C_s	3	-	-	0	-
Q	-	0	-	-	1
$\mathbf{R}$	-	-	-	-	(1, 1, 1)
$\mathbf{I}$	-	0	-	-	-
T	1	1	1	(0.5,1)	0.5

Table 3.1: **Parameters used to test consistency of the numerical routines based on LSM method**. $\mathbf{V}_0$ is the initial value of the state variables; in substitution and splitting $\mathbf{C}$ is the (vector of) incremental realization cost relative to the experimental activity while for the others operators it represent the (vector of) research and realization cost of the modules involved in the modularization process; α and σ are the drifts and volatilities, respectively, of the state variables process under the EMM; ρ are the correlation coefficients. For porting, the correlation parameters are $(\rho_{1,2}, \rho_{1,3}, \rho_{1,4}, \rho_{2,3}, \rho_{2,4}, \rho_{3,4})$. C_s is the splitting cost; Q is the visibility cost; $\mathbf{R}$ is the (vector of) realization cost for the translator modules in the porting operator and $\mathbf{I}$ is the (vector of) research and development cost implied by the substitution operator (which is embedded also in the splitting operator). In all the cases, the constant risk-free rate is $r = 0.05$.

	LSM	*s.d.*	*accurate*
Splitting	1.485	0.013	1.478
Substitution	2.504	0.011	2.526
Augmenting	2.219	0.002	2.217
Excluding	3.642	0.012	3.678
Porting	2.418	0.004	2.391

Table 3.2: **Accuracy of the estimates obtained using the Least–Squares Monte Carlo method**. "*LSM*" is the average value obtained over 30 simulations, for splitting, substitution, augmenting and excluding and 20 simulations for porting. "*s.d.*" is the corresponding standard deviation. Each of the LSM estimates are obtained with 8000 paths and 100 steps for substitution, 8000 paths and 120 steps for porting, 10000 and 100 for splitting, 15000 and 100 for augmenting, 10000 and 400 for excluding. The simulated paths are obtained using the Euler discretization of the continuous time dynamics and the antithetic variates technique. "*accurate*" is the value obtained using a lattice method together with a two point Richardson extrapolation. For the augmenting we use the CRR approximation with $(100, 200)$ steps respectively; for the remaining operators we adopt the AGLT approximation by Gamba and Trigeorgis (2007). In particular, for substitution, excluding and splitting we used $(75, 150)$ steps; for porting $(10, 20)$ steps.

Variable	Description	Initial Value	Standard Deviation
W_0^A	value of module A in the initial design	29	0.05
W_0^B	value of module B in the initial design	35	0.05
$W_0^{A,s}$	value of specific component of module A	10	0.06
$W_0^{B,s}$	value of specific component of module B	15	0.04
$W_0^{A,c}$	value of common component of module A	19	0.05
$W_0^{B,c}$	value of common component of module B	20	0.05
$W_0^{I,1}$	value of the first version of the inverted module	50	0.08
$W_0^{I,2}$	value of the second version of the inverted module	50	0.05
$W_0^{I,3}$	value of the third version of the inverted module	50	0.06
$\widehat{W}_0^{A,c}$	value of the alternative version of the common module of A	22	0.0405
$\widehat{W}_0^{B,c}$	value of the alternative version of the common module of B	23	0.0361
$K_{A,c}$	R&D cost of the common module of A	0.5	-
$K_{A,s}$	R&D cost of the specific module of A	0.5	-
$K_{B,c}$	R&D cost of the common module of B	0.5	-
$K_{B,s}$	R&D cost of the specific module of B	0.5	-
$K_{\widehat{I},1}$	R&D cost of the first version of the inverted module	1.5	-
$K_{\widehat{I},2}$	R&D cost of the second version of the inverted module	1.5	-
$K_{\widehat{I},3}$	R&D cost of the third version of the inverted module	1.5	-
$C_{A,c}$	R&D cost of the alternative version of the common module of A	0.5	-
$C_{B,c}$	R&D cost of the alternative version of the common module of A	0.5	-
C_s	splitting cost	0.5	-
Q	visibility cost	2	-
T_1	time horizon of the splitting operator	1	-
T_2	time horizon of the inverting operator	2	-

Table 3.3: **Valuation of a complex design**. Base case parameters.

$$\begin{pmatrix}
1 & 0.02 & 0.02 & 0.05 & 0.02 & 0.04 & 0.04 & 0.05 & 0.02 & 0.1 & 0.1 \\
0.02 & 1 & 0.02 & 0.02 & 0.03 & 0.1 & 0.02 & 0.03 & 0.1 & 0.03 & 0.1 \\
0.02 & 0.02 & 1 & 0.5 & 0.1 & 0.02 & 0.13 & 0.16 & 0.12 & 0.1 & 0.1 \\
0.05 & 0.02 & 0.5 & 1 & 0.2 & 0.2 & 0 & 0.3 & 0.2 & 0.1 & 0.2 \\
0.02 & 0.03 & 0.1 & 0.2 & 1 & 0.5 & 0.4 & 0.5 & 0.4 & 0.3 & 0.3 \\
0.04 & 0.1 & 0.02 & 0.2 & 0.5 & 1 & 0.5 & 0.5 & 0.4 & 0.3 & 0.3 \\
0.04 & 0.02 & 0.13 & 0 & 0.4 & 0.5 & 1 & 0.6 & 0.1 & 0.5 & 0.3 \\
0.05 & 0.03 & 0.16 & 0.3 & 0.5 & 0.5 & 0.6 & 1 & 0.3 & 0.1 & 0.1 \\
0.02 & 0.1 & 0.12 & 0.2 & 0.4 & 0.4 & 0.1 & 0.3 & 1 & 0.5 & 0.1 \\
0.1 & 0.03 & 0.1 & 0.1 & 0.3 & 0.3 & 0.5 & 0.1 & 0.5 & 1 & 0.2 \\
0.1 & 0.1 & 0.1 & 0.2 & 0.3 & 0.3 & 0.3 & 0.1 & 0.1 & 0.2 & 1
\end{pmatrix}$$

