NUMERICAL METHODS FOR PRICING BASKET OPTIONS

DISSERTATION

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Aniela Karina Iancu, M.S.

The Ohio State University

2004

Dissertation Committee:  
Professor Bostwick Wyman, Adviser  
Professor Neil Falkner  
Professor Rick Evans

Approved by  
Adviser

Department of Mathematics
Most of the time, when pricing financial instruments, it is impossible to find closed form solutions for their values. Finding numerical solutions for the governing pricing equations becomes therefore an appealing approach to pricing, especially since powerful desktop computers are now available.

In this paper we demonstrate how two of the main numerical methods known today — the finite differences method and the Monte Carlo simulation — can be used for pricing discretely measured lookback basket options.

We also take a look at one of the most competitive markets today, The Individual Variable Annuity marketplace, at some of the currently sold death benefits and how they are related to the lookback put options.
In the memory of my parents
ACKNOWLEDGMENTS

A paper is almost always the product not only of its authors, but also of the environment where the authors work, of the encouragements and criticisms gathered from colleagues and teachers, conversations after seminars and many analogous events.

While I cannot do justice to all of the above, I thank explicitly my adviser Dr. Bostwick Wyman for intellectual support, invaluable guidance and encouragement throughout the years.

I am deeply indebted to Daniel Heyer, who introduced me to the area of my current research and suggested many deep and stimulating problems.

I also want to thank Yung-Chen Lu for bringing me to The Ohio State University and Richard Evans for introducing me to the actuarial profession.

My perception of Mathematics and Finance has developed under the influence of many people. Especially, I would like to mention Ioan Tomescu and Misu Negrioiu from the University of Bucharest and Neil Falkner, Neil Robertson and John Hsia from The Ohio State University.

Finally, my ultimate gratitude goes towards my husband, whose tremendous love, help and support made this work possible.
VITA

October 31, 1975 .......... Born - Bacau, Romania
1997 ..................... B.S. Mathematics, University of Bucharest
1998 ..................... B.S. International Transactions, Academy of Economical Studies, Bucharest
1999 ..................... M.S. Management, The National School of Political and Administrative Studies, Bucharest
2000 ..................... M.S. Statistics and Optimization, University of Bucharest
2000 - present ............ Graduate Teaching Assistant, The Ohio State University
Summer 2003 .............. Actuarial Intern, Asset Liability Management, Nationwide Financial

FIELDS OF STUDY

Major Field: Mathematics
# TABLE OF CONTENTS

ABSTRACT ............................................ ii

DEDICATION ........................................... iii

ACKNOWLEDGMENTS ................................. iv

VITA .................................................... v

LIST OF FIGURES ....................................... ix

1 INTRODUCTION ................................. 1

1.1 Stocks and Their Derivatives ............... 1

1.2 Hedging ........................................ 2

1.3 Arbitrage ....................................... 3

1.4 A Simple Model for Asset Pricing–The Lognormal Random Walk ......................... 3

1.5 Ito’s Lemma for Two Assets ................. 5
1.6 Derivation of the Black-Scholes Formula for Two Assets . . . . 6
1.7 Options on Dividend Paying Assets . . . . . . . . . . . . . . 9
1.8 The Explicit Finite Difference Method . . . . . . . . . . . . 10
1.9 Monte-Carlo Simulation . . . . . . . . . . . . . . . . . . . . 13

2 EUROPEAN LOOKBACK PUT BASKET OPTION WITH
CONSTANT NUMBER OF SHARES . . . . . . . . . . . . . . . . . . . . 16
2.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . 16
2.2 Similarity Reductions . . . . . . . . . . . . . . . . . . . . . . 17
2.3 Implementation of the Finite-differences Method . . . . . . . 21
2.4 Numerical Examples . . . . . . . . . . . . . . . . . . . . . . . 27

3 EUROPEAN LOOKBACK PUT BASKET OPTION WITH
REBALANCING . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29
3.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . 29
3.2 The Monte-Carlo Simulation Algorithm . . . . . . . . . . . . . 31
3.3 Numerical Example . . . . . . . . . . . . . . . . . . . . . . . 36

4 GUARANTEED MINIMUM DEATH BENEFITS . . . . . . . . 38
4.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . 38
4.2 Survival Distributions . . . . . . . . . . . . . . . . . . . . . . 39
   4.2.1 The Survival Function . . . . . . . . . . . . . . . . . . 39
   4.2.2 Time-until-death for a Person Age \( x \) . . . . . . . . . . 40
   4.2.3 Force of Mortality . . . . . . . . . . . . . . . . . . . . . 40
   4.2.4 Constant Force of Mortality . . . . . . . . . . . . . . . 41
4.3 Multiple Decrement Models . . . . . . . . . . . . . . . . . . . 41
4.4 Associated Single Decrement Tables . . . . . . . . . . . . . . 44
LIST OF FIGURES

2.1 The Three-dimensional Mesh for $U(i, j, K + 1)$ . . . . . . . 26
2.2 Lookback Put Basket Option . . . . . . . . . . . . . . . . . 27
2.3 European Lookback Put Basket Option . . . . . . . . . . . . 28
3.1 Lookback Put with Rebalancing . . . . . . . . . . . . . . . . 37
4.1 Claims Values over 10 Years . . . . . . . . . . . . . . . . . . 50
CHAPTER 1

INTRODUCTION

1.1 Stocks and Their Derivatives

A company that needs to raise money can do so by selling its shares to investors. The company is then owned by its shareholders. These owners possess shares (also known as equity certificates or stocks) and may or may not receive dividends, depending on whether the company makes a profit and decides to share this profit with its owners.

The value of the company’s stock reflects the views or predictions of investors about the likely dividend payments, future earnings and resources that the company will control.

A stock’s derivative is a specific contract whose value at some future date will depend entirely on the stock’s future value. The stock that the contract is based on is called the underlying equity.
One of the simplest financial derivatives is a *European call option*. This is a contract with the following conditions: at a prescribed time in the future, known as the *expiry date*, the *holder* of the option may purchase a prescribed asset, known as the *underlying asset* for a prescribed amount, known as the *exercise price* or *strike price*.

The word *may* in the above description implies that, for the holder of the option, the contract is a *right* and not an *obligation*. The other party to the contract, who is known as the *writer*, does have a potential obligation: he must sell the asset if the holder chooses to buy it.

The right to *sell* an asset is known as a *put option* and has payoff properties which are opposite to those of a call.

The options that may be exercised only at expiry are called *European options* and those that may be exercised at any time prior to the expiry date are called *American options*.

*Multiasset options* include options involving a choice between two or more instruments (e.g. *rainbow options*) and options involving baskets (e.g. options on a weighted sum of two or more assets).

*Exotic options* (also known as *path-dependent options*) have values that depend on the history of an asset price, not just on its value at expiry date.

### 1.2 Hedging

Every day banks, multinational corporations, investment houses, funds and investors enter into large financial positions. These entities and individuals
wish to protect themselves from risk and uncertainty or wish to limit the risk and uncertainty to tolerable levels.

Hedging is a means of minimizing this risk—i.e. is a form of insurance.

*Hedging* is the reduction of the sensitivity of a portfolio to the movement of an underlying asset by taking opposite positions in different financial instruments.

### 1.3 Arbitrage

Traders define *arbitrage* as a form of trading that makes a bet on a differential between instruments, generally with the belief that the return will be attractive in relation to the risks incurred. Arbitrageurs believe in capturing mispricings between instruments on markets.

In more formal academic literature, *arbitrage* means that a linear combination of securities costing zero can have the possibility of turning up with a positive value without ever having a negative value.

### 1.4 A Simple Model for Asset Pricing—The Lognormal Random Walk

It is often stated that asset prices must move randomly because of the *efficient market hypothesis*. There are several different forms of this hypothesis with different restrictive assumptions, but they are basically asserting two things:
1. The past history is fully reflected in the present price, which does not hold any further information, and

2. Markets respond immediately to any new information about an asset.

Thus the modelling of asset prices is really about modelling the arrival of new information which affects the price.

The absolute change in the asset price is not by itself a useful quantity: a change of $1 is much more significant when the asset price is $10 than when it is $100.

A better indicator of the size of a change in asset’s price is the return, defined to be the change in asset price divided by the original value. This return can be decomposed into two parts: a predictable, deterministic and anticipated return similar to the return on money invested in a risk-free bank; the second part models the random change in the asset price in response to external factors, such as unexpected news.

