

VICTORIA UNIVERSITY OF WELLINGTON

Valuation of a Higher Dimensional Optimal Switching Policy

by

Sam Kilkenny-Brown

Supervisor: Dr Toby DGLISH

Honours thesis submitted for the fulfillment of

FINA430

in the

School of Economics and Finance

October 2014

VICTORIA UNIVERSITY OF WELLINGTON

Abstract

School of Economics and Finance

FINA430

by **Sam Kilkenny-Brown**

Supervisor: Dr Toby Daghish

Real options problems frequently occur in land use decisions, however these problems are often of high dimension, since land has many uses. This work implements a Monte Carlo pricing method for land use real options. This is a challenging problem, due to the need to track multiple asset values, and manage option exercise decisions that are a function of multiple state variables.

Glossary

a_i	binary variable which defines production mode i as on/off
B_t	standard Brownian motion
$E^*[\dots]$	expected value
$f(x)$	profit per unit time when the system is in the state x
$J^{w_i}(x)$	expected value of pathway w_i relative to current state x
m	number of possible modes of production
n	number of stochastic state variables
p	current state price vector
P_t	vector of n underlying stochastic state variables
s	current time
t	time
T	termination time
w_i	i th pathway formed by a sequence of ordered pairs $w_i = (\theta(i); \eta(i))$
\tilde{w}	optimal pathway
W	set of pathways, such that $w_i \in W$
$x(s, p, z)$	defines current state
X_t	state of the whole stochastic system at time t

X_{θ_i-}	the limit of stochastic system X as time approaches stopping time θ_i
$X_t^{w_i}$	evaluation of stochastic system X_t with pathway w_i
z	vector that indicates production mode $z = (a_1, a_2, \dots, a_m)$
Z	set of all possible possible production indicator vectors $z \in Z$
Z_t	production mode indicator at time t
$\eta(i)$	ordered set of production modes, such that $\eta(i) \in Z$
$\theta(i)$	ordered set of stopping times where switching decision is made
$\mu(P_t, t)$	expectation of normally distributed change in stochastic process P_t
$\rho_{i,j}$	correlation matrix of state variables contained within P_t
$\sigma(P_t, t)^2$	variance of normally distributed change in stochastic process P_t
$\tilde{\phi}(x)$	value of optimum pathway given initial condition $x = (s, p, z)$

Chapter 1

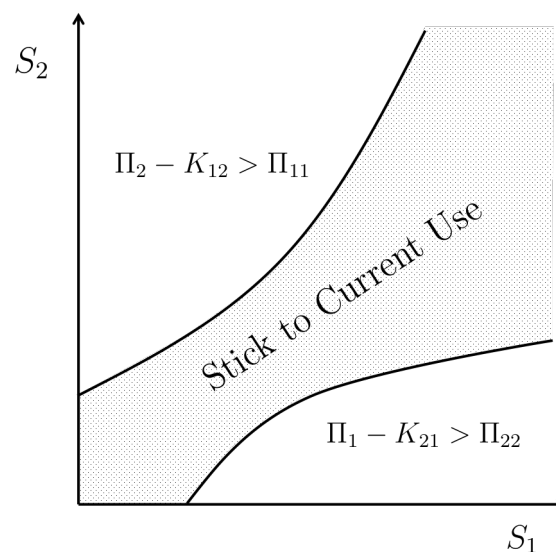
Introduction

Entry-exit problems arise frequently in real options analysis of land use. For example, a farmer may be trying to decide which of several crops/animals to grow/run on her land. In the farmer example, any decision can be reversed, however it may be costly as farming infrastructure may have to be added or removed. For example, animals may need to be liquidated or purchased, and a change of land ownership or management may be required. Currently, most theoretical discussions of this problem focus on a single choice. For example, [Song, Zhao, and Swinton \(2011\)](#) look at the switching between two crops with switching cost and reversibility. [Dixit and Pindyck \(2012\)](#) provide a close form solution for the valuation of two options with reversibility. However, these approaches are severely limited when considering the real world environment where there are many of options available for land use.

Song et al. (2011) use a real options analysis of the switching problem in two dimensions to inform an empirical tests of whether farmer obey the optimal policy. Furthermore, using the two-dimensional model they expand their analysis to look at the consequences of government policy intervention on switching costs. They find that government subsidization of land transition cost intended to move farmers from one production mode to another has unforeseen consequences, in that by reducing the cost of switching one way makes a system more dynamic.

Figure 1.1 illustrates a solution of the optimal conversion boundaries for a two-dimensional problem, where transition from state S_1 occurs if the indifference condition of continuation value minus cost of switching is met. As intuition would suggest the optimum boundary that dictates switching is more stringent than a net present value analysis would suggest because it includes the valuation of uncertainty in future returns and cost of switching (Dixit and Pindyck, 2012).

FIGURE 1.1: General Switching Boundaries



Currently, the Net Present Value(NPV) theory of investment is often used to assess

the switching problem. However, this theory does not allow for the implications of the interaction between uncertainty, irreversibility, and the choice of timing. In contrast to NPV theory, real options analysis has multiple benefits in that it allows us to capture the uncertainty of future returns, the sunk cost of switching that cannot be regained, as well as the value of reversible switching and the choice of timing (Dixit and Pindyck, 2012). The interactions between these characteristics are important to investigate in order for a landowner to make decisions that are consistent with optimal policy.

There are four basic computational methodologies for assessing real options: closed-form analytic solutions, numerical solutions to partial differential equations (PDEs), lattice models and simulation models. Closed-form solutions are the best approach but are generally not available for real options and lattice/PDE methods. While closed-form solutions are entirely competent, they suffer from exponential computational cost growth with dimensionality. We selected the Monte Carlo simulation method because it can easily handle multiple underlying processes and computational effort grows linearly with dimensionality (Broadie and Glasserman, 1997; Gamba, 2003).

The option to change land use exists in perpetuity and can be exercised at any time. Therefore, the option is akin to an American option. To value American options, Longstaff and Schwartz (2001) developed the least squares Monte Carlo method (LSM). The LSM method allows for the evaluation of the optimal policy through estimating the continuation value of the Bellman equation. Furthermore,

[Gamba \(2003\)](#) provides a method for extending the LSM approach to the real options problem with interaction between options, many state variables and optimal switching problems.

We implement a solution to the optimal switching problem using a Bellman equation and a value function approximated using basis functions. The Bellman equation is derived according to [Brekke and Øksendal \(1994\)](#) and we implement a Monte Carlo numerical methodology based on the [Longstaff and Schwartz \(2001\)](#) LSM method, expanded by [Gamba \(2003\)](#). We test the performance of the methodology in two-dimensions and three against the findings of [Longstaff and Schwartz \(2001\)](#) and [Moreno and Navas \(2003\)](#).

Following on from limitations present in current methods of appraising the switching problem, the motivation and aim of this research project is to develop a computational solution for switching between multiple land uses while taking into account reversibility and switching costs. To achieve this purpose, the paper consists of five sections. Section two presents a numerical example to demonstrate the application of real options analysis to the switching problem, before a numerical methodology for estimating the optimal exercise frontier is proposed in section three. Section four presents a performance test of the proposed methodology, presents test case results from the operational application in three dimensions, and suggests future extensions of the operational application. Finally, a brief summary of this study is offered in section five.

Chapter 2

Numerical Example

The purpose of this section is to illustrate the concepts underlying the estimation of the optimal switching policy using a numerical methodology. This is achieved by proposing and solving a simple numerical example.

Suppose there are two farmers, one farms pigs, the other corn. Over time, each farmer is able to switch between the production of pigs and corn, but in doing so incurs a cost of switching, denoted K . Assuming that each farmer maximises his/her return from the land, a farmer is faced with the problem of finding the optimal switching policy, given the costs of switching. Following the determination of the optimal switching policy, the land can be valued as the sum of the present value of all future cash-flows, minus switching costs. Lastly, we can find the value of the switching option by taking the difference between the value of the land under a no-switching policy and the value under the optimal switching policy.

For the purpose of presenting a clear example, we restrict the problem to five periods $t = (0, 1, 2, 3, 4)$, where the option to switch land use is exercisable at cost $K = 1$ at times $t = 0, 1, 2, 3$. For simplicity we consider a single simulation, with no discounting. The simulated period returns generated from each land use are illustrated in Table 2.1.

	$t = 0$	$t = 1$	$t = 2$	$t = 3$	$t = 4$
Corn	2	3	4	1	2
Pigs	2	1	3	4	5

TABLE 2.1: Simulated returns to land use

The problem with all possible switching policies can be visualised as a decision tree. The nodes represent times at which the switching decision can be made and the branches represent the cash-flows associated with the land use between switching times. A collection of branches form a policy, of all the possible policies there is one optimal policy. Conditional on farming corn or pigs at time $t = 0$ then the options available to each farmer for our proposed example are illustrated in Figure 2.1.

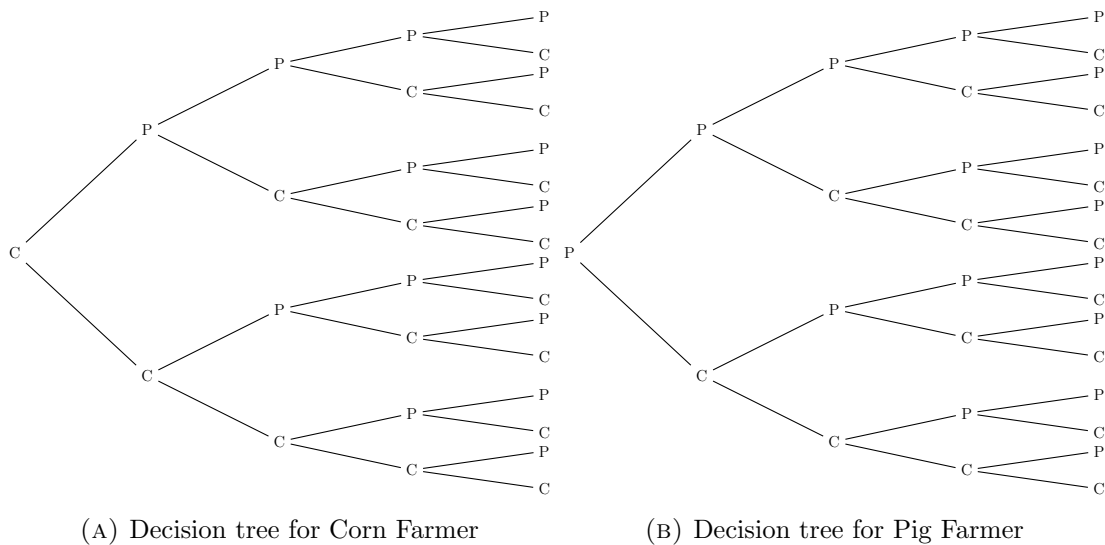


FIGURE 2.1: Available options for switching

In order to find the optimal pathway, we start at the final exercise date $t = 3$ and compare the value from exercising the option to switch to the value from continuing in the current state. Exercise will occur if switching is more valuable. Intuitively at $t = 3$ we can only form an expectation of the future $t = 4$ based on our current knowledge. However, for the purpose of this example we consider the future is known with certainty. Therefore, a decision rule is applied at time $t = 3$ based on the cash flows known at $t = 4$. The assessment of optimal cash-flow at $t = 4$ is shown by the solid red line in Figure 2.2, where the value of the optimal switching policy, given switching costs, and conditional on farming corn or pigs is denoted A .