Table 3.4: **Correlation matrix**. The state variables are presented in the same order as in the first panel of Table 3.3.

	Value	Exercise Probability	Average Time
Base Case	3.9935	0.0851	1.6603
	(0.0049)	*(0.0007)*	*(0.0040)*
Splitting & inversion triple test, no opp. cost	9.0545	0.1187	1.5378
	(0.0068)	*(0.0016)*	*(0.0123)*
Splitting & inversion single test, no opp. cost	5.8324	0.3114	0.7726
	(0.0050)	*(0.0180)*	*(0.0376)*
Splitting & inversion single test, opp. cost	2.3447	0.1116	1.6673
	(0.0052)	*(0.0008)*	*(0.0026)*
Splitting only	0.2782	-	-
	(0.0013)	-	-

Table 3.5: **The value of a complex design**. The table presents the value for the base case modularization problem, and for four variations, based on the number of tests (one vs three) for the inverted module, and on the consideration of the opportunity costs related to the reduced flexibility of the system with the inverted module. As a benchmark case, also the value of the splitting operator (with no additional flexibility) is presented. The table reports also the exercise probability (under the EMM) for the decision to invert, and the average time of exercise of the decision to invert (under the EMM), for those paths where the inversion takes place. These statistics are estimated from a sample of 40 independent Monte Carlo experiments (in parentheses we report the standard deviations of the estimates). Each experiment is based on 10,000 paths and 100 time steps and the solution (value and policy) is found using the LSM method.

Appendix A

A.1 Proof of Lemma 1

Proof. Expanding equation (1.2), we obtain

$$\begin{aligned}\beta'(\mathbf{RV}_t, L_t) &= \ddot{\beta}_1 RV_t + \frac{1}{5}\ddot{\beta}_2 RV_t + \frac{1}{5}\ddot{\beta}_2 RV_{t-1} + \cdots + \frac{1}{5}\ddot{\beta}_2 RV_{t-4} + \frac{1}{22}\ddot{\beta}_3 RV_t + \\ &\quad \frac{1}{22}\ddot{\beta}_3 RV_{t-1} + \cdots + \frac{1}{22}\ddot{\beta}_3 RV_{t-21} + \Lambda L_t\end{aligned}$$

and re–arranging terms in order to collect all the variables with the same lag on RV, we have

$$\beta'(\mathbf{RV}_t, L_t) = \beta_1 RV_t + \frac{\beta_2}{5}(\sum_{i=1}^{4} RV_{t-i}) + \frac{\beta_3}{22}(\sum_{i=5}^{21} RV_{t-i}) + \beta_4 L_t$$

where $\beta_1 = (\ddot{\beta}_1 + \frac{1}{5}\ddot{\beta}_2 + \frac{1}{22}\ddot{\beta}_3)$, $\beta_2 = (\frac{1}{5}\ddot{\beta}_2 + \frac{1}{22}\ddot{\beta}_3)$, $\beta_3 = \frac{1}{22}\ddot{\beta}_3$ and $\beta_4 = \Lambda$. Clearly the shape and the scale parameters are not affected by the re–arrangement and therefore they remain unchanged. □

A.2 Proof of Proposition 1.1

Proof. Let us compute

$$\begin{aligned}\mathbb{E}_t^{\mathbb{P}}\left[\exp(-\alpha' K_{t+1})\right] &= \mathbb{E}_t^{\mathbb{P}}\left[\exp(-\alpha_1 y_{t+1} - \alpha_2 RV_{t+1})\right] \\ &= \mathbb{E}_t^{\mathbb{P}}\left[\exp(-\alpha_1\sqrt{RV_{t+1}}\varepsilon_{t+1} - (\alpha_2 + \gamma\alpha_1)RV_{t+1})\right] \\ &= \mathbb{E}_t^{\mathbb{P}}\left[\exp\left(-\left(\alpha_2 + \gamma\alpha_1 - \frac{1}{2}\alpha_1^2\right)RV_{t+1}\right)\right] \\ &= \exp\left[-b\left(\alpha_2 + \gamma\alpha_1 - \frac{1}{2}\alpha_1^2\right) - a\left(\alpha_2 + \gamma\alpha_1 - \frac{1}{2}\alpha_1^2\right)\beta'(\mathbf{RV}_t, L_t)\right] \qquad \text{(A.1)} \\ &= \varphi_K^{\mathbb{P}}(v), \qquad \text{(A.2)}\end{aligned}$$