We can write:

\[ \frac{dS}{S} = \mu dt + \sigma dW, \]

where:

- \( \mu \) = drift, a measure of the average rate of growth of the asset price;
- \( \sigma \) = volatility, a measure of the standard deviation of the return;
- \( W \) = a Wiener process,

which leads to:

\[ d(\ln(S_t)) = \left( \mu - \frac{\sigma^2}{2} \right) dt + \sigma dW_t. \]
The solution of the above equation is:

\[ S_t = S_0 \exp \left[ \left( \mu - \frac{\sigma^2}{2} \right) t + \sigma W_t \right]. \]

1.5 Ito’s Lemma for Two Assets

Let \( V(S_1(t), S_2(t), t) \) be the price of a derivative, with \( S_1(t) \) and \( S_2(t) \) the prices of the two assets at time \( t \). We assume that \( V \) is an indefinitely differentiable function of \( S_1, S_2 \) and \( t \). From Taylor series expansion for \( V \), we obtain:

\[
dV = \frac{\partial V}{\partial t} dt + \frac{\partial V}{\partial S_1} dS_1 + \frac{1}{2} \frac{\partial^2 V}{\partial S_1^2} (dt)^2 + \frac{1}{2} \frac{\partial^2 V}{\partial S_1 \partial S_2} dS_1 dS_2 + \frac{1}{2} \frac{\partial^2 V}{\partial S_2^2} (dS_2)^2 + \frac{\partial^2 V}{\partial S_1 \partial t} dS_1 dt + \frac{\partial^2 V}{\partial S_2 \partial t} dS_2 dt + \frac{\partial^2 V}{\partial S_1 \partial S_2} dS_1 dS_2 + \text{higher order terms}
\]

Let \( \rho = \text{corr}(S_1, S_2) \) and assume:

\[
\begin{align*}
    dS_1 &= S_1 \mu_1 dt + S_1 \sigma_1 dW_1 \\
    dS_2 &= S_2 \mu_2 dt + S_2 \sigma_2 dW_2,
\end{align*}
\]

with:

\[
W_2 = \rho W_1 + \sqrt{1 - \rho^2} W_3
\]

where \((W_1, W_3)\) is a 2-dimensional Brownian motion.
Since \((dt)^2 \simeq 0\), \(dtdW_1 \simeq 0\), \(dtdW_2 \simeq 0\), \((dW_1)^2 \simeq dt\), \((dW_2)^2 \simeq dt\) and \(dW_1dW_2 \simeq \rho dt\), we get:

\[
\begin{align*}
(dS_1)^2 & \simeq \sigma_1^2 S_1^2 dt \\
(dS_2)^2 & \simeq \sigma_2^2 S_2^2 dt \\
dtdS_1 & \simeq 0 \\
dtdS_2 & \simeq 0 \\
ds_1dS_2 & \simeq \sigma_1 \sigma_2 \rho S_1 S_2 dt.
\end{align*}
\]

Then \(dV\) becomes:

\[
dV = \sigma_1 S_1 \frac{\partial V}{\partial S_1} dW_1 + \sigma_2 S_2 \frac{\partial V}{\partial S_2} dW_2 + \left[ \frac{\partial V}{\partial t} + \mu_1 S_1 \frac{\partial V}{\partial S_1} + \mu_2 S_2 \frac{\partial V}{\partial S_2} + \frac{1}{2} \sigma_1^2 S_1^2 \frac{\partial^2 V}{\partial S_1^2} + \frac{1}{2} \sigma_2^2 S_2^2 \frac{\partial^2 V}{\partial S_2^2} + \rho \sigma_1 \sigma_2 S_1 S_2 \frac{\partial^2 V}{\partial S_1 \partial S_2} \right] dt
\]

### 1.6 Derivation of the Black-Scholes Formula for Two Assets

We make the following assumptions:

1. The assets prices follow the lognormal random walk;

2. The risk-free interest rate \(r\), the assets volatilities \(\sigma_1\) and \(\sigma_2\) and the correlation coefficient \(\rho\) are known functions of time over the life of the option;
3. There are no transactions costs associated with hedging a portfolio;

4. There are no arbitrage possibilities;

5. The underlying assets pay no dividends;

6. Trading of the underlying assets can take place continuously;

7. Short selling is permitted and the assets are divisible.

Construct a portfolio of one option and $-\Delta_1$ shares of asset 1 and $-\Delta_2$ shares of asset 2. The value of the portfolio is:

$$\pi = V - \Delta_1 S_1 - \Delta_2 S_2.$$

Then:

$$d\pi = dV - \Delta_1 dS_1 - \Delta_2 dS_2.$$

Replacing $dV$ with the value from Ito’s Lemma, we get:

$$d\pi = dt \left[ \frac{\partial V}{\partial t} + \mu_1 S_1 \frac{\partial V}{\partial S_1} - \Delta_1 \mu_1 S_1 + \mu_2 S_2 \frac{\partial V}{\partial S_2} - \Delta_2 \mu_2 S_2 + \frac{1}{2} \sigma_1^2 S_1^2 \frac{\partial^2 V}{\partial S_1^2} + \frac{1}{2} \sigma_2^2 S_2^2 \frac{\partial^2 V}{\partial S_2^2} + \rho \sigma_1 \sigma_2 S_1 S_2 \frac{\partial^2 V}{\partial S_1 \partial S_2} \right] +
\quad + dW_1 (\sigma_1 S_1 \frac{\partial V}{\partial S_1} - \Delta_1 \sigma_1 S_1) + dW_2 (\sigma_2 S_2 \frac{\partial V}{\partial S_2} - \Delta_2 \sigma_2 S_2).$$

We can eliminate the random component in this random walk by choosing:

$$\Delta_1 = \frac{\partial V}{\partial S_1},$$
$$\Delta_2 = \frac{\partial V}{\partial S_2}.$$
The result is a portfolio whose increment is wholly deterministic:

\[ d\pi = dt \left[ \frac{\partial V}{\partial t} + \frac{1}{2} \sigma_1^2 S_1^2 \frac{\partial^2 V}{\partial S_1^2} + \frac{1}{2} \sigma_2^2 S_2^2 \frac{\partial^2 V}{\partial S_2^2} + \rho \sigma_1 \sigma_2 S_1 S_2 \frac{\partial^2 V}{\partial S_1 \partial S_2} \right]. \]

We now appeal to the concepts of arbitrage and supply/demand with the assumption of no transaction costs. The return on an amount \( \pi \) invested in riskless assets would see a growth of \( r\pi dt \) in a time \( dt \). So if the right-hand side of the above equation were greater than this amount, an arbitrager could make a guaranteed riskless profit by borrowing an amount \( \pi \) to invest in the portfolio. Conversely, if the right-hand side of the equation were less then \( r\pi dt \), then the arbitrager would short the portfolio and invest \( \pi \) in the bank.

Thus we have:

\[ r\pi dt = dt \left[ \frac{\partial V}{\partial t} + \frac{1}{2} \sigma_1^2 S_1^2 \frac{\partial^2 V}{\partial S_1^2} + \frac{1}{2} \sigma_2^2 S_2^2 \frac{\partial^2 V}{\partial S_2^2} + \rho \sigma_1 \sigma_2 S_1 S_2 \frac{\partial^2 V}{\partial S_1 \partial S_2} \right]. \]

But:

\[ \pi = V - \Delta_1 S_1 - \Delta_2 S_2 = V - \frac{\partial V}{\partial S_1} S_1 - \frac{\partial V}{\partial S_2} S_2, \]

so we obtain:

\[ \frac{\partial V}{\partial t} + \frac{1}{2} \sigma_1^2 S_1^2 \frac{\partial^2 V}{\partial S_1^2} + \frac{1}{2} \sigma_2^2 S_2^2 \frac{\partial^2 V}{\partial S_2^2} + \rho \sigma_1 \sigma_2 S_1 S_2 \frac{\partial^2 V}{\partial S_1 \partial S_2} + \\ + rS_1 \frac{\partial V}{\partial S_1} + rS_2 \frac{\partial V}{\partial S_2} - rV = 0. \]

This is the **Black-Scholes** formula for pricing an option on two assets.
1.7 Options on Dividend Paying Assets

The price of an option on assets that pay out dividends is affected by the payments, so we must modify the Black-Scholes formula.

Consider a constant dividend yield, so in a time \( dt \) the asset pays out a dividend \( qSdt \), with \( q = \text{constant} \).