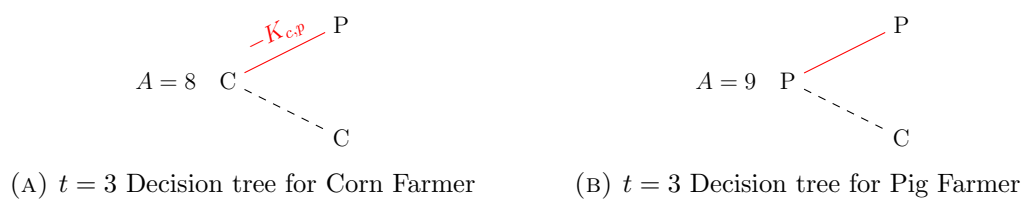


FIGURE 2.2: $t = 3$ Optimal Policy

Figure 2.2 illustrates that it is optimal for both farmers to be using the land for pigs and not corn despite the cost of switching from corn to pigs. Specifically, a farmer who enters $t = 3$ with pigs elects not to change, and a farmer who enters $t = 3$ with corn would exercise the option to switch under the optimal policy, with the expense of the switching cost and changes land use from corn to pigs.

In stepping back to $t = 2$, the optimal strategy is again found by comparing the value of continuation and the value from switching, conditional on what the land use is when the farmer entered $t = 2$. Figure 2.3 illustrates that it is again optimal for both farmers to be using the land for pigs and not corn. That is, the farmer

who enters $t = 2$ with pigs elects not to change and the farmer who enters $t = 2$ with corn exercises the option to switch (paying the switching cost) and changes production mode from corn to pigs. Note also that the optimal policy determined at $t = 4$ for the farmer who entered $t = 3$ with corn is now considered sub-optimal and is represented by a dashed red line in Figure 2.3.

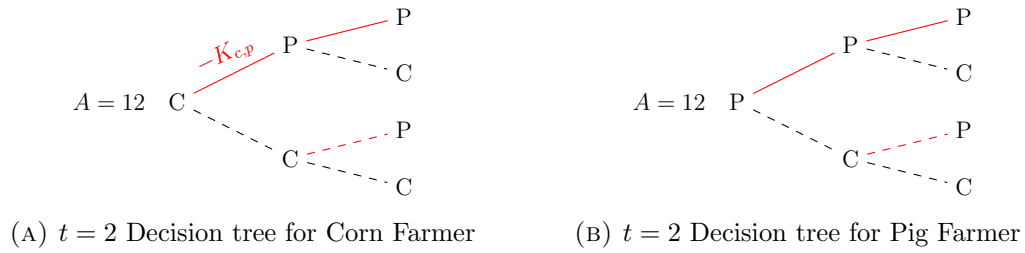


FIGURE 2.3: $t = 2$ Optimal Policy

Stepping back another time step to $t = 1$, the optimal strategy is illustrated in Figure 2.4. The decision rule is applied and the farmer who enters $t = 1$ with pigs is indifferent to switching or continuation, as the value of each is the same; therefore, the farmer elects continuation. In contrast, for the farmer who enters $t = 1$ farming corn it is optimal to remain producing corn, then to switch to pigs at $t = 3$. Furthermore, Figure 2.4 illustrates the sub-optimal pathways that have been discarded as the optimal policy is recursively formed. It is important to note that although all the alternative sub-optimal policies have been considered, we have only ever had to consider two possible decisions at each time: one if the farmer reaches that time and is producing corn, and one if he/she is producing pigs.

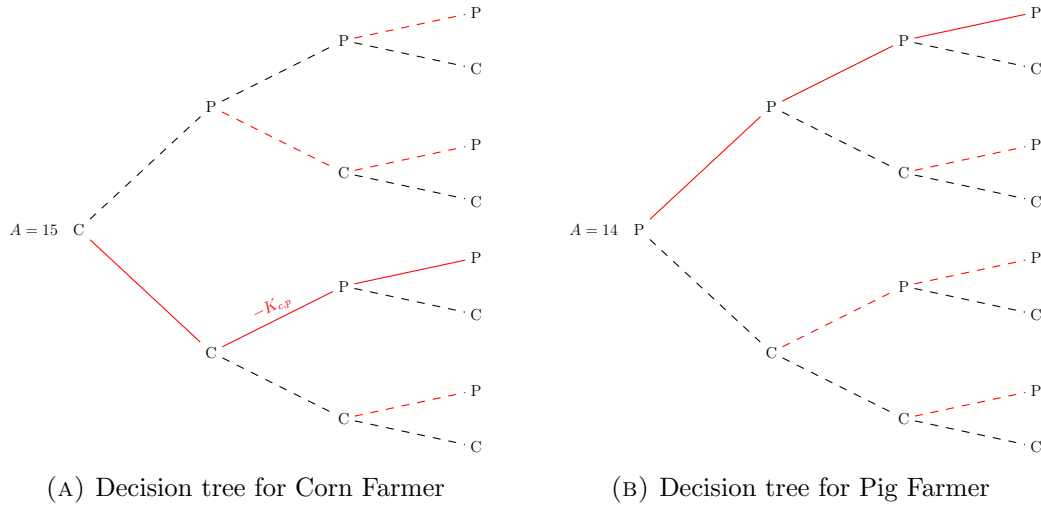


FIGURE 2.4: $t = 1$ Optimal policy

Stepping back to time $t=0$, the optimal switching policy is completed, as illustrated in Figure 2.5. Note that the farmer who enters at $t = 0$ farming pigs chooses to switch immediately to corn and switches back to pigs at $t = 2$.

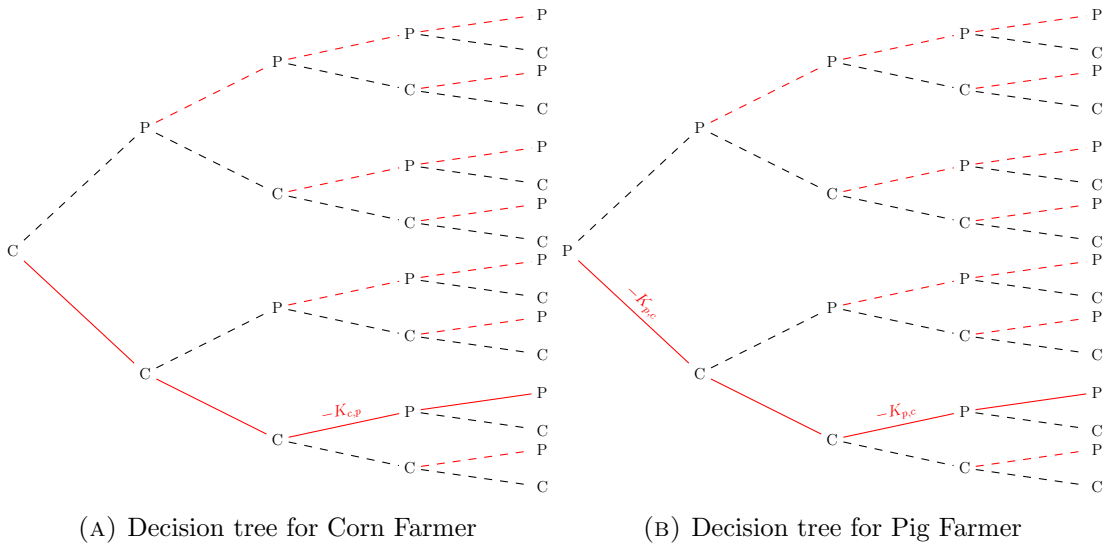


FIGURE 2.5: $t = 0$ Optimal Policy

The value of the optimal switching policy, given switching costs, and conditional on farming corn or pigs at $t = 0$ is recorded in Table 2.2. We consider the option to switch will add value to the land. For this example, the value of the optimal

switching policy is contrasted with the no-switching policy. Interestingly, for the farmer who enters $t = 0$ farming pigs there is little value in the switching option. However, for the corn farmer the value is substantial.

	No Switching	Switching, $K = 1$
A_c	12	17
A_p	15	16

TABLE 2.2: Valuation of No-Switching and Optimal Switching Policies

The problem illustrated in this section was simplified to only one simulation. However, in accordance with the law of large numbers, we need many simulations to make an adequate approximation of an expected value. Furthermore, in this example because we considered only one simulation path, we proposed that the farmer knew the future state prices with certainty. In reality decisions require the formation of expectations. [Longstaff and Schwartz \(2001\)](#) propose a methodology for forming an expectation that is based on regression of the present value of simulated cash-flows from $t + 1$ to T on a set of basis functions that are evaluated using current state prices. This methodology is expanded on in the following section.

Chapter 3

Methodology

This section proposes a methodology for solving the problem of finding the optimal switching policy in a multi-production mode environment, given the costs of switching and assuming that the state of the economic system is a stochastic process. Firstly, we discuss the underlying state variables. Subsequently, the switching problem is formulated as an extended impulse control problem as originally proposed by [Brekke and Øksendal \(1994\)](#). Lastly, the problem is solved using recursive Monte Carlo simulation as initially proposed by [Longstaff and Schwartz \(2001\)](#) and expanded for the switching problem by [Gamba \(2003\)](#).