with $v := \alpha_2 + \gamma\alpha_1 - \frac{1}{2}\alpha_1^2$. □

A.3 Proof of Proposition 1.2

Proof. For the sake of simplicity we assume a zero expected instantaneous rate of return ($r = 0$). The no-arbitrage restrictions are:

$$\begin{aligned} \mathbb{E}_t^{\mathbb{P}}(M_{t,t+1}) &= 1 \\ \mathbb{E}_t^{\mathbb{P}}(M_{t,t+1}\exp(y_{t+1})) &= 1 \end{aligned}$$

$$\mathbb{E}_t^{\mathbb{P}}(\exp(-\nu_0 - \nu_1 RV_{t+1} - \nu_2 RV_t^d - \nu_3 RV_{t-1}^w - \nu_4 RV_{t-5}^m - \nu_5 y_{t+1} - \nu_6 L_t)) = 1 \tag{A.3}$$

$$\mathbb{E}_t^{\mathbb{P}}(\exp(-\nu_0 - \nu_1 RV_{t+1} - \nu_2 RV_t^d - \nu_3 RV_{t-1}^w - \nu_4 RV_{t-5}^m - (\nu_5 - 1) y_{t+1} - \nu_6 L_t)) = 1 \tag{A.4}$$

Using the moment generating function of y_{t+1}, the LHS of the first restriction (Eq. (A.3)) becomes:

$$\begin{aligned} & \mathbb{E}_t^{\mathbb{P}}\{\exp(-\nu_0 - \nu_1 RV_{t+1} - \nu_2 RV_t^d - \nu_3 RV_{t-1}^w - \nu_4 RV_{t-5}^m - \nu_5 y_{t+1} - \nu_6 L_t)\} \\ = \; & \mathbb{E}_t^{\mathbb{P}}\{\exp(-\nu_0 - \nu_1 RV_{t+1} - \nu_2 RV_t^d - \nu_3 RV_{t-1}^w - \nu_4 RV_{t-5}^m - \nu_6 L_t) \\ & \times \mathbb{E}_t^{\mathbb{P}}[\exp(-\nu_5 y_{t+1}) | RV_{t+1}]\} \\ = \; & \mathbb{E}_t^{\mathbb{P}}\{\exp\left(-\nu_0 - \nu_1 RV_{t+1} - \nu_2 RV_t^d - \nu_3 RV_{t-1}^w - \nu_4 RV_{t-5}^m - \nu_5 \gamma RV_{t+1} - \nu_6 L_t\right) \\ & \times \exp\left(\frac{RV_{t+1}\nu_5^2}{2}\right)\} \\ = \; & \mathbb{E}_t^{\mathbb{P}}\{\exp(-\nu_0 - \left(\nu_1 + \nu_5\gamma - \frac{\nu_5^2}{2}\right) RV_{t+1} - \nu_2 RV_t^d - \nu_3 RV_{t-1}^w - \nu_4 RV_{t-5}^m - \nu_6 L_t)\}. \end{aligned}$$

We observe that the last equation is equivalent to:

$$\begin{aligned} & \mathbb{E}_t^{\mathbb{P}}\{\exp(\nu_0 - \left(\nu_1 + \nu_5\gamma - \frac{\nu_5^2}{2}\right) RV_{t+1} - \nu_2 RV_t^d - \nu_3 RV_t^w - \nu_4 RV_t^m - \nu_6 L_t)\} \\ = \; & \exp\left(-\nu_0 - \nu_2 RV_t^d - \nu_3 RV_t^w - \nu_4 RV_t^m - \nu_6 L_t\right) \varphi_{ARG}^{\mathbb{P}}(u), \end{aligned}$$

where $\varphi_{ARG}(u)$ in given in (1.5) and $u := \nu_1 + \nu_5\gamma - \nu_5^2/2$. More specifically, we have

$$\begin{aligned} & \exp\left(-\nu_0 - \nu_2 RV_t^d - \nu_3 RV_t^w - \nu_4 RV_t^m - \nu_6 L_t\right) \exp\left(-a(u,\rho)\beta'(\mathbf{RV}_t, L_t) - b(u)\right) \\ = \; & \exp\left(-\nu_0 - \nu_2 RV_t^d - \nu_3 RV_t^w - \nu_4 RV_t^m - \nu_6 L_t\right) \exp\left(-b(u)\right) \\ & \times \exp\left(-\frac{cu}{1+cu}\left(\beta_1 \underbrace{RV_t}_{RV_t^d} + \beta_2 \underbrace{(\sum_{i=1}^{4} RV_{t-i})}_{RV_t^w} + \beta_3 \underbrace{(\sum_{i=5}^{21} RV_{t-i})}_{RV_t^m} + \beta_4 L_t\right)\right), \end{aligned}$$

where $b(u)$ is given in (1.6).