Arbitrage considerations show that in each time-step \( dt \), the asset price must fall by the amount of the dividend payment in addition to the usual fluctuations. It follows that the random walk for the asset price is modified to:

\[
dS = \sigma SdW + (\mu - q)Sdt.
\]

Since we receive \( q_i \cdot S_i \cdot dt \) for every asset of type \( i \) held and since we hold \( -\Delta_i \) shares of asset \( i \) \((i = 1, 1)\), our portfolio changes to:

\[
d\pi = dV - \Delta_1 dS_1 - \Delta_2 dS_2 - q_1 S_1 \Delta_1 dt - q_2 S_2 \Delta_2 dt.
\]

The Black-Scholes equation becomes:

\[
\frac{\partial V}{\partial t} + \frac{1}{2} \sigma_1^2 S_1^2 \frac{\partial^2 V}{\partial S_1^2} + \frac{1}{2} \sigma_2^2 S_2^2 \frac{\partial^2 V}{\partial S_2^2} + \rho \sigma_1 \sigma_2 S_1 S_2 \frac{\partial^2 V}{\partial S_1 \partial S_2} + (r - q_1) S_1 \frac{\partial V}{\partial S_1} + (r - q_2) S_2 \frac{\partial V}{\partial S_2} - rV = 0.
\]
1.8 The Explicit Finite Difference Method

Rarely can we find closed form solutions for the values of options. Unless the problem is very simple, we are going to have to solve a partial differential equation numerically.

Finite-difference methods are means of obtaining numerical solutions to partial differential equations. They constitute a very powerful and flexible technique and are capable of generating accurate numerical solutions to many differential equations used for option pricing.

The idea underlying finite-difference methods is to replace the partial derivatives occurring in partial differential equations by approximations based on Taylor series expansions of functions near the point or points of interest.

For example, consider the diffusion equation:

\[
\frac{\partial u}{\partial \tau} = \frac{\partial u}{\partial x^2}.
\]

The partial derivative \(\frac{\partial u}{\partial \tau}\) may be defined as a limit:

\[
\frac{\partial u}{\partial \tau}(x, \tau) = \lim_{d\tau \to 0} \frac{u(x, \tau + d\tau) - u(x, \tau)}{d\tau},
\]

and the second partial derivative \(\frac{\partial^2 u}{\partial x^2}\) can be defined as:

\[
\frac{\partial^2 u}{\partial x^2}(x, \tau) = \lim_{dx \to 0} \frac{u(x + dx, \tau) - 2u(x, \tau) + u(x - dx, \tau)}{(dx)^2}.
\]

Next, we divide the \(x\)-axis into equally spaced nodes at distance \(dx\) apart, and the \(\tau\)-axis into equally spaced nodes at distance \(d\tau\) apart. This di-
vides the \((x, \tau)\)-plane into a mesh, where the mesh points have the form \((ndx, md\tau)\). We then concern ourselves only with the values of \(u(x, \tau)\) at the mesh points \((ndx, md\tau)\).

We write \(U(n, m) = u(ndx, md\tau)\). Using the above differences, the diffusion equation becomes:

\[
U(n, m + 1) = \alpha U(n + 1, m) + (1 - 2\alpha)U(n, m) + \alpha U(n - 1, m),
\]

where \(\alpha = \frac{d\tau}{(dx)^2}\).

If, at time-step \(m\), we know \(U(n, m)\) for all \(n\), we can explicitly evaluate \(U(n, m + 1)\). This is why this method is called explicit.

The above equation may be considered a random walk on a regular lattice, where \(U(n, m)\) denotes the probability of being at position \(n\) at time-step \(m\), \(\alpha\) denotes the probability of moving to the right or left by one unit and \((1 - 2\alpha)\) is the probability of staying put.

If we choose a constant \(x\) step \(dx\), we cannot solve the problem for all \(-\infty < x < \infty\) without taking an infinite number of \(x\)-steps. We get around this problem by taking a finite, but suitably large number of \(x\)-steps:

\[-Ndx \leq x \leq Ndx,\]

where \(N\) is a large positive number.

To start the iterative process, we use the payoff formula as the initial condition. To determine \(U(N, m)\), we need boundary conditions.
When using numerical schemes for solving partial differential equations, one needs to address three fundamental issues:

**Consistency:** A numerical scheme is said to be *consistent* if the finite difference representation converges to the partial differential equation we are trying to solve as the space and time steps tend to zero;

**Stability:** A numerical scheme is said to be *stable* if the difference between the numerical solution and the exact solution remains bounded as the number of time steps tends to infinity;

**Convergence:** A scheme is said to be *convergent* if the difference between the numerical solution and the exact solution at a fixed point in the domain of interest tends to zero uniformly as the space and time discretizations tend to zero.

A powerful statement links these issues together. This statement is the **Lax Equivalence Theorem:** *Given a properly posed linear initial value problem and a consistent finite difference scheme, stability is the only requirement for convergence.*

When using the explicit finite-difference method for pricing options on one underlying asset, it can be shown that the system is stable if $0 < \alpha \leq \frac{1}{2}$.

The stability condition puts severe constraints on the size of the time-steps. We must have:

$$0 < \frac{d\tau}{(dx)^2} \leq \frac{1}{2}.$$  

When dealing with options on two assets, the stability problem becomes much more complicated. In one special, but very important case, when we
have a *pure diffusion* problem:

\[
\frac{\partial V}{\partial \tau} = a \frac{\partial^2 V}{\partial x_1^2} + b \frac{\partial^2 V}{\partial x_2^2},
\]

the stability condition becomes:

\[
0 \leq a \frac{dt}{(dx_1)^2} + b \frac{dt}{(dx_2)^2} \leq \frac{1}{2}.
\]

In the last chapter we will attempt to give some directions for overcoming this time-step constraint.

### 1.9 Monte-Carlo Simulation

The basis of *Monte-Carlo simulation* is the strong law of large numbers, stating that the arithmetic mean of independent, identically distributed random variables, converges towards their mean almost surely.

Computing an option price means computing the discounted expectation of the payoff \(X\). This suggests the following:

**Algorithm**—determining the option price via Monte-Carlo simulation:

1. Simulate \(n\) independent realizations \(X_i\) of the final payoff \(X\);

2. Choose \((\frac{1}{n} \sum_{i=1}^{n} X_i) e^{-rT}\) as an approximation of the option price.

To simulate the final payoff, we first need to simulate a path for \(S(t)\), the stock price process. We can use the following:

**Algorithm:**

1. Divide the interval \([0, T]\) into \(N\) equidistant parts;
2. Generate $N$ independent random numbers $Y_i$ which are standard normally distributed;

3. From those, simulate an (approximate) path $W(t)$ of the Brownian motion on $[0, T]$: \[
W(0) = 0
\]
\[
W\left( j \frac{T}{N} \right) = W\left( (j - 1) \frac{T}{N} \right) + \sqrt{\frac{T}{N}} \cdot Y_j, \quad j = 1, \ldots, N
\]
\[
W(t) = W\left( (j - 1) \frac{T}{N} \right) + \left( t - (j - 1) \frac{T}{N} \right) \cdot \frac{N}{T} \cdot W\left( j \frac{T}{N} \right) - W\left( (j - 1) \frac{T}{N} \right), \quad \text{for } t \in \left( (j - 1) \frac{T}{N}, j \frac{T}{N} \right).
\]

4. Use $W(t)$ to generate an (approximate) path of the price process $S(t)$:
\[
S(t) = S_0(t) \cdot e^{(r - \frac{1}{2} \sigma^2)t} \cdot e^{\sigma W(t)}, \quad t \in [0, T];
\]

5. Use this simulated path of the price process to compute an estimate for the payoff $X$.

**Advantages:** The Monte-Carlo method for estimating an option price is very easy to implement. Nowadays, reasonable random numbers generators can be found in every programming language;
**Disadvantages:** Even given high speed computers, the method is time-consuming, as both $n$ and $N$ have to be very large to yield good estimates for the option price.
CHAPTER 2
EUROPEAN LOOKBACK
PUT BASKET OPTION
WITH CONSTANT NUMBER
OF SHARES

2.1 Introduction

A lookback option is a derivative product whose payoff depends on the maximum or minimum realized asset price over the life of an option. For example, a lookback put has a payoff at expiry that is the difference between the maximum realized price and the spot price at expiry. This may be written as:

$$\max(J - S, 0)$$,
where $J$ is the maximum realized price of the asset:

$$J = \max_{0 \leq \tau \leq t} S(\tau).$$

The maximum or minimum realized asset price may be measured continuously or, more commonly, discretely.

Most commercial lookback contracts are based on a discretely measured maximum or minimum because it is easier to measure the maximum of a small set of values, all of which can be guaranteed to be "real" prices at which the underlying assets have traded and also because by decreasing the frequency at which the maximum and the minimum are measured, some contracts become cheaper and therefore more appealing.