3.1 Modelling The Economic System

At any point in time t the world can be represented as a system of n underlying state variables (commodity prices) P

$$P_t = (P_{t,1}, P_{t,2}, \dots, P_{t,n}) \quad (3.1)$$

We assume that P_t is a continuous time stochastic process that evolves according to the following stochastic differential equation

$$dP_{i,t} = \mu_i(P_t, t)dt + \sum_j \sigma_{i,j}(P_t, t)dB_{j,t} \quad (3.2)$$

where we understand the stochastic process $P_{i,t}$ changes in accordance with a normal distribution, with expectation $\mu_i(P_t, t)$, variance $\sum_j \sigma_{i,j}^2(P_t, t)$ and $B_{j,t}$ is a Brownian motion. Lastly, it is important when simulating multiple state variables that correlations of components are included. Without loss of generality, Brownian motions are independent and correlations between components of the vector, P_t , can be induced through the matrix $\sum_j \sigma_{i,j}(P_t, t)$.¹

The state of the economic system for crop production z , at time t can be represented by the stochastic process

$$X_t = [P_t, t, z] \quad (3.3)$$

¹ $\text{Var}(dP) = G'G$, where $G_{i,j} = \sigma_{i,j}$, so G is Cholesky Decomposition of variance-covariance matrix.

The behaviour of a stochastic process is governed by the specified functional form of the stochastic differential equation. There are several common functional forms that are well understood for their applicability to economics such as the Arithmetic Brownian motion, Lognormal Diffusion (Geometric Brownian motion), Ornstein-Uhlenbeck Mean Reverting process, and the Cox-Ingersoll-Ross model. We briefly expand on the financial application of Lognormal Diffusion and Ornstein-Uhlenbeck(OU) processes. Refer to [Hull \(2009\)](#) for further discussion.

Lognormal Diffusion has the property that the process cannot pass through zero(provided the initial state is positive). Therefore, it is useful for modeling for stock prices because it fits the assumption of limited liability. It has the following functional form

$$\mu_i(P_t, t) = \mu_i P_t \quad \text{and} \quad \Sigma_j \sigma_{i,j}(P_t, t) = \Sigma_j \sigma_{i,j}(P_t, t) P_t. \quad (3.4)$$

The OU Mean Reverting process, has the property that the random variable will revert to a long term stationary average. Therefore, it is useful for modelling processes such as commodity prices. It has the following functional form

$$\mu(P_{t,i}, t) = \lambda(\mu_{P_i} - P_{t,i}) \quad \text{where} \quad \lambda > 0, \quad \text{and} \quad \sigma(P_{t,i}) = \sigma_{P_i} \quad (3.5)$$

In the context of land use, theoretically both processes could be justified in describing how returns evolve over time. For example, a lognormal process can better represent a trend that could be positive due to technological advances that boost productivity Alternatively, an OU process can better reflect long-term equilibrium conditions ([Song et al., 2011](#)). In contrast to the first example, this could be where

technology is not changing. The specific functional form of each state variable is a matter of inference. Simulation using Brownian motion also allows the flexibility to collate multiple functional forms. For example, in the agricultural economy production modes such as wool or produce will be considered commodity like and therefore a mean-reverting process is useful for modeling prices.² In contrast, there may be a production mode such as dairy farming which is not well explained by a mean reverting process and better explained by Lognormal Diffusion.

3.2 Formulation of Switching Problem

Suppose there are multiple possible production modes in the economy, and given the ability to switch modes through time, a series of modes will concatenate into a pathway. Therefore, the value of a specific pathway can be determined by taking the present value of cash flow net of switching costs along the path. Fundamentally, there are numerous combinations of modes that can be concatenated into possible pathways. Then through multiple simulations, the law of large numbers will dominate and allow for determination of expected value. The purpose of this section is to formulate a mathematical foundation for establishing the possible pathways such that an optimal pathway can then be found.

Consider a piece of land that can be used for M different production modes, where only one can be on at a time. We can define any possible state of the world using

²The trouble with the OU process is that it can become negative. This would be an acceptable trait if we were modelling profitability as negative state prices would translate to crop/land returns making a loss, but still in operation.

an integer z_m denoting the land's current use such that

$$z \in (1, 2, \dots, M) \quad (3.6)$$

and let the state of the world Z_t at any time t be defined

$$Z_t \in \{z_1, \dots, z_m\}. \quad (3.7)$$

Secondly, we take the modes of operation and use them to specify a pathway through time. Such a pathway is defined by two elements 1) the time spent in each mode, and 2) the sequence of occupation.

Therefore, we suppose there are certain 'stopping times' denoted θ when a land owner makes a switching decision. At any stopping time θ , there exists a production mode η that corresponds to the crop or mode of production taking place on the land z , such that at any time t

$$\eta_\theta = Z_t \in Z \quad \text{for} \quad \theta = t. \quad (3.8)$$

Let the stopping times θ and associated production modes η be collated through time to define a pathway w , which can be represented as a double sequence

$$w = (\theta_1, \theta_2, \dots; \eta_1, \eta_2, \dots). \quad (3.9)$$

With many possible combinations/sequences of operation, suppose that W represents a set of all possible pathways w_1, w_2, \dots

$$W = \{w_1, w_2 \dots\}. \quad (3.10)$$

If $w \in W$ is applied to the the system $X_t = [P_t, t]$, it takes the form

$$X_t = X_t^{(w)} = [t; P_t; \eta_i] \quad \text{if } \theta_i \leq t < \theta_{i+1}, \quad (3.11)$$

which is to say, the production mode remains the same until the next switching decision θ_{i+1} , while the state variable P_t fluctuates randomly.

Having developed an expression for the possible pathways available to the land owner through time, we now consider the optimal switching problem given costs.

Let E^* stand for the expected value, conditional to the current state $X_0 = x = (s, p, z)$. Secondly, define profit per time unit when the system is in the state x as $\Pi(x)$. Lastly, let $K(x, \eta)$ be the cost of switching from the production mode z to η when the state is $x = (s, p, z)$, such that the expected value from selecting pathway $w_i \in W$ is given by

$$J^{w_i}(x) = E^* \left[\int_s^\infty e^{-r(t-s)} \Pi(X_t^{w_i}) dt - \sum_{j=1}^\infty e^{-r(\theta_j-s)} K(Z_{\theta_j, \eta_j}) \right]. \quad (3.12)$$

We assume that switching cost satisfies:

$$K(z, \eta) > 0 \quad \text{for all } \eta \neq z \quad (3.13)$$

and if Z contains more than 2 elements, we assume there is a *no free lunch* condition such that

$$K(z_1, z_2) \leq K(z_1, z_3) + K(z_3, z_2) \text{ if } z_1 \neq z_3 \neq z_2 \neq z_1 \quad (3.14)$$

Furthermore, as there is no switching where $z = \eta$, the cost of switching $K(z, z)$ is zero and we only need to define $K(z, \eta)$ for $z \neq \eta$. Note that irreversibility from the current mode can be accommodated by specification of a disproportionate switching cost such that $K(z, \eta)$ is so large that it is irrational to consider switching. Finally, given the current state $x = (s, p, z)$, the optimal switching problem is one of finding the pathway $\tilde{w} \in W$ such that the expected discounted cash flow from operations net of switching costs is maximised

$$\tilde{\phi}(x) = J^{\tilde{w}}(x) := \sup_{w \in W} J^w(x). \quad (3.15)$$

Equation (3.15) represents the multi-period problem that must be evaluated to find the value of the optimal switching policy. For the purposes of implementing a numerical solution, we introduce a Bellman equation, that reduces the problem into a sequence of single period decisions. Therefore, for all stopping times τ conditional on being in mode z at time t the Bellman equation is:

$$\tilde{\phi}_z(x) \geq E^* \left[\int_s^\tau e^{-r(t-s)} \Pi_z(X_t) dt - e^{-r(\tau-s)} K_{z, \eta(\tau)} + \tilde{\phi}_{\eta(\tau)}(X_\tau) \right] \quad (3.16)$$

3.3 Solving the Switching Problem

This section proposes a numerical solution for approximating the valuation of the switching problem by simulation. Simulation methods have been widely researched and a number of methods exist for approximation by simulation including those by [Glasserman \(2004\)](#), [Boyle, Broadie, and Glasserman \(1997\)](#) as well as [Longstaff and Schwartz \(2001\)](#). The [Longstaff and Schwartz \(2001\)](#) Monte Carlo method of least squares (LSM) is used in this paper. LSM was proposed by the authors as a simulation based solution to the valuation of financial derivatives that require estimation of the optimal exercise frontier. For example, an American type option has an optimal exercise frontier because it can be exercised at any time during its life. At the core, LSM is a recursive method that uses least squares to estimate the value function of the option holder from continuation. With respect to the switching problem in the context of LSM, [Gamba \(2003\)](#) proposes an extension to the LSM method. The following sub-sections first introduce the fundamental LSM and second extend LSM to the switching problem.

3.3.1 Monte Carlo Simulation for Real Options

[Longstaff and Schwartz \(2001\)](#) propose a simulation based method to determine the optimum exercise frontier of options. This section presents the LSM methodology for a simple American option and explains how it is used to solve for the optimum exercise boundary and hence value the option.

Consider an American option that gives a contingent claim on X that expires at T . Monte Carlo simulation is used to generate J paths for X . Where, $X_t(u_j)$ represents the value of the process at time t along the u_j simulated path. Suppose we then divide the time span $[0, T]$ into N intervals of $\delta t = T/N$ and let $\tilde{\theta}(u_j)$ denote the optimal policy along simulated path u_j . Given that we have restricted the problem to a set of N dates, it follows that the dates at which the farmer can switch are restricted. Therefore, the optimal switching policy for simulation path j is restricted to the set of possible switching times: $\tilde{\theta}(u_j) \in (0 < t_1 \leq t_2 \leq \dots t_N = T)$.