Now, let us consider also the the LHS of the equation (A.4) (applying the law of iterated expectations, as we have done for the LHS of equation (A.3)), we have:

$$\begin{aligned}
& \mathbb{E}_t^{\mathbb{P}}\{\exp(-\nu_0-\nu_1 RV_{t+1}-\nu_2 RV_t^d-\nu_3 RV_t^w-\nu_4 RV_t^m-\nu_6 L_t) \\
& \times \mathbb{E}_t^{\mathbb{P}}[\exp(-(\nu_5-1)y_{t+1})|RV_{t+1}]\} \\
= \; & \mathbb{E}_t^{\mathbb{P}}\{\exp(-\nu_0-\nu_1 RV_{t+1}-\nu_2 RV_t^d-\nu_3 RV_t^w-\nu_4 RV_t^m-(\nu_5-1)\gamma RV_{t+1}-\nu_6 L_t) \\
& \times \exp\left(\frac{RV_{t+1}(\nu_5-1)^2}{2}\right)\} \\
= \; & \mathbb{E}_t^{\mathbb{P}}\{\exp(-\nu_0-\left(\nu_1+(\nu_5-1)\gamma-\frac{(\nu_5-1)^2}{2}\right)RV_{t+1}-\nu_2 RV_t^d-\nu_3 RV_t^w-\nu_4 RV_t^m-\nu_6 L_t)\} \\
= \; & \exp\left(-\nu_0-\nu_2 RV_t^d-\nu_3 RV_t^w-\nu_4 RV_t^m-\nu_6 L_t\right)\varphi_{ARG}^{\mathbb{P}}(\tilde{u})
\end{aligned}$$

where $\tilde{u}:=\left(\nu_1+(\nu_5-1)\gamma-\frac{(\nu_5-1)^2}{2}\right)$. Then the last equation becomes:

$$\begin{aligned}
& \exp\left(-\nu_0-\nu_2 RV_t^d-\nu_3 RV_t^w-\nu_4 RV_t^m-\nu_6 L_t\right)\exp\left(-a(\tilde{u},\rho)\beta'(\mathbf{RV}_t,L_t)-b(\tilde{u})\right) \\
= \; & \exp\left(-\nu_0-\nu_2 RV_t^d-\nu_3 RV_t^w-\nu_4 RV_t^m-\nu_6 L_t\right)\exp\left(-b(\tilde{u})\right) \\
& \times \exp\left(-\frac{c\tilde{u}}{1+c\tilde{u}}\left(\beta_1\underbrace{RV_t}_{RV_t^d}+\beta_2\underbrace{(\sum_{i=1}^{4}RV_{t-i})}_{RV_t^w}+\beta_3\underbrace{(\sum_{i=5}^{21}RV_{t-i})}_{RV_t^m}+\beta_4 L_t\right)\right),
\end{aligned}$$

where $b(\tilde{u})$ is given in (1.6).

Therefore, in order to satisfy both the no–arbitrage conditions in (A.3) and (A.4), the following system has to be satisfied:

$$\begin{cases} -\nu_0-\nu_2 RV_t^d-\nu_3 RV_t^w-\nu_4 RV_t^m-\nu_6 L_t-a(u,\rho)\;\beta'(\mathbf{RV}_t,L_t))-b(u)=0 \\ -\nu_0-\nu_2 RV_t^d-\nu_3 RV_t^w-\nu_4 RV_t^m-\nu_6 L_t-a(\tilde{u},\rho)\;\beta'(\mathbf{RV}_t,L_t))-b(\tilde{u})=0 \end{cases},$$

or equivalently

$$\begin{cases} -\nu_0-\nu_2 RV_t^d-\nu_3 RV_t^w-\nu_4 RV_t^m-\nu_6 L_t-\frac{cu}{1+cu}\left(\beta_1 RV_t^d+\beta_2 RV_t^w+\beta_3 RV_t^m+\beta_4 L_t\right)-b(u)=0 \\ -\nu_0-\nu_2 RV_t^d-\nu_3 RV_t^w-\nu_4 RV_t^m-\nu_6 L_t-\frac{c\tilde{u}}{1+c\tilde{u}}\left(\beta_1 RV_t^d+\beta_2 RV_t^w+\beta_3 RV_t^m+\beta_4 L_t\right)-b(\tilde{u})=0 \end{cases}. \tag{A.5}$$

The system in (A.5) implies (considering (1.7) and both the expressions for u and $\tilde{u}$) the following system of equations:

$$\left\{\begin{array}{c}
\nu_0+b\left(\nu_1+\nu_5\gamma-\frac{\nu_5^2}{2}\right)=0\\
\nu_0+b\left(\nu_1+(\nu_5-1)\gamma-\frac{(\nu_5-1)^2}{2}\right)=0\\
\nu_2+\frac{c\left(\nu_1+\nu_5\gamma-\frac{\nu_5^2}{2}\right)}{1+\left(\nu_1+\nu_5\gamma-\frac{\nu_5^2}{2}\right)c}\beta_1=0\\
\nu_3+\frac{c\left(\nu_1+\nu_5\gamma-\frac{\nu_5^2}{2}\right)}{1+\left(\nu_1+\nu_5\gamma-\frac{\nu_5^2}{2}\right)c}\beta_2=0\\
\nu_4+\frac{c\left(\nu_1+\nu_5\gamma-\frac{\nu_5^2}{2}\right)}{1+c\left(\nu_1+\nu_5\gamma-\frac{\nu_5^2}{2}\right)c}\beta_3=0\\
\nu_6+\frac{c\left(\nu_1+\nu_5\gamma-\frac{\nu_5^2}{2}\right)}{1+c\left(\nu_1+\nu_5\gamma-\frac{\nu_5^2}{2}\right)}\beta_4=0
\end{array}\right. .$$