In this chapter, we analyze the lookback put option on a basket of two assets, with a discretely measured maximum.

We find that the lookback option leads to a partial differential equation with final and boundary conditions and that jump conditions apply across sampling dates.

The option value is a function of four variables: asset 1, asset 2, time and maximum realized basket price, but similarity reductions are going to be used to obtain more efficient numerical solutions.

### 2.2 Similarity Reductions

Consider two assets with $S_1(t)$ and $S_2(t)$ the prices at time $t$ and $\alpha_1, \alpha_2$ the number of shares of asset $i$ in the basket, $1 \leq i \leq 2$. 

17
The lookback option is a path dependent option. Its value $V$ is going to be a function of $S_1, S_2$, time and the maximum realized basket price.

Between samplings the maximum is constant, so during this time it is a parameter in the value of the option, in the same way that the exercise price is a parameter in the value of the vanilla option.

Therefore, the only random variables are $S_1$ and $S_2$ and the option price must satisfy the **Black-Scholes equation**:

$$
\frac{\partial V}{\partial t} + \frac{1}{2}\sigma_1^2 S_1^2 \frac{\partial^2 V}{\partial S_1^2} + (r - q_1)S_1 \frac{\partial V}{\partial S_1} - rV + \frac{1}{2}\sigma_2^2 S_2^2 \frac{\partial^2 V}{\partial S_2^2} + \rho \sigma_1 \sigma_2 S_1 S_2 \frac{\partial^2 V}{\partial S_1 \partial S_2} + (r - q_2)S_2 \frac{\partial V}{\partial S_2} = 0
$$

For arbitrage reasons, the realized option price must be continuous across sampling dates, so if $M$ is the maximum and $t_0$ is a sampling time, we have:

$$
V(S_1, S_2, M, t^-) = V(S_1, S_2, \text{Max}(M, \alpha_1 S_1 + \alpha_2 S_2), t_0^+).
$$

(the *jumping condition*).

We want to reformulate the problem so that the new function involves only three variables. Let $V(S_1, S_2, M, t) = M \cdot W(s_1, s_2, t)$ with $s_1 = \frac{S_1}{M}$, $s_2 = \frac{S_2}{M}$.
Then:

\[
\frac{\partial V}{\partial t} = M \frac{\partial W}{\partial t} \\
\frac{\partial V}{\partial S_1} = \frac{\partial V}{\partial s_1} \cdot \frac{\partial s_1}{\partial S_1} = \frac{\partial W}{\partial s_1} \\
\frac{\partial^2 V}{\partial S_1 \partial S_2} = \frac{\partial^2 W}{\partial s_1 \partial s_2} \cdot \frac{1}{M} \\
\frac{\partial^2 V}{\partial S_1^2} = \frac{\partial^2 W}{\partial s_1^2} \cdot \frac{1}{M}
\]

Then the partial differential equation for \( W \) is:

\[
\frac{\partial W}{\partial t} + \frac{1}{2} \sigma_1^2 s_1^2 \frac{\partial^2 W}{\partial s_1^2} + (r - q_1) s_1 \frac{\partial W}{\partial s_1} - rW + \]
\[
\frac{1}{2} \sigma_2^2 s_2^2 \frac{\partial^2 W}{\partial s_2^2} + \rho \sigma_1 \sigma_2 s_1 s_2 \frac{\partial^2 W}{\partial s_1 \partial s_2} + (r - q_2) s_2 \frac{\partial W}{\partial s_2} = 0
\]

which is exactly the \textit{Black-Scholes equation}.

The \textit{final condition} for the lookback put

\[ V(S_1, S_2, M, T) = \max(M - (\alpha_1 S_1 + \alpha_2 S_2), 0) \]

becomes:

\[ W(s_1, s_2, T) = \max(1 - (\alpha_1 s_1 + \alpha_2 s_2), 0). \]

The \textit{jump condition} across sampling dates

\[ V(S_1, S_2, M, t^-_0) = V(S_1, S_2, \text{Max}(M, \alpha_1 S_1 + \alpha_2 S_2), t^+_0). \]
becomes:

\[ W(s_1, s_2, t^+) = \max(1, \alpha_1 s_1 + \alpha_2 s_2), \]

\[ V\left(\frac{s_1}{\max(1, \alpha_1 s_1 + \alpha_2 s_2)}, \frac{s_2}{\max(1, \alpha_1 s_1 + \alpha_2 s_2)}; t^+\right). \]

**Boundary conditions:**

- \( S_1 = 0 \) and \( S_2 = 0 \). In this case, they can never be greater than zero, so the payoff at time \( T \) is known with certainty to be \( M \). Hence the interest rate discounted present value of the option is \( V(0, 0, M, t) = Me^{-r(T-t)} \).

We obtain the following condition:

\[ W(0, 0, t) = e^{-r(T-t)} \]

- \( S_1 = 0 \) and \( S_2 \neq 0 \). Then \( s_1 = 0 \) and \( W \) depends only on \( s_2 \) and \( t \):

\[
\frac{\partial W}{\partial t} + \frac{1}{2} \sigma_2^2 s_2^2 \frac{\partial^2 W}{\partial s_2^2} + (r - q_2)s_2 \frac{\partial W}{\partial s_2} - rW = 0. 
\]

- \( S_2 = 0 \) and \( S_1 \neq 0 \). Then \( s_2 = 0 \) and \( W \) depends only on \( s_1 \) and \( t \):

\[
\frac{\partial W}{\partial t} + \frac{1}{2} \sigma_1^2 s_1^2 \frac{\partial^2 W}{\partial s_1^2} + (r - q_1)s_1 \frac{\partial W}{\partial s_1} - rW = 0. 
\]

- \( S_1 \to \infty \) and \( S_2 \to \infty \). The value of the option is insensitive to small changes in \( M \), so \( \frac{\partial V}{\partial M} = 0 \). For \( W \) this condition becomes:

\[
\frac{\partial W}{\partial s_1} \frac{s_1}{W} + \frac{\partial W}{\partial s_2} \frac{s_2}{W} \sim 1 \text{ as } s_1 \to \infty \text{ and } s_2 \to \infty. 
\]
• $S_1 \rightarrow \infty$ and $S_2$ is finite. Then:

$$\frac{\partial W}{\partial s_1} \cdot \frac{s_1}{W} \sim 1.$$ 

• $S_2 \rightarrow \infty$ and $S_1$ is finite. Then:

$$\frac{\partial W}{\partial s_2} \cdot \frac{s_2}{W} \sim 1.$$ 

Thus, the strategy for valuing the lookback put option is as follows:

1. Solve the Black-Scholes equation for $W$ between sampling dates, using the value of the option immediately before the next sampling date as final data. This gives the value of the option until immediately after the present sampling date.

2. Apply the appropriate jump condition across the current sampling date to deduce the option value immediately before the present sampling date.

3. Repeat this process as needed to arrive at the current value of the option.

### 2.3 Implementation of the Finite-differences Method

We have three variables $s_1$, $s_2$ and $t$, so we are going to divide the $s_1$-axis into equally spaced nodes at distance $ds_1$ apart, the $s_2$-axis into equally
spaced nodes at distance $ds_2$ apart and the $t$-axis into equally spaced nodes at distance $dt$ apart.

Therefore, the grid will consist of the following points:

$$s_1(i) = (i - 1)ds_1, \quad 1 \leq i \leq I + 1$$

$$s_2(j) = (j - 1)ds_2, \quad 1 \leq j \leq J + 1$$

$$t(k) = (k - 1)dt, \quad 1 \leq k \leq K + 1$$

Since the original equation cannot be solved numerically for $0 \leq S_1 < \infty$ and $0 \leq S_2 < \infty$, we have to choose reasonable upper bounds for $S_1$ and $S_2$, for example four times the value of the maximum realized basket price. So $0 \leq s_1 \leq 4$ and $0 \leq s_2 \leq 4$, which would imply $ds_1 = \frac{4}{I}$ and $ds_2 = \frac{4}{J}$.

The value of $W$ at a certain point of the grid:

$$W ((i - 1)ds_1, (j - 1)ds_2, T - (k - 1)dt)$$

will be written as $U(i, j, k)$ for $1 \leq i \leq I + 1$, $1 \leq j \leq J + 1$ and $1 \leq k \leq K + 1$.