The purpose of the LSM algorithm is to find the optimal exercise time for each simulation with the restriction of discrete dates. The optimal policy conditional on initial state $X = (s, p, z)$ is found through dynamic programming. Working backwards through time, the optimal decision is made by comparing the pay-off, $\Pi(t_n, X_{t_n}(u_j))$, with the value function, $\phi(t_n, X_{t_n}(u_j))$, such that we can define the recursive decision rule as

$$\phi(t_n, X_{t_n}(u_j)) \geq \Pi(t_n, X_{t_n}(u_j)) \quad (3.17)$$

as a result of the decision being optimal to exercise, the optimal stopping time is updated $\tilde{\theta}(u_j) = t_n$. Alternatively, if it is not optimal to exercise, then the stopping time variable remains unchanged. A Bellman equation is used to split the continuous period problem into one with respect to the current period and the

next. Therefore, providing a solution to $\phi(t_n, X_{t_n}(u_j))$ in discrete time

$$\phi(t_n, X_{t_n}(u_j)) = \max \left\{ \Pi(t_n, X_{t_n}), e^{-r(t_{n+1}-t_n)} E_{t_n}^* [\phi(t_{n+1}, X_{t_{n+1}})] \right\}. \quad (3.18)$$

This decision rule requires an estimate of the derivative's continuation payoff. Reality dictates that at time t_n we cannot have perfect foresight of the forthcoming solution. Therefore we take expectations, denoting the continuation value

$$\Phi(t_n, X_{t_n}) = e^{-r(t_{n+1}-t_n)} E_{t_n}^* [\phi(t_{n+1}, X_{t_{n+1}})], \quad \Phi(T, X_T) = 0 \quad (3.19)$$

The decision rule compares Φ with pay-off, Π . We work recursively back through time from T , updating the stopping time along simulation u_j . If we already have recorded an optimal exercise time $\theta(u_j)$ then it is overwritten by the new optimal time $\tilde{\theta}(u_j)$. Ultimately, once the recursion has working backwards through time to the initial condition, the value of the option is estimated by averaging the discounted value of the optimal pay-off across all simulations.

$$\phi(0, x) = \frac{1}{N} \sum_{j=1}^J e^{-r\tilde{\theta}_{u_j}} \Pi(\tilde{\theta}_{u_j}, X_{\tilde{\theta}_{u_j}}(u_j)) \quad (3.20)$$

At time t_n , the value of continuation can be expressed as

$$\Phi(t_n, X_{t_n}) = E^* \left[\sum_{i=n+1}^N e^{-r(t_i-t_n)} \Pi(u, t_i; t_n, T) \right] \quad (3.21)$$

We form an approximation of the conditional expectation function at time t_n using least squares regression onto a set of basis functions $L_i(X)$, such that

$$\tilde{\Phi}(u; t_n) = \sum_{i=0}^N a_i L_i(X) \quad N \rightarrow \infty, \quad (3.22)$$

where the a_i represent the regression constants. This process is stepped back through time recursively and at each step the decision rule is evaluated and recorded.

The LSM method involves two approximations: firstly, the Monte Carlo simulations are used to estimate the conditional expectation function, and secondly, the conditional expectation in value equation is replaced by its orthogonal projection using a finite set of basis functions. Therefore, the accuracy of the LSM method is determined by the number of simulations and choice of basis functions and consequently, each must be selected carefully. [Longstaff and Schwartz \(2001\)](#) and [Moreno and Navas \(2003\)](#) use the LSM method to value a multi-dimensional derivative and compare their findings to the valuation of the same option found by [Broadie and Glasserman \(1997\)](#) using the stochastic mesh method. They find that with reasonable approximation parameters³ the option value estimated is within the 90% confidence interval established by [Broadie and Glasserman \(1997\)](#). Furthermore, [Moreno and Navas \(2003\)](#) found that a minimum of 5 polynomial terms were required to obtain results using LSM that are robust to the [Broadie](#)

³[Longstaff and Schwartz \(2001\)](#) and [Moreno and Navas \(2003\)](#) both test the LSM method on a 5-dimensional call option on the maximum of five uncorrelated assets using 50,000 simulations and the following 19 basis functions: a constant, the first five Hermite $H_n(x)$ polynomials in the maximum of the five assets, the second to fifth maximums and their square values, the four products of consecutive pairs of maximums, the product of the five assets

and Glasserman (1997) 90% confidence interval. Lastly, Clément, Lamberton, and Protter (2002) prove convergence of LSM as the number of simulated paths and basis functions tend toward infinity.

3.3.2 The Optimal Switching Problem

Following the methodology outlined in Gamba (2003), we extend the Broadie and Glasserman (1997) LSM approach to the optimal switching problem.

Given the Bellman Equation (3.16), consider a finite time horizon $[0, T]$, and simulation path denoted u . The value of the land equals the maximized value of the discounted cash flows, where the maximum is taken over all stopping times with respect to the system X . Therefore, along simulation u , we wish to find the optimal stopping times $\tilde{w}(u) = (\tilde{\theta}(u); \tilde{\eta}(\tilde{\theta}, u))$.

Propose that there are M different land uses in the given system, with current land use denoted z and the set of all possible production modes $Z = (z_1, z_2, \dots, z_M)$. Furthermore, let the cost of switching from z to z_m be $K_{z,z_m}(t, X_t) = K(t, X_t, z, z_m)$ and, let $\Pi_z(t, X_t) = \Pi(t, X_t, z)$ denote the payoff from being in mode z at time t .

We define the net payoff when transitioning from z to any mode $z_m \in Z$

$$\Pi'_{z,z_m}(t_n, X_{t_n}) = \int_{t_n}^{t_{n+1}} \Pi_{z_m}(t, X_t) e^{-r(t_{n+1}-t)} dt - K_{z,z_m}(t_n, X_{t_n}); \quad (3.23)$$

we then redefine the Bellman equation as follows:

$$\tilde{\phi}_z(t_n, X_{t_n}) = \max_{z_m \in Z} \left\{ \Pi'_{z, z_m}(t_n, X_{t_n}) + e^{-r(t_{n+1}-t_n)} E_{t_n}^* [\tilde{\phi}_{z_m}(t_{n+1}, X_{t_{n+1}})] \right\}. \quad (3.24)$$

The farmer switches from current mode, z to z_m if the payoff plus the continuation value from mode z is lower than the payoff plus continuation value from mode z_m ; otherwise he/she remains in mode z . Subsequently, in the context of dynamic programming, and placing the problem into an LSM setting, we can define the continuation value for mode z_m

$$\tilde{\Phi}_{z_m}(t_n, X_{t_n}) = e^{-r(t_{n+1}-t_n)} E_{t_n}^* [\tilde{\phi}_{z_m}(t_{n+1}, X_{t_{n+1}})]. \quad (3.25)$$

Hence along simulation path u , the decision rule for switching mode is that if

$$\Pi'_{z, z}(t_n, X_{t_n}(u)) + \tilde{\Phi}_z(t_n, X_{t_n}(u)) < \max_{z_m \in Z} \left\{ \Pi'_{z, z_m}(t_n, X_{t_n}(u)) + \tilde{\Phi}_{z_m}(t_n, X_{t_n}(u)) \right\} \quad (3.26)$$

then a switch takes place at time $\tau = t_n \in \tilde{\theta}(u)$ along path u , and the optimum operating mode is recorded as:

$$\eta(\tau, u) = \operatorname{argmax}_{z_m \in Z} \left\{ \Pi'_{z, z_m}(t_n, X_{t_n}(u)) + \tilde{\Phi}_{z_m}(t_n, X_{t_n}(u)) \right\} \quad (3.27)$$

otherwise, $t_k \neq \tau \notin \tilde{\theta}(u)$.

We define the path of cash-flows $A_z(u, s; t, T)$ generated by land use conditional on the landowner following the optimal stopping strategy for all times $s, t < s \leq T$

$$A_z(u, s; t_n, T) = \begin{cases} \text{if } t_n \notin \tilde{\theta}(u) & \begin{cases} A_z(u, s; t, T) & \text{for } s = t_{n+1}, t_{n+2}, \dots, t_N \\ \Pi'_{z,z}(t_n, X_{t_n}(u)) & \text{for } s = t_n \end{cases} \\ \text{if } t_n \in \tilde{\theta}(u) & \begin{cases} A_\eta(u, s; t, T) & \text{for } s = t_{n+1}, t_{n+2}, \dots, t_N \\ \Pi'_{z, \eta_{\tilde{\theta}(u)}}(t_n, X_{t_n}(u)) & \text{for } s = t_n \end{cases} \end{cases} \quad (3.28)$$

If time t_n is a switching time $\tau \in \tilde{\theta}(u)$, and the new production mode is $\eta_{\tilde{\theta}(u)}$, then the current optimal net cash flow coincides with $\Pi'_{z, \eta_{\tilde{\theta}(u)}}$ and the optimal cash flow for $s, t < s \leq T$ along the same simulation is determined in previous steps of the algorithm by backward induction. Alternatively, if t_n is not a switching time, we stay in the current mode z receiving the optimal cash flow $\Pi'_{z,z}$, and for $s, t < s \leq T$ along the same path we get the optimal cash flow determined in previous steps by backward recursion. Therefore, the continuation value in mode z_m is the expectation, conditional on being at (t_n, X_{t_n}) , of optimal discounted cash flows:

$$\tilde{\Phi}_{z_m}(t_n, X_{t_n}) = E_{t_n}^* \left[\sum_{i=n+1}^N e^{-r(t_i - t_n)} A(u, t_i; t_n, T) \right] \quad (3.29)$$

Chapter 4

Results

This section initially quantifies the accuracy of the proposed methodology and then goes on to present test case results from the operational application in three-dimensions. All analyses were implemented in Matlab and simulations were run using a 1.4 GHZ Intel Core i5 processor.

The Least Squares Monte Carlo method makes two approximations that affect the accuracy of the estimate. Firstly, the conditional expectation in value equation is replaced by its orthogonal projection using a finite set of basis functions. Secondly, the Monte Carlo simulations and least-squares regressions that estimate the conditional expectation function have a finite number of paths. We quantify what the effect on accuracy is from making these approximations by varying the number of simulations and basis functions on three test cases.

Subsequently, having established reasonable choices for the family of basis functions, number of polynomials, and number of simulations we present an application

of the methodology. In applying the methodology we find and discuss the optimal switching boundaries for a three dimensional problem.

4.1 Accuracy of LSM

The accuracy of the LSM method is determined by the choice of three approximation parameters: the family of basis functions, the number of polynomials, and number of simulations. We analyse the relationship between these choices, computational time, and accuracy.

To this end, we test the accuracy and computational time of the methodology by establishing three test cases. The first model is a two-dimensional no-switching model. A no-switching model is selected because it allows the opportunity to compare the estimated valuation to the Gordon growth model (Gordon and Shapiro, 1956). The Gordon growth model is a closed form model that can be used to determine the value of a perpetuity, given a constant growth rate and constant cost of capital. The parameters of the underlying stochastic processes are summarised in Table 4.1.

TABLE 4.1: Two dimensional test problem input parameters

	Returns to Corn		Returns to Pigs	
Initial Return	$S_{t=0,c}$	100	$S_{t=0,p}$	100
Drift Parameter	μ_c	0.04	μ_p	0.04
Variance Parameter	σ_c	0.20	σ_p	0.40
Correlation Parameter	$\rho_{c,p}$	0.40		
Discount rate	r	0.08		

Land conversion costs: corn to pigs $K_{c,p} : \infty$; pigs to corn $K_{p,c} : \infty$, terminal time $T=200$ years

The second model is a two-dimensional model *with* switching. The two dimensional switching model is included as a minimum benchmark for the Monte Carlo method. The parameters from Table 4.1 are reused with the exception that land conversion costs are included. Specifically, switching from corn to pigs incurs $K_{c,p} = 200$; and pigs to corn $K_{p,c} = 200$.