To solve it, we first set

$$\begin{aligned}
\nu_5\gamma-\frac{\nu_5^2}{2} &= (\nu_5-1)\gamma-\frac{(\nu_5-1)^2}{2}\\
\nu_5 &= \gamma+\frac{1}{2}
\end{aligned}$$

and the other solutions are:

$$\nu_0=-\delta\log\left(1+c\left(\frac{1}{2}\gamma^2+\nu_1-\frac{1}{8}\right)\right),$$

$$\begin{aligned}
\nu_2 &= -\frac{4c\beta_1\gamma^2-c\beta_1+8c\nu_1\beta_1}{4c\gamma^2-c+8c\nu_1+8}\\
&= -c\beta_1\frac{\gamma^2/2+\nu_1-1/8}{c\left(\gamma^2/2-1/8-\nu_1\right)+1}
\end{aligned}$$

$$\nu_3=-c\beta_2\frac{\gamma^2/2+\nu_1-1/8}{c\left(\gamma^2/2-1/8-\nu_1\right)+1},$$

$$\nu_4=-c\beta_3\frac{\gamma^2/2+\nu_1-1/8}{c\left(\gamma^2/2-1/8-\nu_1\right)+1},$$

$$\nu_6=-c\beta_4\frac{\gamma^2/2+\nu_1-1/8}{c\left(\gamma^2/2-1/8-\nu_1\right)+1}.$$

□

A.4 Proof of Proposition 1.3

Proof. Let us compute

$$\begin{aligned}
& \mathbb{E}_t^{\mathbb{Q}}\left[\exp(-\alpha' K_{t+1})\right] = \mathbb{E}_t^{\mathbb{Q}}\left[\exp(-\alpha_1 y_{t+1} - \alpha_2 RV_{t+1})\right] \\
= \;& \mathbb{E}_t^{\mathbb{P}}\left[M_{t,t+1}\exp(-\alpha_1 y_{t+1} - \alpha_2 RV_{t+1})\right] \\
= \;& \mathbb{E}_t^{\mathbb{P}}\left[\exp(-\nu_0 - \nu_2 RV_t^d - \nu_3 RV_t^w - \nu_4 RV_t^m - y_{t+1}\left(\nu_5+\alpha_1\right) - RV_{t+1}\left(\nu_1+\alpha_2\right) - \nu_6 L_t)\right] \\
= \;& \mathbb{E}_t^{\mathbb{P}}(\exp(-\nu_0 - \nu_2 RV_t^d - \nu_3 RV_t^w - \nu_4 RV_t^m - \left(\gamma RV_{t+1} + \sqrt{RV_{t+1}}\epsilon_{t+1}\right)\left(\nu_5+\alpha_1\right)) \\
& \times \exp(-RV_{t+1}\left(\nu_1+\alpha_2\right) - \nu_6 L_t)) \\
= \;& \mathbb{E}_t^{\mathbb{P}}(\exp(-\nu_0 - \nu_2 RV_t^d - \nu_3 RV_t^w - \nu_4 RV_t^m + \frac{\left(\nu_5+\alpha_1\right)^2}{2} RV_{t+1}) \\
& \times(-RV_{t+1}\left(\left(\nu_5+\alpha_1\right)\gamma + \left(\nu_1+\alpha_2\right)\right) - \nu_6 L_t) \\
= \;& \mathbb{E}_t^{\mathbb{P}}\left[\exp\left(-\nu_0 - \nu_2 RV_t^d - \nu_3 RV_t^w - \nu_4 RV_t^m - \varpi RV_{t+1} - \nu_6 L_t\right)\right] \\
= \;& \exp\left(-\nu_0 - \nu_2 RV_t^d - \nu_3 RV_t^w - \nu_4 RV_t^m - \nu_6 L_t\right)\mathbb{E}_t^{\mathbb{P}}\left[\exp\left(-\varpi RV_{t+1}\right)\right] \\
= \;& \exp\left(-\nu_0 - \nu_2 RV_t^d - \nu_3 RV_t^w - \nu_4 RV_t^m - \nu_6 L_t\right)\varphi_{ARG}^{\mathbb{P}}\left(\varpi\right) \\
= \;& \exp\left(-\nu_0 - \nu_2 RV_t^d - \nu_3 RV_t^w - \nu_4 RV_t^m - \nu_6 L_t\right)\exp\left(-a\left(\varpi\right)\left(\beta'\left(\mathbf{RV}_t, L_t\right)\right) - b\left(\varpi\right)\right) \\
= \;& \exp\left(-\nu_0 - b\left(\varpi\right)\right)\exp\left(-\nu_2 RV_t^d - \nu_3 RV_t^w - \nu_4 RV_t^m - \nu_6 L_t\right) \\
& \times \exp\left(-a\left(\varpi\right)\left(\beta_1 RV_t^d + \beta_2 RV_t^w + \beta_3 RV_t^m + \beta_4 L_t\right)\right) \\
= \;& \exp\left(-\nu_0 - b\left(\varpi\right)\right) \\
& \times \exp\left(RV_t^d(-a\left(\varpi\right)\beta_1 - \nu_2) + RV_t^w(-a\left(\varpi\right)\beta_2 - \nu_3) + RV_t^m(-a\left(\varpi\right)\beta_3 - \nu_4) + L_t(-a\left(\varpi\right)\beta_4 - \nu_6)\right)
\end{aligned}$$