For $1 \leq i \leq I + 1$, $1 \leq j \leq J + 1$, let:

$$A(i) = \frac{1}{2} \sigma_1^2 [(i - 1)ds_1]^2$$

$$B(i) = (r - q_1)(i - 1)ds_1$$

$$C = -r$$

$$D(j) = \frac{1}{2} \sigma_2^2 [(j - 1)ds_2]^2$$

$$E(i, j) = \rho \sigma_1 \sigma_2 (i - 1)ds_1 (j - 1)ds_2$$

$$F(j) = (r - q_2)(j - 1)ds_2$$
The explicit difference scheme for the PDE is:

\[
\frac{U(i, j, k) - U(i, j, k + 1)}{dt} + \frac{A(i)}{(ds_1)^2} \cdot [U(i + 1, j, k) - 2U(i, j, k) + U(i - 1, j, k)] + \\
+ \frac{B(i)}{2ds_1} [U(i + 1, j, k) - U(i - 1, j, k)] + CU(i, j, k) + \\
+ \frac{D(j)}{(ds_2)^2} [U(i, j + 1, k) - 2U(i, j, k) + U(i, j - 1, k)] + \\
+ \frac{E(i, j)}{4ds_1ds_2} [U(i + 1, j + 1, k) - U(i + 1, j - 1, k) - U(i - 1, j + 1, k) + \\
+U(i - 1, j - 1, k)] + \frac{F(j)}{2ds_2} [U(i, j + 1, k) - U(i, j - 1, k)] = O(ds_1^2, ds_2^2, dt).
\]

The final condition:

\[
W(s_1, s_2, T) = \max(1 - (\alpha_1s_1 + \alpha_2s_2), 0) \text{ becomes:}
W(1, 1, 1) = \max(1 - (\alpha_1(i - 1)ds_1 + \alpha_2(j - 1)ds_2), 0).
\]

**Boundary conditions:**

• \(s_1 \to 0\) and \(s_2 \to 0\). Then:

\[
W(0, 0, t) = e^{-r(T-t)} \text{ becomes:}
W(1, 1, k) = e^{-r(k-1)dt}.
\]
\( s_1 \to 0. \) Then:

\[
\frac{U(1, j, k) - U(1, j, k + 1)}{dt} + D(j) \left[ \frac{U(1, j + 1, k) - 2U(1, j, k) + U(1, j - 1, k)}{(ds_2)^2} \right] + F(j) \left[ \frac{U(1, j + 1, k) - U(1, j - 1, k)}{2ds_2} \right] + CU(1, j, k) = 0 \]

becomes:

\[
\frac{U(1, j + 1, k + 1) - U(1, j, k)}{dt} + D(j) \left[ \frac{U(1, j + 1, k) - 2U(1, j, k) + U(1, j - 1, k)}{(ds_2)^2} \right] + F(j) \left[ \frac{U(1, j + 1, k) - U(1, j - 1, k)}{2ds_2} \right] + CU(1, j, k). \]

\( s_2 \to 0. \) Then:

\[
\frac{U(i, 1, k + 1)}{dt} = \frac{U(i, 1, k)}{dt} + A(i) \left[ \frac{U(i + 1, 1, k) - 2U(i, 1, k) + U(i - 1, 1, k)}{(ds_1)^2} \right] + B(i) \left[ \frac{U(i + 1, 1, k) - U(i - 1, 1, k)}{2ds_1} \right] + CU(i, 1, k). \]

\( s_1 \to \infty \) and \( s_2 \to \infty. \) Then:

\[
W = s_1 \frac{\partial W}{\partial s_1} + s_2 \frac{\partial W}{\partial s_2}
\]

\[
U(I + 1, J + 1, k) = I \left[ U(I + 1, J + 1, k) - U(I, J + 1, k) \right] + J \left[ (U(I + 1, J + 1, k) - U(I + 1, J, k) \right]
\]

\[
U(I + 1, J + 1, k) = \frac{I}{I + J - 1} U(I, J + 1, k) + \frac{J}{I + J - 1} U(I + 1, J, k).
\]
• \( s_1 \to \infty \). Then:

\[
W = s_1 \frac{\partial W}{\partial s_1} \\
U(I+1, j, k) = I [U(I+1, j, k) - U(I, j, k)] \\
U(I+1, j, k) = \frac{I}{I-1} U(I, j, k).
\]

• \( s_2 \to \infty \). Then:

\[
U(i, J + 1, k) = \frac{J}{J-1} U(i, J, k).
\]

These conditions can be summarized as follows:

1. The explicit difference scheme for the PDE: for \( 2 \leq i \leq I \), \( 2 \leq j \leq J \) and \( 1 \leq k \leq K \), knowing

\[
U(i-1, j-1, k) \quad U(i, j-1, k) \quad U(i+1, j-1, k) \\
\bullet \quad \bullet \quad \bullet \\
U(i-1, j, k) \quad U(i, j, k) \quad U(i+1, j, k) \\
\bullet \quad \bullet \quad \bullet \\
U(i-1, j+1, k) \quad U(i, j+1, k) \quad U(i+1, j+1, k) \\
\bullet \quad \bullet \quad \bullet 
\]

we can find \( U(i, j, k + 1) \).

2. Final condition: for \( 1 \leq i \leq I+1 \) and \( 1 \leq j \leq J+1 \), we know \( U(i, j, 1) \).

3. Boundary condition at \( s_1 = 0 \) and \( s_2 = 0 \): for \( 2 \leq k \leq K + 1 \), we know \( U(1, 1, k) \).
4. Boundary condition at $s_1 = 0$: for $2 \leq j \leq J$, knowing $U(1, j - 1, k)$, $U(1, j, k)$ and $U(1, j + 1, k)$, we can find $U(1, j, k + 1)$.

5. Boundary condition at $s_2 = 0$: for $2 \leq i \leq I$, knowing $U(i - 1, 1, k)$, $U(i, 1, k)$ and $U(i + 1, 1, k)$, we can find $U(i, 1, k + 1)$.

6. Boundary condition at $s_1 \rightarrow \infty$: knowing $U(I, j, k)$, we can compute $U(I + 1, j, k)$ for $1 \leq j \leq J$.

7. Boundary condition at $s_2 \rightarrow \infty$: knowing $U(i, J, k)$, we can compute $U(i, J + 1, k)$ for $1 \leq i \leq I$.

8. Boundary condition at $s_1 \rightarrow \infty$ and $s_2 \rightarrow \infty$: knowing $U(I, J + 1, k)$ and $U(I + 1, J, k)$, we can compute $U(I + 1, J + 1, k + 1)$.

We want to compute $U(i, j, K + 1)$ for $1 \leq i \leq I + 1$ and $1 \leq j \leq J + 1$.

![Figure 2.1: The Three-dimensional Mesh for $U(i, j, K + 1)$](image)

We can accomplish this by using the above conditions in the following order: 3, 2, 1, 4, 5, 6, 7, 8 (for details, please see Appendix A).
2.4 Numerical Examples

Table 2.2 shows the value of the option using different sets of sampling times. In each case the values shown are at 10 years before expiry with zero dividend yield for both assets and \( r = .1, \sigma_1 = \sigma_2 = .2, \rho = .1, \alpha_1 = .3, \alpha_2 = .7 \) and \( \max = 1 \). The columns of the table contain the following cases:

**Case 1** Sampling at times .5, 1.5, 2.5, \( \cdots \), 9.5 years.

**Case 2** Sampling at times 1.5, 3.5, 5.5, 7.5, 9.5 years.

**Case 3** Sampling at times 1.5, 5.5, 9.5 years.

**Case 4** No sampling—i.e. *vanilla put option*.

Using the MATLAB function `lookbasket.m`, with source code in Appendix A, we obtain the following values for the lookback put basket option:

<table>
<thead>
<tr>
<th>Basketprice/Max</th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
<th>Case 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>.9</td>
<td>.038</td>
<td>.031</td>
<td>.024</td>
<td>.0125</td>
</tr>
<tr>
<td>1.0</td>
<td>.026</td>
<td>.017</td>
<td>.010</td>
<td>.009</td>
</tr>
<tr>
<td>1.1</td>
<td>.032</td>
<td>.021</td>
<td>.013</td>
<td>.0066</td>
</tr>
</tbody>
</table>

Figure 2.2: Lookback Put Basket Option

For example, if the current basket price is 90 and the current maximum is 100, then we must search along the row starting with .9. The value of the
option will be, under sampling strategy 2, $100 \times .031 = 3.1$. Observe that the option price decreases as the number of samples decreases (from Case 1 to Case 3). This is financially obvious, since the lower the number of samples is, the lower the final payoff is expected to be. Decreasing the frequency of measurement of the maximum decreases their cost. This may be important, since one of the main commercial criticisms of lookback options is that they are expensive.