The last model is a three-dimensional model with switching. We expect that there is a negative correlation between accuracy and higher dimensionality. Therefore, we select a three dimensional model to allow us to estimate the decline in accuracy as dimension increases. The parameters for the three-dimensional problem are summarised in Table 4.3.

TABLE 4.2: Three dimensional test problem input parameters

	Returns to Corn		Returns to Pigs		Returns to Grapes	
Initial Returns	$S_{t=0,c}$	100	$S_{t=0,p}$	100	$S_{t=0,g}$	100
Drift Parameter	μ_c	0.04	μ_p	0.04	μ_g	0.04
Variance Parameter	σ_c	0.20	σ_p	0.40	σ_g	0.60
Discount rate	r	0.08				

Assume correlation between all state variables is $\rho_{i,j} = 0.4$, and land conversion costs are: $K_{i,j} = 200$;

Initially, to observe the effect of simulations on accuracy in isolation, we select a set of basis functions that consist of a constant, the first ten Laguerre polynomials in each asset and the cross-product of each asset. This gives us a total of 30/50 basis functions for the two/three dimensional test cases respectively. We measure accuracy of the corn farming land value, while increasing simulations from 1,000 through to 1,000,000. For the no-switching model we use the true value ascertained from the Gordon growth model to measure the relative accuracy of the methodology. We do not know the true value for the second and third test cases; therefore

we assume that the valuation obtained using 1,000,000 simulations should be close enough to the true value to allow us to measure the accuracy of smaller numbers of simulations. Figures 4.1 and 4.2 present the results of our analysis.

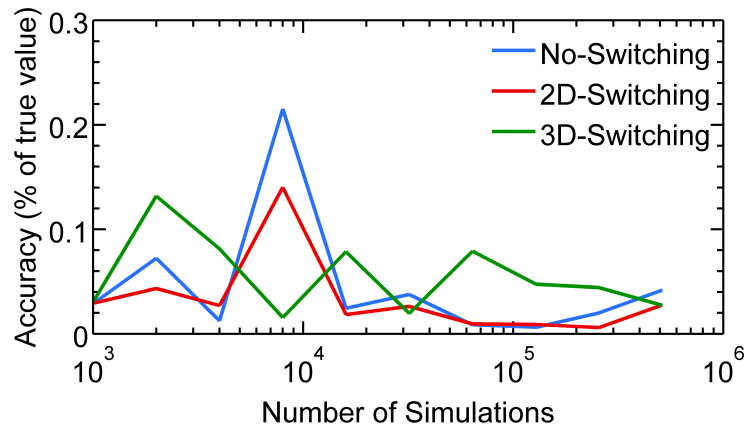


FIGURE 4.1: Relationship between number of simulations and accuracy

Figure 4.1 illustrates that the valuation methodology produces oscillatory convergence with an increasing number of simulations. There is not a significant loss in latency between 2 and 3 dimensions. We expect from this analysis that a choice of 50,000-100,000 simulations is sufficient to convincingly produce a valuation with accuracy of plus or minus 10% of the true value. This is in accordance with the findings of [Longstaff and Schwartz \(2001\)](#) and [Moreno and Navas \(2003\)](#) who use 50,000 and 100,000 simulations respectively to price a 5 dimensional Bermudian call option.

Figure 4.2 presents accuracy with respect to the computational time. It is observed that with greater accuracy comes greater computational cost. The proportionality is log-linear, and therefore it is desirable to keep the number of simulations as low as possible while retaining the desired accuracy. It is considered that computation

of 50,000/100,000 simulations will yield adequate accuracy without disproportionate computational time. Secondly, to observe the effect of basis function selection

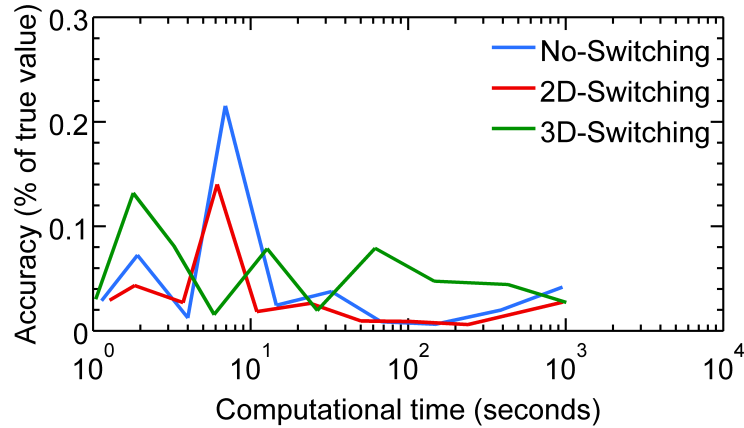


FIGURE 4.2: Relationship of computational time taken to accuracy

on accuracy in isolation we select 100,000 simulations and increase the number of basis functions. Additionally, we contrast the application of Laguerre and Hermite families of orthogonal polynomials.¹ The number basis functions increase by incrementally including higher order terms of the polynomial family being evaluated. Each set of polynomials is evaluated for each state price and the cross-product of each state price. Lastly, we continue with the same accuracy measure that was used to compare the choice of simulations. Figures 4.3 and 4.4 present the results of our analysis.

Figures 4.3 and 4.4 illustrate that the selection of basis functions does not have a significant effect on accuracy when considered relative to choice of number of simulations. These figures also demonstrate that there is no significant difference between the two families of polynomials, and that the measure of accuracy becomes increasingly volatile with higher numbers of basis functions. These findings are

¹For a discussion of these functions and other families of orthogonal polynomials refer to Chapter 22 of [Abramowitz and Stegun \(1972\)](#)

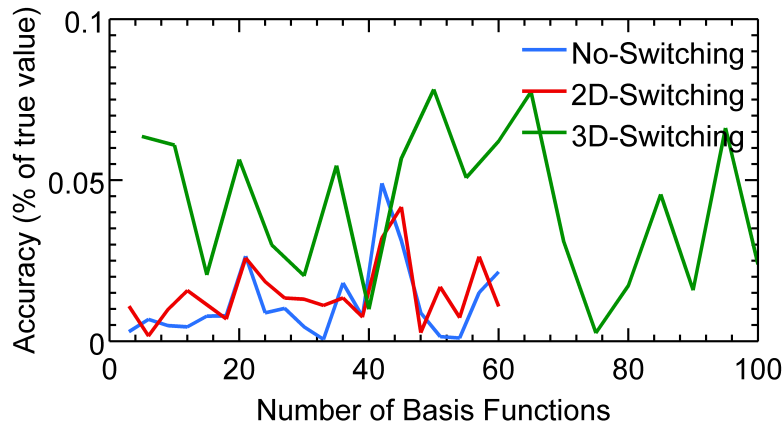


FIGURE 4.3: Relationship between number of basis functions from the Laguerre polynomial family and accuracy

consistent with [Moreno and Navas \(2003\)](#) who found that the choice of basis functions only made a slight difference on valuation, and that when the number of polynomial terms is large (20 or more), numerical problems can appear.

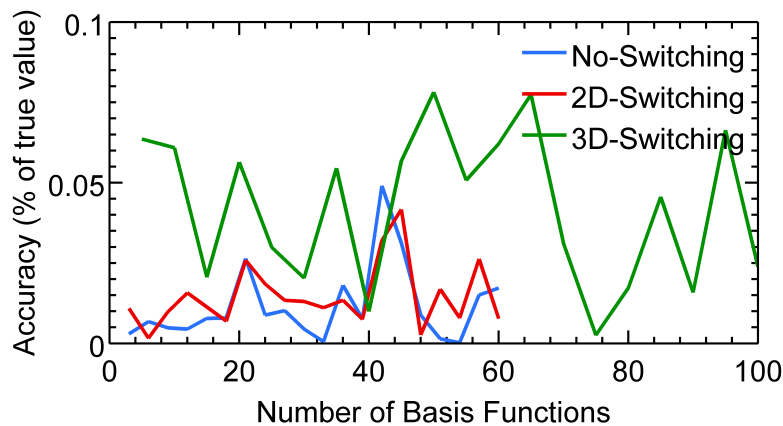


FIGURE 4.4: Relationship of computational time taken versus accuracy

Lastly, with regard to dimensionality and the use of basis functions, Figures 4.3 and 4.4 indicate a difference in accuracy when contrasting two and three state variable switching models. However, it appears that there exists no proportionality between the number of basis functions, or the family of basis functions that has any significance in determining a more accurate valuation.

4.1.1 Three Mode Problem

The purpose of this section is to present our extension of the switching problem to multiple dimensions. We extend the type of two-dimensional analysis presented by Song et al. (2011), Dixit and Pindyck (2012), Luong and Tauer (2006) and Schmit, Luo, and Conrad (2011) to a three dimensional problem, using the proposed numerical methodology to determine the optimal switching boundaries. The parameters of the underlying stochastic processes used for analysis are summarised in Table 4.3.

TABLE 4.3: Three dimensional test problem input parameters

	Returns to Corn		Returns to Pigs		Returns to Grapes	
Drift Parameter	μ_c	0.04	μ_p	0.04	μ_g	0.04
Variance Parameter	σ_c	0.20	σ_p	0.40	σ_g	0.60
Discount rate	r	0.08				

Assume correlation between all state variables is $\rho_{i,j} = 0.4$, land conversion costs are given in Table 4.4, and the terminal time is $T=200$ years. The data set used for finding the switching boundaries is created by establishing a three-dimensional grid of initial return conditions. Each point on the grid is valued in accordance with the proposed methodology, conditional on the farmer entering at $t = 0$ in each state of land use. We use 100,000 simulations, and choose 20 basis functions consisting of: a constant, the first three Laguerre polynomials in each state variable, the first two Laguerre polynomials in the first, second and third highest of the state variables, the lagged product of the state variables and the lagged product of the maximum of the state variables.

We use a real options analysis in order to capture the uncertainty of future returns, the sunk cost of switching that cannot be regained, the value of reversible switching and the choice of timing. Therefore, transition from one state to another will only occur if the returns from an alternative state are sufficiently high that it is optimal to switch. Due to sunk costs and uncertainty the transition boundary will exist below a 45 degree line. How far below the 45 degree line is determined by the cost and uncertainty parameters of valuation. To observe these characteristics in

three dimensions we have specifically selected the test case parameters presented in Tables 4.3 and 4.4 to create structured variance in the cost and uncertainty of each state. For example, the volatility of returns is lowest when growing corn and highest when growing grapes. Therefore, if a farmer is considering a switch to corn, his/her expectation of uncertainty in future returns will be less than if the farmer was to switch from corn to grapes. Subsequently, the farmer needs greater compensation by way of returns before leaving growing corn to grapes. In contrast, the farmer growing grapes needs relatively less compensation to move to corn.