with $\varpi := -\frac{\left(\nu_5+\alpha_1\right)^2}{2} + \left(\nu_5+\alpha_1\right)\gamma + \left(\nu_1+\alpha_2\right)$. Now considering the non–arbitrage conditions in Proposition 1.2, ϖ becomes

$$\begin{aligned}
\varpi &= -\frac{\left(\nu_5+\alpha_1\right)^2}{2} + \left(\nu_5+\alpha_1\right)\gamma + \left(\nu_1+\alpha_2\right) = \alpha_1(\gamma-\nu_5) + \alpha_2 - \frac{1}{2}\alpha_1^2 + \gamma\nu_5 + \nu_1 - \frac{\nu_5^2}{2} \\
&= -\frac{1}{2}\alpha_1 + \alpha_2 - \frac{1}{2}\alpha_1^2 + \lambda
\end{aligned}$$

where $\lambda = \nu_1 + \frac{\gamma^2}{2} - \frac{1}{8}$, $\nu_0 = -b(\lambda)$ and we also have

$$\nu_2 = -a(\lambda)\beta_1, \nu_3 = -a(\lambda)\beta_2,\ \nu_4 = -a(\lambda)\beta_3 \text{ and } \nu_6 = -a(\lambda)\beta_4$$

Thus, we get:

$$\mathbb{E}_t^{\mathbb{Q}}\left[exp(-\alpha' K_{t+1})\right] = exp\left[-a^*\left(-\frac{1}{2}\alpha_1 + \alpha_2 - \frac{1}{2}\alpha_1^2\right)\left(\beta\left(\mathbf{RV}_t, L_t\right)\right) - b^*\left(-\frac{1}{2}\alpha_1 + \alpha_2 - \frac{1}{2}\alpha_1^2\right)\right],$$

in which the vector $a^*(\zeta)\beta^*$ is such that:

$$a^*(\zeta)\beta = a(\zeta+\lambda)\beta - a(\lambda)\beta = \frac{c^*\beta\zeta}{1+c^*\zeta},$$

$$b^*(\zeta) = b(\zeta+\lambda) - b(\lambda) = \delta^* ln(1+c^*\zeta),$$

where

$$\begin{aligned} \beta^* &= \frac{\beta}{(1+c\lambda)}, \\ \delta^* &= \delta, \\ c^* &= \frac{c}{1+c\lambda}. \end{aligned}$$

□

Bibliography

Adrian, T. and Rosenberg, J. (2007). Stock Returns and Volatility: Pricing the Long-Run and Short-Run Components of Market Risk. *Journal of Finance, forthcoming*.

Ahn, D. H., Figlewski, S., and Gao, B. (1999). Pricing discrete barrier options with an adaptive mesh model. *Working paper*.

Aït-Sahalia, Y. and Mancini, L. (2008). Out of sample forecasts of quadratic variation. *Journal of Econometrics*, 147(1):17–33.

Andersen, T. G., Bollerslev, T., Diebold, F. X., and Ebens, H. (2001a). The distribution of stock returns volatilities. *Journal of Financial Economics*, 61:43–76.

Andersen, T. G., Bollerslev, T., Diebold, F. X., and Labys, P. (2000). Great realizations. *Risk*, 13(3):105–8.

Andersen, T. G., Bollerslev, T., Diebold, F. X., and Labys, P. (2001b). The distribution of realized exchange rate volatility. *Journal of the American Statistical Association*, 96:42–55.

Andersen, T. G., Bollerslev, T., Diebold, F. X., and Labys, P. (2003). Modeling and forecasting realized volatility. *Econometrica*, 71:579–625.

Ane, T. and Geman, H. (2000). Order flow, trasaction clock, and normality of asset returns. *The Journal of Finance*, LV:2259–2284.

Baldi, P., Caramellino, L., and Iovino, M. G. (1999). Pricing general barrier options: a numerical approach using sharp large deviations. *Mathematical Finance*, 9:293–322.

Baldwin, C. Y. and Clark, K. B. (2000). *Design Rules: The Power of Modularity*. MIT Press, Cambridge, MA.

Baldwin, C. Y. and Clark, K. B. (2004). Modularity in the design of complex engineering system. working paper 04–055, Harvard Business School.

Barndorff-Nielsen, O., Kent, J., and Sorensen, M. (1982). Normal variance-mean mixtures and z distribution. *International Statistical Review*, 50:145–159.

Barndorff-Nielsen, O. and Shephard, N. (2002a). Econometric analysis of realized volatility and its use in estimating stochastic volatility models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64:253–280.

Barndorff-Nielsen, O. E. and Shephard, N. (2001). Non-gaussian ornstein-uhlembech-based models and some of their rses in financial economics. *Journal of the Royal Statistical Society*, Series B(63):167–241.

Barndorff-Nielsen, O. E. and Shephard, N. (2002b). Estimating quadratic variation using realized variance. *Journal of Applied Econometrics*, 17:457–477.