![European lookback put basket with sampling increment 1 year, time to maturity 10 years and current maximum M = 1](image)

Figure 2.3: European Lookback Put Basket Option.

Note also that the option price reaches a minimum close to the point where the basket price is equal to the current maximum. The option delta can become positive, since it is beneficial for the holder of the option if the basket price rises just before a sampling date and then falls.
3.1 Introduction

In Chapter 2, when we have evaluated the lookback put basket option, we have assumed that the number of shares of each stock in the basket is constant for the life of the option.

The problem changes completely if we rebalance the basket at each sampling date, such that the percentages of the money invested in each stock are constant.
We start with an initial amount $D$, and a constant vector:

$$\phi = (\phi(1), \phi(2), \cdots, \phi(n)),$$

with

$$\phi(1) + \phi(2) + \cdots + \phi(n) = 1,$$

containing the percentages invested in each of the stocks $S_1, S_2, \cdots, S_n$ from the basket.

At time zero, the number of shares of each stock in the basket is:

$$a_0 = \left( \frac{D \cdot \phi(1)}{S_1(0)}, \frac{D \cdot \phi(2)}{S_2(0)}, \cdots, \frac{D \cdot \phi(n)}{S_n(0)} \right).$$

At time $t_1$ (the first sampling time), we find

$$S = a_0(1)S_1(t_1) + a_0(2)S_2(t_1) + \cdots + a_0(n)S_n(t_1),$$

then we compute the new vector containing the number of shares of each stock in the basket:

$$a_1 = \left( \frac{S \cdot \phi(1)}{S_1(t_1)}, \frac{S \cdot \phi(2)}{S_2(t_1)}, \cdots, \frac{S \cdot \phi(n)}{S_n(t_1)} \right),$$

and so on, up to the last sampling time.

The number of shares of each stock in the basket is no longer a constant, moreover it is path depending —i.e. it depends on the values of the assets at each sampling date.
The finite-difference method is much more complicated to implement, therefore we choose to use the Monte-Carlo simulation method to compute the value of this option.

### 3.2 The Monte-Carlo Simulation Algorithm

The Monte Carlo simulation method uses the risk neutral valuation result. The expected payoff in a risk-neutral world is calculated using a sampling procedure. It is then discounted at the risk-free interest rate.

The algorithm proceeds as follows:

1. Simulate $M$ independent realizations $B_i$ of the final payoff $B$;

2. Choose $\left(\frac{1}{M}\sum_{i=1}^{M} B_1\right) e^{-rT}$ as an approximation for the option price.

The payoff $B$, for a lookback put basket option, is a function of the price processes $S_1(t), S_2(t), \cdots, S_n(t)$. Thus, to simulate $B$, we first have to simulate the paths $S_i(t)$ for $t \in [0,T]$ and $1 \leq i \leq n$. 

Suppose that the process followed by $\ln(S_i(t))$ in a risk-neutral world is:

$$d(\ln(S_i(t))) = \left( r - \frac{\sigma_i^2}{2}\right) dt + \sigma_i dW_i.$$ 

where $r$ is the risk-free interest rate and $\sigma_i$ is the standard deviation of $S_i$.

We divide the interval $[0,T]$, i.e. the life of the derivative, into $N$ subintervals and approximate the above equation as:

$$\ln \left( S_i \left( \frac{kT}{N}\right) \right) - \ln \left( S_i \left( \frac{(k-1)T}{N}\right) \right) = \left( r - \frac{\sigma_i^2}{2}\right) \frac{T}{N} + \sigma_i x_i \sqrt{\frac{T}{N}}.$$
or, equivalently:

$$S_i\left(\frac{k T}{N}\right) = S_i\left(\frac{(k - 1) T}{N}\right) \exp\left(\left(r - \frac{\sigma_i^2}{2}\right) \frac{T}{N} + \sigma_i x_i \sqrt{\frac{T}{N}}\right).$$

where $x_i$ is a random sample from a standard normal distribution.

The above equation enables the recursive computation of the values of $S_i(t)$ starting with the initial value $S_i(0)$, which is given.

If we consider a basket of $n$ assets $S_1, S_2, \ldots, S_n$, we have to generate $n \times N$ random numbers $X(i, k)$, $1 \leq i \leq n$, $1 \leq k \leq N$ which are $\mathcal{N}(0, 1)$-distributed such that:

$$\text{corr} (X(i, k), X(j, k)) = \text{corr} (S_i, S_j) \text{ for all } 1 \leq i, j \leq n \text{ and } 1 \leq k \leq N.$$ 

For each $k$, we generate $n$ standard normally distributed random variables $Y(1), Y(2), \ldots, Y(n)$ and then the numbers $X(i, k)$, $1 \leq i \leq n$, $1 \leq k \leq N$ are obtained as follows:

\[
X(1, k) = \varepsilon(1) \cdot Y(1),
\]
\[
X(2, k) = \varepsilon(2) \cdot Y(1) + \varepsilon(3) \cdot Y(2),
\]
\[
X(3, k) = \varepsilon(4) \cdot Y(1) + \varepsilon(5) \cdot Y(2) + \varepsilon(6) \cdot Y(3),
\]
\[
\vdots, \quad \vdots,
\]
\[
X(n, k) = \varepsilon\left(\frac{n(n - 1)}{2} + 1\right) Y(1) + \cdots + \varepsilon\left(\frac{n(n - 1)}{2} + n\right) Y(n).
\]
with:

\[ \varepsilon(1) - 1 = 0 \]

\[ (\varepsilon(2))^2 + (\varepsilon(3))^2 - 1 = 0 \]

\[ \vdots \]

\[ \left[ \varepsilon \left( \frac{n(n-1)}{2} + 1 \right) \right]^2 + \left[ \varepsilon \left( \frac{n(n-1)}{2} + 2 \right) \right]^2 + \cdots + \left[ \varepsilon \left( \frac{n(n-1)}{2} + n \right) \right]^2 - 1 = 0. \]

(for correct variance),

and:

\[ \varepsilon \left( \frac{i(i-1)}{2} + 1 \right) \varepsilon \left( \frac{j(j-1)}{2} + 1 \right) + \cdots + \varepsilon \left( \frac{i(i-1)}{2} + i \right) \varepsilon \left( \frac{j(j-1)}{2} + i \right) - \]

\[ - \text{corr} (S_i, S_j) = 0 \text{ for } i = 1, n-1, j = i+1, n. \]

(for correct correlations).

This system can be solved by using the MATLAB function \texttt{fsolve}, but first we have to transform the correlation matrix:

\[ (\text{corr}(S_i, S_j))_{1 \leq i, j \leq n} = (\rho_{ij})_{1 \leq i, j \leq n} \]

into a correlation vector:

\[ \text{corrvec} = (\rho_{12}, \rho_{13}, \cdots, \rho_{1n}, \rho_{23}, \rho_{24}, \cdots, \rho_{2n}, \cdots, \rho_{n-1n}), \]
with \( \frac{n(n-1)}{2} \) elements.

Then the system becomes:

\[
\left[ \varepsilon \left( \frac{i(i - 1)}{2} + 1 \right) \right]^2 + \left[ \varepsilon \left( \frac{i(i - 1)}{2} + 2 \right) \right]^2 + \cdots + \left[ \varepsilon \left( \frac{i(i - 1)}{2} + i \right) \right]^2 - 1 = 0
\]

for \( i = 1, n - 1, \)

\[
\varepsilon \left( \frac{i(i - 1)}{2} + 1 \right) \varepsilon \left( \frac{j(j - 1)}{2} + 1 \right) + \cdots + \varepsilon \left( \frac{i(i - 1)}{2} + i \right) \varepsilon \left( \frac{j(j - 1)}{2} + i \right) - \\
\text{corrvec} \left( n(i - 1) - \frac{(i - 1)i}{2} + i + j - 1 \right) = 0 \text{ for } i = 1, n - 1, j = i + 1, n.
\]

After solving the system, we find \( X(i, k) \) and then simulate the paths of the price processes:

\[
S_i \left( k \frac{T}{N} \right) = S_i \left( (k - 1) \frac{T}{N} \right) \cdot \exp \left[ \left( r - \frac{1}{2} \sigma^2(i) \right) \cdot \frac{T}{N} \right] \cdot \\
\cdot \exp \left[ \sigma(i) \cdot \sqrt{\frac{T}{N}} \cdot X(i, k) \right],
\]

for \( 1 \leq k \leq N. \)

Next, we compute the matrix of the number of shares of each stock in the basket at each sampling time with the algorithm described at the beginning of this chapter.