TABLE 4.4: Switching costs between different modes of production

	Corn	Pigs	Grapes
Corn	0	150	200
Pigs	100	0	200
Grapes	100	150	0

Additionally, the cost to switch between different modes of operation has been specified for effect. It is specified that corn is the cheapest mode to change to, followed by pigs and lastly, grapes is most expensive. The option to switch therefore is most valuable to farmers growing grapes because he/she can exercise the option to transition at a far lower cost. Consequently, the corn farmer is least likely to switch due to the relatively higher cost of transition.

We present the results of the test case in Figures 4.5, 4.6, and 4.7. In order to present the data in two-dimensions we have taken three slices of the three dimensional interaction profile: the first slice where the current state returns are high, $S_{0,mode} = 300$, second where the state returns are middle range, $S_{0,mode} = 150$, and last where the state returns are low, $S_{0,mode} = 75$. Each sub-figure

represents the optimal switching boundaries for each mode of production: corn, pigs and grapes. Figure 4.5 illustrates that when the returns for a current mode

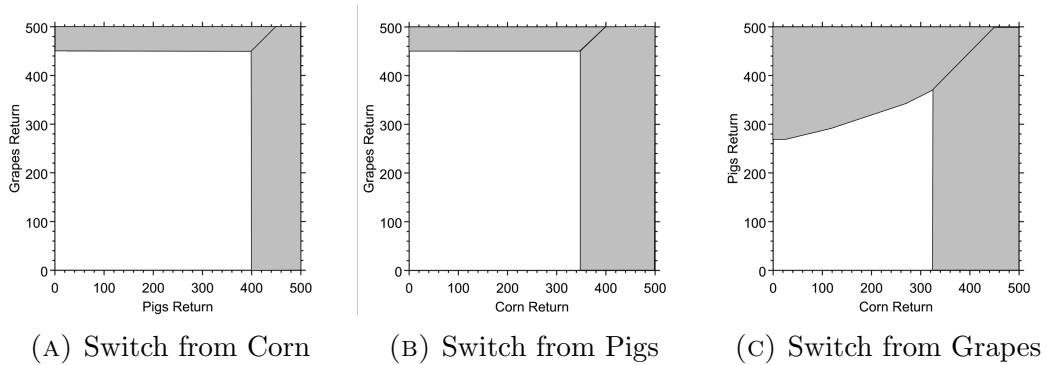


FIGURE 4.5: Switching Boundaries, current operating mode returns high-range $S_{0,mode} = 300$

of production are high, the farmer is less likely to switch. Additionally, the state prices of the alternative modes of productions must be higher for transition to occur. For example, in sub-figure 4.5a the farmer is producing corn at $t = 0$ and the current returns are 300, if the farmer is to consider switching, the returns from pigs or grapes must be sufficiently greater to compensate adequately for the uncertainty and the cost of switching. Figure 4.6 illustrates that the optimal

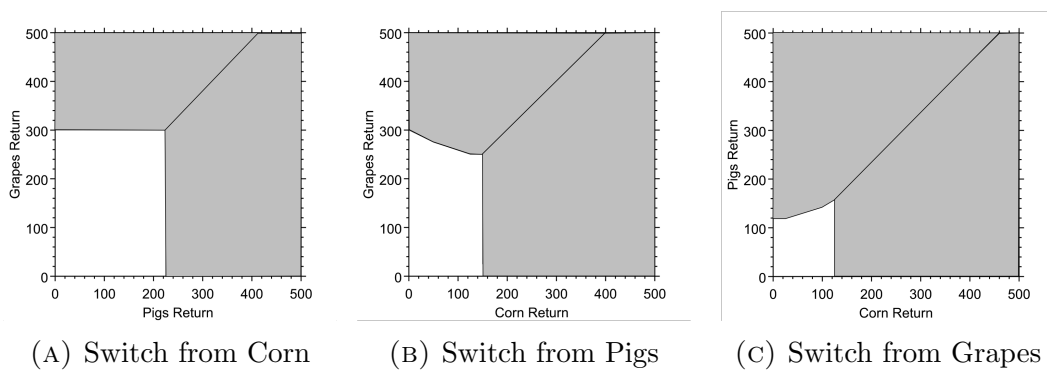


FIGURE 4.6: Switching Boundaries, current operating mode returns mid-range $S_{0,mode} = 150$

switching boundary for a grape farmer dictates that he/she will switch if returns of corn are slightly greater than grapes. In contrast, for the farmer to

transition from corn to grapes the price of grapes must be greater than twice the price of corn. Furthermore, Figure 4.7 illustrates clearly that the corn farmer is less likely to switch as there is a greater area of space where switching is not optimal relative to the other modes of operation. This is because the alternative modes are relatively more volatile and the cost to transition is high.

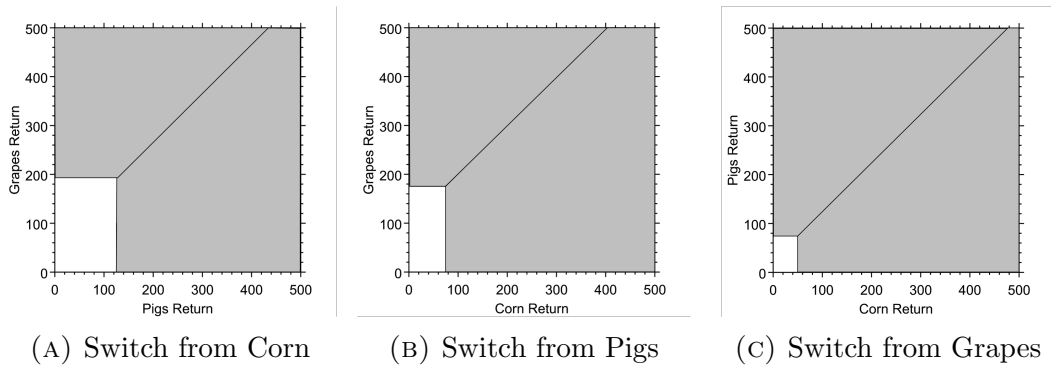


FIGURE 4.7: Switching Boundaries, current operating mode returns low-range
 $S_{0,mode} = 75$

The results of our three-dimensional analysis have been produced with acceptable reliability, cost of computation and conform with the intuitive expectation of real options analysis. They also present an extension to multi-dimensional real options analysis of the optimal land use problem.

4.2 Extensions

There are a vast number of empirical applications associated with the optimal switching policy of land use. [Luong and Tauer \(2006\)](#) and [Song et al. \(2011\)](#) have applied a two-dimensional switching model to an empirical agricultural model. Their analysis observed the impacts of political and technological shocks to the

economic system. An extension of their analysis to higher dimensions is possible with the higher-dimensional methodology proposed in this paper.

Furthermore, historically we are aware of periods through time where technology has made processes more efficient or enabled access to market places previously unreachable. For example, kiwifruit horticulture is a major export earner in New Zealand. Kiwifruit production in New Zealand has recently experienced both negative and positive shocks. Interestingly, some varieties have been afflicted by bacterial disease, while another non-furry variety is very profitable from strong demand in the Asian markets. The Kiwifruit farmers have an option to switch varieties by ripping up the vines but at a cost and with additional land use options.

Also an interesting extension would be to include in the model the transitional states that occur between switching. For example, land use transformation will not occur immediately upon the decision to do so being made. We understand that transformation includes a number of transitional states, which represent options to continue, pause, accelerate or revert. For example, a sheep farmer may decide to transition to grapes. During the transition he will have the option to revert to sheep, or pause the transition, anticipating continuation in more favorable circumstances. Alternatively, he may wish to accelerate the transformation if grape prices increase further.

Lastly, the performance testing we have performed could be expanded to measure the performance of the LSM model against a parallel method of valuation. In two dimensions, [Dixit and Pindyck \(2012\)](#) propose a semi-closed-form method for real options valuation with switching costs and reversibility. Also, in higher dimensions

[Broadie and Glasserman \(1997\)](#) propose a numerical method for pricing higher dimensional options uses a stochastic mesh methodology.

Chapter 5

Conclusion

We have presented and discussed a methodology for the valuation of the optimal switching policy in multiple dimensions. The valuation approach is implemented using a numerical method based on the [Longstaff and Schwartz \(2001\)](#) Least Squares method(LSM) which is appropriate for multi-dimensional problems and has the flexibility to accommodate multiple stochastic processes. Further, the LSM method is extended to the switching problem using [Gamba \(2003\)](#).

We have assessed the accuracy of the methodology, and demonstrated the use of the methodology for a three-dimensional problem.