Barndorff-Nielsen, O. E. and Shephard, N. (2005). How accurate is the asymptotic approximation to the distribution of realized volatility? In Andrews, D. W. F. and Stock, J. H., editors, *Identification and Inference for Econometric Models. A Festschrift in Honour of T.J. Rothenberg*, pages 306–331. Cambridge University Press.

Bertholon, H., Monfort, A., and Pegoraro, F. (2008). Econometric asset pricing modelling. *Journal of Financial Econometrics*, 6:407–458.

Bollerslev, T. and Ole Mikkelsen, H. (1996). Modeling and pricing long memory in stock market volatility. *Journal of Econometrics*, 73(1):151–184.

Bonaccorsi, A. and Rossetto, S. (1998). On the economic value of modularity in software development: A real options approach. *International Journal on Applied Software Technologies*.

Boyle, P. and Lau, S. (1994). Bumping up against the barrier with the binomial method. *Journal of Derivatives*, 1:6–14.

Boyle, P. P., Evnine, J., and Gibbs, S. (1989). Numerical evaluation of multivariate contingent claims. *Review of Financial Studies*, 2:241–250.

Breen, R. (1991). The accelerated binomial option pricing model. *Journal of Financial and Quantitative Analysis*, 26(2):153–164.

Broadie, M. and Detemple, J. (1996). American option valuation: New bounds, approximations, and a comparison of existing methods. *The Review of Financial Studies*, 9(4):1211–1250.

Broadie, M., Glasserman, P., and Kou, S. (1999). Connecting discrete and continuous path-dependent options. *Finance and Stochastics*, 3(1):55–82.

Broadie, M., Glasserman, P., and Kou, S. G. (1997). A continuity correction for discrete barrier options. *Mathematical Finance*, 1(4):325–348.

Cai, Y., Huynh, S., and Xie, T. (2007). A framework and tool supports for testing modularity of software design. In *Proc. 22nd IEEE/ACM International Conference on Automated Software Engineering (ASE 2007)*, pages 441–444.

Carr, P. (1988). The valuation of sequential exchange opportunities. *Journal of Finance*, 43(5):1235–1256.

Chesney, M. and Elliott, R. (1995). Estimating the istantaneous volatility and covariance of risky assets. *Applied stochastic models and data analysis*, 11:51–58.

Chesney, M., Elliott, R., Madan, D., and Yang, H. (1993). Diffusion coefficient estimation and asset pricing when risk premia and sensitivities are time variyng. *Mathematical Finance*, 3(2):85–99.

Christoffersen, P. and Jacobs, K. (2004). Which garch model for option valuations. *Management Science*, 50:1204–1221.

Christoffersen, P., Jacobs, K., and Ornthanalai, C. (2008). Exploring time–varying jump intensities: Evidence from sp500 returns and options. Manuscript, McGill University.

Clark, P. (1973). A subordinated stochastic process model with finite variance for speculative prices. *Econometrica*, 41:135–155.

Comte, F. and Renault, E. (1998). Long memory in continuous time stochastic volatility models. *Mathematical Finance*, 8:291–323.

Copeland, T. E., Weston, J. F., and Shastri, K. (2005). *Financial Theory and Corporate Policy*. Pearson Addison Wesley, Readings, MA, IV edition.

Corsi, F. (2009). A simple approximate long-memory model of realized-volatility. *Journal of Financial Econometrics*, 7:174–196.

Corsi, F. and Renò, R. (2009). Volatility determinants: heterogeneity, leverage, and jumps. Working paper.

Costabile, M. (2002). Extending the cox-ross-rubinstein algorithm for pricing options with exponential boundaries. *Proceedings of Algoritmy Conference on Scientific Computing*, pages 23– 32.

Cox, J., Ross, A., and Rubinstein, M. (1979a). Option pricing: a simplified approach. *Journal of Financial Economics*, 7:229–263.

Cox, J. C., Ross, S. A., and Rubinstein, M. (1979b). Option pricing: a simplified approach. *Journal of Financial Economics*, 7:229–263.

Dixit, A. and Pindyck, R. (1994). *Investment Under Uncertainty*. Princeton University Press, Princeton, NJ.

Duan, J. C., Dudley, E., Gauthier, G., and Simonato, J. G. (2003). Pricing discretely monitored barrier options by a markov chain. *Journal of Derivatives*, 10(4):9–31.

Eppinger, S. (1991). Model-based approaches to managing concurrent engineering. *Journal of Engineering Design*, 2(4):283–290.

Eppinger, S., Whitney, D., Smith, R., and Gebala, D. (1994). A model-based method for organizing task in product development. *Research in Engineering Design*, 6(1):1–13.

Figlewski, S. and Gao, B. (1999). The adaptive mesh model: a new approach to efficient option pricing. *Journal of Financial Economics*, 53:313–351.

Gagliardini, P., Gouriéroux, C., Renault, E., Gallen, U. S., and für Nationalökonomie, F. (2005). Efficient derivative pricing by extended method of moments. Departement of Economics, University of St. Gallen.

Gamba, A. (2008). Real options valuation: A Monte Carlo approach. In Myers, S. and Sick, G. A., editors, *Real Options*, volume 2 of *Handbooks in Finance*. Elsevier North–Holland, Amsterdam, NL.