The maximum value of the basket is obtained as follows:

\[
\text{Max} = \max_{t \in \{t_1, t_2, \cdots, t_{\text{final}}\}} \left[ a(1, t)S_1(t) + a(2, t)S_2(t) + \cdots + a(n, t)S_n(t) \right];
\]

34
where \( \{t_1, t_2, \ldots, t_{\text{final}}\} \) is the set of sampling times.

The payoff becomes:

\[
B = \max (\text{Max} - \text{Basket value at } T, 0).
\]

Finally, we compute the option value:

\[
V = \exp(-rT) \cdot B.
\]

The number of simulation runs carried out depends on the accuracy required. If \( M \) is the number of runs and \( \omega \) is the standard deviation of the values of the option calculated from the simulation runs, the standard error of the estimate of \( V \) is \( \frac{\omega}{\sqrt{M}} \).

The advantage of the Monte-Carlo simulation method is that it is efficient when there are several variables involved. This is true since the time taken out to carry out a Monte-Carlo simulation increases approximately linearly with the number of variables, whereas the time required for most other procedures increases exponentially with the number of variables.

The Monte-Carlo simulation is also an approach that can accommodate complex payoffs, stochastic volatility and variable interest rates.
3.3 Numerical Example

Consider 4 assets with the standard deviation vector $\sigma$ and the percentage vector $\phi$ given by:

$$\sigma = \begin{pmatrix} .005 \\ .052 \\ .006 \\ .22 \end{pmatrix}, \quad \phi = \begin{pmatrix} .1 \\ .2 \\ .1 \\ .6 \end{pmatrix}.$$

The correlation matrix is:

$$\begin{bmatrix}
1 & .133 & .788 & .007 \\
.133 & 1 & .339 & .285 \\
.788 & .339 & 1 & .111 \\
.007 & .285 & .111 & 1 \\
\end{bmatrix}.$$

The other parameters are:

Current values of the assets: $S_1(0) = S_2(0) = S_3(0) = S_4(0) = 100$;

Current maximum: Max = 100;

Risk-free interest rate: $r = .1$;

Time to maturity: $T = 1$. 
Using the MATLAB function \texttt{lkb.m}, with source code in Appendix B, we obtain the following values for the lookback put basket option with rebalancing:

<table>
<thead>
<tr>
<th>Sampling times</th>
<th>Option value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Every month</td>
<td>11.25</td>
</tr>
<tr>
<td>Every two months</td>
<td>8.5</td>
</tr>
<tr>
<td>Every four months</td>
<td>6.3</td>
</tr>
<tr>
<td>No sampling</td>
<td>3.8</td>
</tr>
</tbody>
</table>

Figure 3.1: Lookback Put with Rebalancing
CHAPTER 4
GUARANTEED MINIMUM DEATH BENEFITS

4.1 Introduction

The multiple decrement model provides a framework for studying many financial security systems. For example, life insurance policies frequently provide for special benefits if death occurs by accidental means or if the insured becomes disabled. Another major application is in pension plans. A plan, upon a participant’s retirement, typically provides pensions for age in service or for disability. In case of withdrawal from employment, there can be a return of accumulated participant contributions or a deferred pension. For death occurring before the other contingencies, there could be a lump sum or income payable to a beneficiary.
In general, actuarial applications of multiple decrement models arise when the amount of benefit payment depends on the mode of exit from the group of active lives.

The Individual Variable Annuity marketplace is highly competitive. One of the key competitive elements is the death benefit.

Examples of currently sold death benefits are:

**Return of premium (ROP):** the death benefit is the greatest of the account value and the net premium paid;

**Reset:** the death benefit is the greatest of the account value, the net premium paid and the highest account value on the last five anniversaries;

**Ratchet:** the death benefit is the greatest of the account value, the net premium paid and the highest account value on all past anniversaries.

In this chapter, we will analyze these death benefits using the results obtained in Chapter 2 and Chapter 3.

### 4.2 Survival Distributions

#### 4.2.1 The Survival Function

Consider a newborn child. The newborn’s age at death, $X$, is a continuous random variable. Let $F_X(x)$ be the distribution function of $X$,

$$F_X(x) = \Pr(X \leq x), \text{ for } x \geq 0,$$
and set:

\[ s(x) = 1 - F_X(x) = \Pr(X > x), \text{ for } x \geq 0. \]

We always assume that \( F_X(0) = 0 \), which implies \( s(0) = 1 \). The function \( s(x) \) is called the survival function. For any positive \( x \), \( s(x) \) is the probability that a newborn will attain age \( x \).

### 4.2.2 Time-until-death for a Person Age \( x \)

The symbol \( (x) \) is used to denote a life-age-\( x \). The future lifetime of \( (x) \), \( X - x \), is denoted by \( T(x) \). In order to make probability statements about \( T(x) \), the following notation is used:

\[
\begin{align*}
\mathit tq_x &= \Pr[T(x) \leq t], & t \geq 0, \\
\mathit tp_x &= 1 - \mathit tq_x &= \Pr[T(x) > t], & t \geq 0.
\end{align*}
\]

The symbol \( \mathit tq_x \) is interpreted as the probability that \( (x) \) will die within \( t \) years and \( \mathit tp_x \) is the probability that \( (x) \) will attain age \( x + t \).

In the special case of a life-age-zero, we have \( T(0) = X \) and \( x_p_0 = s(x), x \geq 0 \). If \( t = 1 \), convention permits us to omit the prefix in the symbols defined above:

\[
\begin{align*}
\mathit q_x &= \Pr[(x) \text{ will die within one year}], \\
\mathit p_x &= \Pr[(x) \text{ will attain age } x + 1].
\end{align*}
\]

### 4.2.3 Force of Mortality

Let \( f_X(x) \) be the probability density function of \( X \), the continuous age-at-death random variable. The expression:

\[
\frac{f_X(x)}{1 - F_X(x)}
\]
has a conditional probability density interpretation. For each age $x$, the expression gives the value of the conditional probability density function of $X$ at exact age $x$, given the survival to that age. It will be denoted by $\mu(x)$ and referred to as the force of mortality. We can write:

$$
\mu(x) = \frac{f_X(x)}{1 - F_X(x)} = \frac{F'_X(x)}{1 - F_X(x)} = \frac{-s'(x)}{s(x)}
$$

$$
x p_0 = s(x) = \exp \left[ -\int_0^x \mu(s) ds \right].
$$

### 4.2.4 Constant Force of Mortality

A widely used assumption for fractional ages is linear interpolation on $\log s(x+t)$:

$$
\log s(x+t) = (1-t) \log s(x) + t \log s(x+1), \text{ for } 0 \leq t < 1.
$$

Under this assumption, $x p_x$ is exponential:

$$
x p_x = (p_x)^t,
$$

and:

$$
\mu(x+t) = \mu(x) = -\log p_x, \text{ for } 0 \leq t < 1.
$$

### 4.3 Multiple Decrement Models

Consider two random variables: $T$, time-until-termination form a status, and $J$, cause of decrement. Assume that $J$ is a discrete random variable. We denote the joint probability density function of $T$ and $J$ by $f_{T,J}(t,j)$,
the marginal probability density function of $J$ by $f_J(j)$ and the marginal probability density function of $T$ by $f_T(t)$.