Appendix A

Matlab Code

```
1 function [ A ] = Gamba( del_t, rf, S, K, basis_functions)
2
3 % ----- GAMBA SWITCHING ----- %
4 %
5 % Switching Options Script. Methodology from Gamba(2003).
6 %
7 % Inputs: del_t - Time step
8 %         : rf - risk free rate
9 %         : S - Asset(mode) correlated GBM pathways. S(#sim,time,asset/mode)
10 %        : K - Transition/switching cost matrix. K(1,2)=cost of transition
11 %          from mode 1 to mode 2. Therefore, K(1,1)=0.
12 %        : basis_functions - basis for LSM regression
13 %
14 % Output: A - The discounted optimum cash flow given switching.
15 %
16
17 % 1.0 Initialization
18 %   1.1 Switching costs
19 %   Establish relative switch cost from-to for each asset transition path.
20 %   Switch_cost has length=number of simulations, width=number of assets
21 %   and holds a third dimensionality which accounts for the relativity of
22 %   switching from-to e.g. with three assets the 1-1(stay), 1-2(move to
23 %   two), 1-3(move to three)... there are nine combinations i.e. there are
24 %   three perspectives which need to be evaluated at any time.
25 T=size(S,2)-1;
26 N=size(S,3);
27 nsims=size(S,1);
28 switch_cost=reshape(ones(nsims,1)*reshape(K',1,[],nsims,[],N);
29
30 % 2.0 Switching at terminal time = T (special case)
31
32 % squeeze extracts time period asset data at termination from larger matrix
33 % S. Then, replicate t=T asset value in three dimensions. This allows
34 % mode conditional perspective to be evaluated as discussed above.
35 period_cashflow = repmat(squeeze(S(:,T+1,:)),[1 1 N]);
36 % transition represents all conditional possibilities for switching, with
37 % these there is an optimum transition.
38 transition = (del_t*exp(-rf*del_t)*period_cashflow-switch_cost);
39 % transition_optimum represents the optimum transition.
40 transition_optimum = repmat(max(transition,[],2),[1 N]);
41 % switch_time is a binary matrix which locates/maps the optimum path
42 % values found in transition_optimum, within the possible pathways matrix
43 % transition.
44 switch_time(:, :, :) = transition == transition_optimum;
45
46 % A is a 2D matrix, length is number of simulations, each column represent
47 % the current state. A is not recorded in time, however, it is discounted
48 % by through time. Therefore, it is the accumulation of optimum cash flows
```

```
49 % given switching back to t=1.
50 A = squeeze(max(transition.*switch_time,[],2));
51
52 % 3.0 Switching from t=1 through t=T-1(general case).
53 % This methodology follows Gamba(2003). In summary this part of the
54 % function works backward in time evaluating the optimum pathway each time
55 % step. At each step expectations of the future are formed using the least
56 % squares method as originally set out in Longstaff and Schwartz and
57 % modified by Gamba (2003). (better explanation required here)Note, we
58 % cannot take expectations from  $S_0$  i.e.  $S_0 \setminus A$ , because all the  $S_0$  values ar
59 % the same.
60
61 for t= T:-1:2
62
63     % squeeze extracts time period asset data at termination from larger
64     % matrix S. Then, replicate t=T asset value in three dimenions. This
65     % allows perspective to be evaluated as discussed above.
66     period_cashflow= repmat(squeeze(S(:,t,:)),[1 1 N]);
67
68     % transition represents all possibilities for switching. Within the
69     % possibilties there is an optimum pathway. transition is calculated t
70     % combining the discounted payoff from being in mode z and subtracting
71     % the cost of switching from z to h.
72     transition=((del_t*exp(-rf*del_t)*period_cashflow)-switch_cost);
73
74     % continuation is the replication of Matrix A (discounted continuation
75     % value from simulation paths) replicated into three dimensions for us
76     % with the mode dependent conditional function of switch_time which
77     % maps optimum mode transition.
78     continuation=repmat(A,[1 1 N]);
79
80     % perform projection - form expectations of payoffs to switching. Cycl
81     % through each mode because optimum cash flow is conditional on being
82     % in each mode. Matrix A is regressed using a set of user defined basi
83     % functions on the set of simulated asset values S at time t. functior
84     % fit_and_evaluate only works in two dimensions, therefore, the for
85     % loop is used to cycle through the respective asset positions(columns
86     % in matrix A.
87
88     [BFn] = evaluate_beta(basis_functions,squeeze(S(:,t,:)));
89
90     for i = 1:1:N
91         regression_coeff = BFn \ A(:,i);
92         expect_A(:,i)=BFn*regression_coeff;
93
94     %         Regression analysis - R^2 and RMSE
95     %         r2(t,i) = 1 - sum((A(:,i)-expect_A(:,i)).^2)/sum((A(:,i)).^2);
96     %         rmse (t,i) = sqrt(mean((A(:,i) - expect_A(:,i)).^2));
```

```
97     end
98
99     % cond_approx represents the expected continuation value(E*[A]) +
100     % transition(current period cash flow - switching cost)
101     cont_approx= repmat(expect_A,[1 N])+transition;
102     cont_approx_optimum=repmat(max(cont_approx,[],2),[1 N]);
103     % switch_time is the decision rule for switching along the w-th
104     % simulation path.
105     switch_time(:,:,:)= cont_approx==cont_approx_optimum;
106     % Update A. Calculate new cashflows using *actual* payoffs to
107     % switching. Lastly, discount back one timestep for next backward
108     % recursion of for loop t-1.
109     A = exp(-rf*del_t)*squeeze(max((transition+continuation).*switch_time,
110
111 end
112
113 % Account for final period from S0 through to S1. Therefore, present value
114 % is at t=0.
115 period_cashflow=del_t*exp(-rf*del_t)*squeeze(S(:,1,:));
116 A = A+period_cashflow;
117
118 end
119
120
121
122
123
124
125
126
127
128
129
```

```
1 function [ S ] = GBM( S0 , mu , sigma , del_t , T , nsims , cov )
2 % Function to generate correlated sample paths for multiple variables
3 % assuming geometric Brownian motion.
4 %
5 % Inputs: S0 - t=0 return
6 %         : mu - expected drift
7 %         : sigma - expected volatility
8 %         : cov - covariance matrix
9 %         : del_t - size of time steps
10 %        : T - number of time steps to calculate
11 %        : nsims - number of simulation paths to generate
12 %
13 % Output: S - a (steps+1)-by-nsims-by-nassets 3-dimensional matrix where
14 %            each column represents a time step, each row represents a
15 %            separate simulation run and each 3rd dimension represents a
16 %            different dimension.
17
18 % number of state variables, N
19 N=size(S0,2);
20
21 % inputs into multiple dimensions
22 del_t= repmat(del_t,[nsims T N]);
23 S0=repmat(reshape(S0,1,1,[]),nsims,T);
24 sigma=repmat(reshape(sigma,1,1,[]),nsims,T);
25 mu=repmat(reshape(mu,1,1,[]),nsims,T);
26
27 % Cholesky factorization on the covariance matrix
28 c=chol(cov);
29
30 % generate correlated random sequences and paths
31 % generate uncorrelated random sequence
32 d_z=randn(T*nsims,N);
33 % correlate the sequences
34 corr=d_z*c;
35
36 % generate potential paths (GBM)
37 % arrange random sequences into simulations
38 innovs=reshape(corr,nsims,[],N);
39 % GBM pathway generation
40 % For correlations:
41 innovs=innovs.*sigma.*sqrt(del_t)+(mu-0.5.*sigma.^2).*del_t;
42 % For covariances:
43 innovs=innovs.*sqrt(del_t)+(mu-0.5.*sigma.^2).*del_t;
44 innovs=cumsum(innovs,2);
45 S=S0.*exp(innovs);
46 S=[S0(:,1,:),S];
47
48 end
```

```
1 function [ polyfn ] = polyfn_build( poly_input )
2 % This function calls the polynomial family builder functions. If
3 % additional polynomial families are required add additional if statement
4 % input and call function.
5 %
6 % Inputs: poly_input = row vector which contains two inputs. The first is
7 % the polynomial type. The second, is the degree of
8 % polynomial required. For example, [ 1 2 ] would
9 % call the first two Laguerre polynomials.
10 %
11 % Outputs: polyfn = matrix which contains all the coefficient terms of the
12 % selected polynomial family. polyfn is flipped to
13 % accommodate the matlab function 'polyval.m' which
14 % works from right to left. eg [2 0 3 0 1] is equivalent
15 % to 2*X^4 + 0*X^3 + 3*X^2 + 0*X + 1*Constant.
16
17
18 if poly_input(1) == 1
19     polyfn=Laguerre_builder(poly_input(2));
20 end
21
22 if poly_input(1) == 2
23     polyfn=Hermite_builder(poly_input(2));
24 end
25
26 % Flip the poly co-eff matrix to correct format for 'polyval.m'
27 polyfn = fliplr(polyfn);
28
29 end
30
31
```

```
1 function [ basis_functions ] = Basis_build( dim, construct,...
2     weight_fn, alpha_values, alpha_power, lags, prod_all)
3 %
4 %----- basis_functions.m -----
5 %
6 % Function to generate a structural array of basis functions for use in th
7 % regression function 'fit_and_evaluate.m'. This function builds a set of
8 % basisfunctions from options which can be toggled by the user. The
9 % available options are consistent with the basis functions selected for
10 % multi-dimensional option analysis in Longstaff & Schwartz, 2001(refer
11 % section 8.1).
12 %
13 % Inputs: dim - number of dimensions to being evaluated
14 %         : construct - first column calls polynomial type (1= Laguerre, 2=
15 %         Hermite). Second column calls degree of polynomial i.e. if 2 then
16 %         the first two polynomials are called from the family excl constant
17 %         Next n(dimensions considered) are assets S1, S2,... etc. lastly,
18 %         the rows from n+2 through 2*n+2 represent the sorted values of the
19 %         asset X1, X2, X3 etc. Note, if construct = 1 then a constant only
20 %         will be included in the basis set. If construct = 0 then no
21 %         constant nor polynomial terms will be included in the basis set.
22 %         Input example: three dimensional problem, third degree laguerre
23 %         polynomial in all assets and sorted assets. Furthermore a lagged
24 %         cross product is included in the second order laguerre in all
25 %         assets and assets sorted.
26 %
27 %         construct = [1 2 1 0 0 0 0 0;
28 %                     1 2 0 1 0 0 0 0;
29 %                     1 2 0 0 1 0 0 0;
30 %                     1 2 0 0 0 1 0 0;
31 %                     1 2 0 0 0 0 1 0;
32 %                     1 2 0 0 0 0 0 1;
33 %                     1 1 1 1 0 0 0 0;
34 %                     1 1 0 1 1 0 0 0;
35 %                     1 1 0 0 0 1 1 0;
36 %                     1 1 0 0 0 0 1 1]
37 %
38 %         : weight_fn - input function handle for weighting of all basis
39 %         terms
40 %         : alpha_values - toggle(1|0 = on|off) includes a basis fn for each
41 %         individual asset value eg S1, S2,... i.e.
42 %         alpha_values(1,:) toggles alpha(unsorted) inclusion in basis
43 %         alpha_values(2,:) toggles alpha sorted inclusion
44 %         : alpha_power - toggle asset values to a power term. Input format=
45 %         (N|0 = on(degree)|off). i.e.
46 %         alpha_power(1,:) toggles alpha(unsorted)^N basis inclusion
47 %         alpha_power(2,:) toggles alpha sorted^N basis inclusion
48 %         : lags - toggle(1|0 = on|off) lagged product of assets eg S1*S2,
```



```
49 %      S2*S3, ...
50 %      : prod_all - toggle(1|0 = on|off) product of all assets eg
51 %      S1*S2*S3...
52 %
53 % Outputs - basis_functions is a structural array with four fields. The
54 % size of the field is determined by the number of basis functions and the
55 % inputs above. Specifically, the following feilds are output:
56 %
57 %      : basis_functions.polynomial - Contains poly coefficients. Further
58 %      ones in the first order are used to ensure compatibility of array
59 %      dimensionality with basis functions set.
60 %
61 %      : basis_functions.weightingfn - Contains handle @function for
62 %      dim=# basis functions
63 %
64 %      : basis_functions.alphas - If eval_sorted_alpha=0. This contains
65 %      binary code which dictates how the assets are selected for the
66 %      collation of basis functions. If eval_sorted_alpha=1, then, this
67 %      feild is full of zeros - effectively turning off the evaluation of
68 %      basis functions with unsorted asset values.
69 %
70 %      : basis_functions.alphasortedS - equivilent and opposite function
71 %      to field: basis_functions.alphas
72 %
73 %Establish the required length(# of basisfn) of array: basis_functions;
74 if sum(construct(:)) == 0
75     constant = 0;
76 else
77     constant = 1;
78 end
79
80 if size(construct,2) > 1
81     num_polyfn = sum(construct(:,2));
82 else
83     num_polyfn = 0;
84 end
85
86 num_basis=(dim-1)*lags + prod_all + constant + sum(alpha_values(:))...
87     + sum(logical(alpha_power(:))) + num_polyfn;
88
89
90 % ----- Build Polynomial Terms -----%
91 [ poly, alpha_poly_eval, alpha_sorted_poly_eval ] = poly_build (construct,
92
93 % Input polynomial coefficeints into structure array field
94 for i=1:num_basis
95     basis_functions(i).polynomial=poly(i,:);
96 end
```

```
97
98 % ----- Weighting Function -----%
99 for i=1:num_basis
100     basis_functions(i).weightingfn=weight_fn;
101 end
102
103 % ----- Build Alphas -----%
104
105 [ alphas , alphasortedS ] = ...
106     alpha_build( dim , alpha_poly_eval, alpha_sorted_poly_eval, alpha_valu
107
108 % Input alpha terms into structure array field
109 for i=1:num_basis
110     basis_functions(i).alphas=alphas(i,:);
111     basis_functions(i).alphasortedS=alphasortedS(i,:);
112 end
113
114
115 end
116
117
```

```
1 function [ poly_output, alpha_poly_eval, alpha_sorted_poly_eval ] = poly_t
2
3 %
4 %The function poly_build.m builds a set of polynomial basis for use in the
5 %function: fit_and_evaluate.m.
6 %
7 %Inputs: construct - The first column calls polynomial type.
8 %       Second column calls degree of polynomial i.e. if 2 then the first
9 %       two polynomials are called from the family excl constant. Next
10 %      n(dimensions considered) are assets S1, S2,... etc. lastly, the
11 %      rows from n+2 through 2*n+2 represent the sorted values of the
12 %      asset X1, X2, X3 etc. Note, first row includes constant in poly
13 %      set.
14 %       Input example: three dimensional problem, third degree laguerre
15 %       polynomial in all assets and sorted assets. Furthermore a lagged
16 %       cross product is included in the second order laguerre in all
17 %       assets and assets sorted.
18 %
19 %       construct = [1 2 1 0 0 0 0 0;
20 %                   1 2 0 1 0 0 0 0;
21 %                   1 2 0 0 1 0 0 0;
22 %                   1 2 0 0 0 1 0 0;
23 %                   1 2 0 0 0 0 1 0;
24 %                   1 2 0 0 0 0 0 1;
25 %                   1 1 1 1 0 0 0 0;
26 %                   1 1 0 1 1 0 0 0;
27 %                   1 1 0 0 0 1 1 0;
28 %                   1 1 0 0 0 0 1 1]
29 % NOTE:
30 % if construct is input as above then a constant will automatically be
31 % included once in the basis set.
32 %
33 % if construct=1 then only a constant will be included in the basis set.
34 % that is no polynomial terms will be included.
35 %
36 % if construct=0 then no constant will be included in the basis set.
37 %
38 % Outputs: poly_output - matrix which contains the polynomial coefficients
39 %       stored row by row for each basis. poly_output is in the form
40 %       required of matlab function: polyval.m which evaluates
41 %       polynomials. polyval.m is used by function fit_and_evaluate.m to
42 %       evaluate polynomial basis functions.
43 %
44 %       alpha_poly_eval - contains the alpha(unsorted asset values)
45 %       which the polynomial is to be evaluated in. Rows correspond to
46 %       rows of polynomial coefficients stored in the matrix
47 %       poly_output.
48 %
```

```
49 %         alpha_sorted_poly_eval - contains the sorted asset values which
50 %         the polynomial is to be evaluated in. Structure is likewise to
51 %         alpha_poly_eval.
52
53
54 % No constant, no polynomal
55 if sum(construct(:)) == 0
56     poly_output = [ones(num_basis,1),zeros(num_basis,1)];
57     alpha_poly_eval = [];
58     alpha_sorted_poly_eval = [];
59     return
60 end
61
62 % Constant only, no polynomal
63 if sum(construct(:)) == 1
64     poly_output = [ones(num_basis,1),zeros(num_basis,1)]
65     alpha_poly_eval = zeros(1,dim)
66     alpha_sorted_poly_eval = zeros(1,dim)
67     return
68 end
69
70 % preallocate matrix hand & poly_output
71 hold = [];
72 poly_output = [];
73 max_degree = max(construct(:,2));
74
75 %calculate # of iteratons required. function moves through input matrix rc
76 %by row.
77 for i=1:size(construct, 1)
78
79     poly_input = construct(i,1:2);
80     polyfn = polyfn_build(poly_input);
81
82     % Adjust size of lesser degree outputs to account for higher orde
83     % polynomials
84
85     if construct(i,2) < max_degree
86         sx = size(polyfn);
87         p = zeros(sx(1),abs(sx(2)-max_degree-1));
88         polyfn = [p , polyfn];
89     end
90
91     % Remove constant term from subsequent iterations such that it is
92     % only included once.
93     if i == 1
94         constant = zeros(1,size(construct, 2)-2);
95         hand = [constant; repmat(construct(i,3:size(construct,2)),pol
96     else
```

```
97         polyfn(1,:)=[];
98         %repmat row to allow for additional degrees of polynomial
99         hand = repmat(construct(i,3:size(construct,2)),poly_input(2),
100     end
101
102     % hand off and store instruction matrix
103     hold = [hold; hand];
104     poly_output = [poly_output ; polyfn];
105
106 end
107
108 % make allowance for other basis functions included. Matlab function
109 % polyval will assess in S or X by using an input of one in the second to
110 % last column.
111 tack_on = zeros(num_basis-size(poly_output,1), size(poly_output,2));
112 tack_on(:,size(poly_output,2)-1) = 1;
113 poly_output= [poly_output ; tack_on];
114
115 % Split hold matrix into alpha and alpha-sorted instruction matrices
116 % respectively.
117 alpha_poly_eval = hold(1:end, 1:end/2);
118 alpha_sorted_poly_eval = hold(1:end, end/2+1:end);
119
120 end
121
122
```

```
1 function [BFn] = evaluate(basis_functions,S)
2 % Function 'evaluate.m' evaluates the regression coefficients of associate
3 % with the set of basis functions (BFn).
4 %
5 % Inputs: basis_functions - Structural array with four fields(.polynomial,
6 %       .alphas, .alphasortedS, and .weightingfn). Length of array is
7 %       equal to number of basis functions. For example,
8 %       basis_function(1) is equivalent to the set of instructions to
9 %       form the first basis function.
10 %
11 %       : S - Matrix of asset values [X]. Length is number of simulations,
12 %       width is number of asset dimensions.
13 %
14 % Outputs: BFn - evaluated basis functions
15 %
16 % Note - sorting of S is performed along rows/simulations and locates the
17 % maximum value in the first column, and so forth.
18
19 % sort along rows, maximum in leading column.
20 S_sorted=sort(S,2,'descend');
21
22 B = arrayfun(@(x)...
23     prod([S.^(ones(size(S,1),1)*[x.alphas]),...
24         S_sorted.^(ones(size(S,1),1)*[x.alphasortedS]),2),...
25     basis_functions,'UniformOutput',false);
26
27 BFn=cell2mat(arrayfun(@(x1,x2)...
28     (prod([x1.weightingfn(cell2mat(x2)),...
29         polyval(x1.polynomial,cell2mat(x2))],2)),...
30     basis_functions,B,'UniformOutput',false));
31
32 end
33
34
```

```
1 function y = expx_func(x)
2
3     y = exp(-x);
4     %y=x.^0;
5
6 end
7
8
```