Gamba, A., Sick, G. A., and Aranda Leon, C. (2008). Investment under uncertainty, debt and taxes. *Economic Notes*, 37(1):31–58.

Gamba, A. and Tesser, M. (2009). Structural estimation of real options models. *Journal of Economic Dynamics & Control*, in press.

Gamba, A. and Trigeorgis, L. (2007). An improved binomial lattice method for multi-dimensional options. *Applied Mathematical Finance*, 14(5):453–475.

Geman, H. and Yor, M. (1996). Pricing and hedging double-barrier options: A probabilistic approach. *Mathematical Finance*, 6:365–378.

Geske, R. (1977). The valuation of corporate liabiliies as compound options. *Journal of Finance and Quantitative Analysis*, 12(4):541–552.

Geske, R. (1979). The valuation of compound options. *Journal of Financial Economics*, 7:63–81.

Geske, R. and Johnson, H. E. (1984). The American put option valued analytically. *Journal of Finance*, 39(5):1511–1524.

Glasserman, P. (2004). *Monte Carlo methods in financial engineering*. Springer-Verlag, New York - NY.

Gollier, C., Prould, D., Thais, F., and Walgenwitz, G. (2005). Choice of nuclear power investments under price uncertainty: Valuing modularity. *Energy Economics*, 27(4):667–685.

Gourieroux, C. and Jasiak, J. (2006). Autoregressive gamma process. *Journal of Forecasting*, 25:129–152.

Gourieroux, C. and Monfort, A. (2007). Econometric specification of stochastic discount factor models. *Journal of Econometrics*, 136(2):509–530.

Judd, K. L. (1998). *Numerical Methods in Economics*. MIT Press, Cambridge, MA.

Kamrad, B. and Ritchken, P. (1991). Multinomial approximating models for options with k state variables. *Management Science*, 37:1640–1652.

Kemna, A. G. Z. (1993). Case studies on real options. *Financial Management*, 22(3):259–270.

Kloeden, P. E. and Platen, E. (1999). *Numerical Solution of Stochastic Differential Equations*. Springer-Verlag, New York - NY, III edition.

Kogut, B. and Kulatilaka, N. (2001). Capabilities as real options. *Organization Science*, 12(6):744–758.

Kunitomo, N. and Ikeda, M. (1992). Pricing options with curved boundaries. *Mathematical Finance*, 2:275298.

Longstaff, F. A. and Schwartz, E. S. (2001). Valuing American Options by Simulation: a Simple Least-Squares Approach. *Review of Financial Studies*, 14(1):113–147.

Martzoukos, S. H. and Trigeorgis, L. (2002). Real (investment) options with multiple sources of rare events. *European Journal of Operational Research*, 136:696–706.

Merton, R. C. (1980). On estimating the expected return on the market: an exploratory investigation. *Journal of Financial Economics*, 8:323–61.

Moreno, M. and Navas, J. F. (2003). On the robustness of Least-Squares Monte Carlo (LSM) for pricing American derivatives. *Review of Derivative Research*, 6(2):107–128.

Pelsser, A. (2000). Pricing double barrier options using laplace transforms. *Finance and Stochastics*, 4:95–104.

Ritchken, P. (1995). On pricing barrier options. *Journal of Derivatives*, 3 (2):19–28.

Rodrigues, A. and Armada, M. J. R. (2007). The valuation of modular projects: A real options approach to the value of splitting. *Global Finance Journal*, 18(2):205–227.

Rudholm, N. (2007). Mergers and Economies of Scale: Volkswagen AG 1976-2000. *Icfai Journal of Mergers and Acquisitions*, 4(3):79–90.

Samuelson, P. A. (1965). Proof that properly anticipated prices fluctuate randomly. *Industrial Management Review*, 6(2):41–49.

Samuelson, P. A. (1973). Proof that properly discounted present values of assets vibrate randomly. *Bell Journal of Economics and Management Science*, 4(2):369–374.

Stentoft, L. (2004). Convergence of the Least Squares Monte Carlo Approach to American Option Valuation. *Management Science*, 50(9):1193–1203.

Stentoft, L. (2008). Option pricing using realized volatility. Working Paper at CREATES, University of Copenhagen.

Steward, D. V. (1981a). The design structure system: A method for managing the design of complex systems. *IEEE Transactions in Engineering Management*, 28(3):71–84.

Steward, D. V. (1981b). *System Analysis and Management: Structure, Strategy and Design*. Petrocelli Books, New York - NY.

Sullivan, W., Griswold, W. G., Cai, Y., and Hallen, B. (2001). The structure and value of modularity in software design. *ESEC/FSE*, pages 99–108.

Trigeorgis, L. (1993). The nature of option interactions and the valuation of investments with multiple real options. *Journal of Financial and Quantitative Analysis*, 28(1):1–20.

Trigeorgis, L. (1996). *Real Options: managerial flexibility and strategy in resource allocation*. MIT Press, Cambridge, MA.

Zhang, L., Aït-Sahalia, Y., and Mykland, P. A. (2005). A tale of two time scales: determining integrated volatility with noisy high frequency data. *Journal of the American Statistical Association*, 100:1394–1411.