Bibliography

- Milton Abramowitz and Irene A Stegun. *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*. Number 55. Courier Dover Publications, 1972.
- Phelim Boyle, Mark Broadie, and Paul Glasserman. Monte carlo methods for security pricing. *Journal of economic dynamics and control*, 21(8):1267–1321, 1997.
- Kjell Arne Brekke and Bernt Øksendal. Optimal switching in an economic activity under uncertainty. *SIAM Journal on Control and Optimization*, 32(4): 1021–1036, 1994.
- Mark Broadie and Paul Glasserman. Pricing american-style securities using simulation. *Journal of Economic Dynamics and Control*, 21(8):1323–1352, 1997.
- Emmanuelle Clément, Damien Lambertson, and Philip Protter. An analysis of a least squares regression method for american option pricing. *Finance and Stochastics*, 6(4):449–471, 2002.
- Robert K Dixit and Robert S Pindyck. *Investment under Uncertainty*. Princeton University Press, 2012.
- Andrea Gamba. Real options valuation: A monte carlo approach. *Faculty of Management, University of Calgary WP*, (2002/3), 2003.
- Paul Glasserman. *Monte Carlo methods in financial engineering*, volume 53. Springer, 2004.
- Myron J Gordon and Eli Shapiro. Capital equipment analysis: the required rate of profit. *Management science*, 3(1):102–110, 1956.
- John Hull. *Options, futures and other derivatives*. Pearson education, 2009.
- Francis A Longstaff and Eduardo S Schwartz. Valuing american options by simulation: A simple least-squares approach. *Review of Financial studies*, 14 (1):113–147, 2001.
- Quoc V Luong and Loren W Tauer. A real options analysis of coffee planting in vietnam. *Agricultural Economics*, 35(1):49–57, 2006.

-
- Manuel Moreno and Javier F Navas. On the robustness of least-squares monte carlo (lsm) for pricing american derivatives. *Review of Derivatives Research*, 6(2):107–128, 2003.
- Todd M Schmit, Jianchuan Luo, and Jon M Conrad. Estimating the influence of us ethanol policy on plant investment decisions: A real options analysis with two stochastic variables. *Energy Economics*, 33(6):1194–1205, 2011.
- Feng Song, Jinhua Zhao, and Scott M Swinton. Switching to perennial energy crops under uncertainty and costly reversibility. *American Journal of Agricultural Economics*, 93(3):768–783, 2011.