The following equalities hold:

\[
\sum_{j=1}^{m} f_J(j) = 1
\]
\[
\int_{0}^{\infty} f_T(t)dt = 1.
\]

Then the probability of decrement due to cause $j$ before time $t$ is:

\[
\Pr [(0 < T \leq t) \cap (J = j)] = \int_{0}^{t} f_{T,J}(s, j)ds = t q^{(j)},
\]

and the probability of decrement due to all causes between $a$ and $b$ is:

\[
\Pr [a < T \leq b] = \sum_{j=1}^{m} \int_{a}^{b} f_{T,J}(t, j)dt.
\]

The probability of decrement due to cause $j$ at any time $t$ in the future is:

\[
f_J(j) = \int_{0}^{\infty} f_{T,J}(s, j)ds = \infty q^{(j)}, \ j = 1, 2, \ldots, m.
\]

For $t \geq 0$, the marginal probability density function for $T$, $f_T(t)$, and the density function $F_T(t)$ are:

\[
f_T(t) = \sum_{j=1}^{m} f_{T,J}(t, j),
\]
\[
F_T(t) = \int_{0}^{t} f_T(s)ds.
\]
Using the superscript \((\tau)\) to indicate that a function refers to all causes, or total force of decrement, we obtain:

\[
\begin{align*}
\iota q_x^{(\tau)} &= \Pr(T \leq t) = F_T(t), \\
\iota p_x^{(\tau)} &= \Pr(T > t) = 1 - \iota q_x^{(\tau)}, \\
\mu_x^{(\tau)} &= \frac{f_T(t)}{1 - F_T(t)} = -\frac{d}{dt} \log \iota p_x^{(\tau)}, \\
\end{align*}
\]

\[
f_{T,J}(t, j)dt = \Pr[T > t] \cdot \Pr[(t < T \leq t + dt) \cap (J = j) | T > t].
\]

This suggests the following definition for the force of decrement due to cause \(j\):

\[
\mu_x^{(j)}(t) = \frac{f_{T,J}(t, j)}{1 - F_T(t)} = \frac{f_{T,J}(t, j)}{\iota p_x^{(\tau)}} = \frac{d}{dt} q_x^{(j)} / \iota p_x^{(\tau)}.
\]

So:

\[
f_{T,J}(t, j)dt = \iota p_x^{(\tau)} \mu_x^{(j)}(t)dt, \quad j = 1, 2, \ldots, m \text{ and } t \geq 0.
\]

Restated, the last equality implies that the probability of decrement between \(t\) and \(t + dt\) due to cause \(j\) is equal to the probability \(\iota p_x^{(\tau)}\) that \((x)\) remains in the given status until time \(t\) times the conditional probability \(\mu_x^{(j)}(t)\) that decrement occurs between \(t\) and \(t + dt\) due to cause \(j\), given that the decrement has not occurred before time \(t\).

Also we have:

\[
\iota q_x^{(\tau)} = \int_0^t f_T(s)ds = \int_0^t \sum_{j=1}^m f_{T,J}(s, j)ds = \sum_{j=1}^m \int_0^t f_{T,J}(s, j)ds = \sum_{j=1}^m \iota q_x^{(j)}.
\]
and it follows that:

\[ \mu_x^{(\tau)}(t) = \sum_{j=1}^{m} \mu_x^{(j)}(t), \]

that is, the total force of decrement is the sum of the forces of decrement due to all causes.

### 4.4 Associated Single Decrement Tables

For each of the causes of decrement recognized in a multiple decrement model, it is possible to define a single decrement model that depends only on the particular cause of decrement. The associated single decrement model functions are defined as follows:

\[ t_p^{(j)}(x) = \exp \left[ - \int_{0}^{t} \mu_x^{(j)}(s) ds \right], \]

\[ t_q^{(j)}(x) = 1 - t_p^{(j)}(x) \quad \text{(absolute rate of decrement)}. \]

Note that:

\[ t_p^{(\tau)}(x) = \prod_{j=1}^{m} t_p^{(j)}(x) \quad \text{and} \]

\[ t_p^{(j)}(x) \geq t_p^{(\tau)}(x). \]

The magnitude of other forces of decrement can cause \( t_p^{(j)}(x) \) to be considerably greater than \( t_p^{(\tau)}(x) \).
For single decrement models, constant-force assumption implies:

\[
\begin{align*}
\mu_x^{(j)}(t) &= \mu_x^{(j)}(0), \\
\mu_x^{(r)}(t) &= \mu_x^{(r)}(0), \\
p_x^{(j)}(t) &= (p_x^{(j)})^t, \\
\mu_x^{(j)}(0) &= -\ln(p_x^{(j)}),
\end{align*}
\]

for \(0 \leq t < 1\).

4.5 Guaranteed Minimum Death Benefits

A premium is paid at time zero and the money is invested in a basket of assets. The account value at time \(t\) is the value of the basket at time \(t\).

The first basic guaranteed minimum death benefit (GMDB) is Return of Premium (ROP), for which the death benefit is the greatest of the account value and the net premium paid.

The amount that the insurance company has to cover is:

\[
\max (\text{Premium} - \text{Account Value at death}, 0).
\]

But this is exactly the payoff for a vanilla put option with the exercise price equal to the premium. The present value of the death benefit paid is equal to the price of the option at time zero.

The second basic GMDB is Reset, for which the death benefit is the greatest of the account value, the net premium paid and the highest account value on the last five anniversaries.
The amount that the insurance company has to cover is:

$$\max (\max (\text{Premium}, \text{Account Values on the last five anniversaries}) - \text{Account Value at death}, 0).$$

This is the same as the payoff for a lookback put option with discretely measured maximum, at the beginning of the period and every year for the past five years.

The third basic GMDB is Ratchet, for which the death benefit is the greatest of the account value, the net premium paid and the highest account value on all past anniversaries.

The amount that the insurance company has to cover is:

$$\max (\max (\text{Premium}, \text{Account Values on all past anniversaries}) - \text{Account Value at death}, 0).$$

This is the same as the payoff for a lookback put option with maximum measured discretely every year.

The most important problem for an insurance company is to compute the present value of the claims over some evaluation period.

When the amount of benefit payment depends on the mode of exit from the group of active lives, we have to use multiple decrement models.
Let $B(j, t)$ be the present value of a benefit at age $(x + t)$ incurred by a decrement at that age by cause $j$. Then the actuarial present value of the benefits that occur between times $t_1$ and $t_2$ is given by:

$$A = \sum_{j=1}^{m} \int_{t_1}^{t_2} B(j, t) \, t^{(\tau)} p_x^{(\tau)} \mu_x^{(j)}(t) \, dt.$$ 

For example, in the case of death benefits, three causes of decrement are usually considered:

1. Mortality;
2. Lapse;
3. Partial withdrawal.

A benefit is paid only at death. The present value function of the benefit payment, $B(1, t)$, is equal to the price of an option (vanilla put for ROP and lookback put for Reset and Ratchet), with expiration date $t$.

The annual mortality, lapse and partial withdrawal tables are given.

Below is a computation of the present value of the claims over the evaluation period $[t_1, t_2]$, for $1 \leq j \leq m$ causes of decrement.

We have, for $t_2 \geq t_1 + 1$:

$$A = \sum_{j=1}^{m} \int_{t_1}^{t_2} B(t, j) p_x^{(\tau)} \mu_x^{(j)}(t) \, dt =$$

$$= \sum_{j=1}^{m} \left( \sum_{k=0}^{[t_2]-[t_1]-1} \int_{[t_1]+k}^{[t_1]+k+1} B(t, j) p_x^{(\tau)} \mu_x^{(j)}(t) \, dt - \int_{[t_1]} \int_{[t_1]} B(t, j) p_x^{(\tau)} \mu_x^{(j)}(t) \, dt + \int_{[t_2]} B(t, j) p_x^{(\tau)} \mu_x^{(j)}(t) \, dt \right).$$
Define:

\[ A(k, j) = \int_{[t_1] + k}^{[t_1] + k + 1} B(t, j) p_x^{(r)} \mu_x^{(j)}(t) dt, \quad \text{for } 0 \leq k \leq [t_2] - [t_1] - 1, \]
\[ A(t_1, j) = \int_{[t_1]}^{t_1} B(t, j) p_x^{(r)} \mu_x^{(j)}(t) dt, \]
\[ A(t_2, j) = \int_{[t_2]}^{t_2} B(t, j) p_x^{(r)} \mu_x^{(j)}(t) dt. \]

We have:

\[ A(k, j) = [t_1] + k p_x^{(r)} \int_0^1 B([t_1] + k + t, j) p_x^{(r)} \mu_x^{(j)}(t) dt, \]
\[ A(t_1, j) = [t_1] p_x^{(r)} \int_{[t_1]}^{t_1 - [t_1]} B([t_1] + t, j) p_x^{(r)} \mu_x^{(j)}(t) dt, \]
\[ A(t_2, j) = [t_2] p_x^{(r)} \int_{[t_2]}^{t_2 - [t_2]} B([t_2] + t, j) p_x^{(r)} \mu_x^{(j)}(t) dt. \]

Next, we assume constant force of mortality for all decrements, so we can write:

\[ \mu_x^{(j)}(t) = -\ln(p_x^{(j)}) \]
\[ q^{(j)} = (p_x^{(j)})^t, \]

for \( 0 \leq t < 1. \)
Then $A(k, j)$, $A(t_1, j)$ and $A(t_2, j)$ become:

$$A(k, j) = \prod_{j=1}^{m} \left(p_x^{(j)} p_{x+1}^{(j)} \cdots p_{x+[t_1]+k-1}^{(j)}\right) \cdot$$

$$\cdot \int_0^1 B([t_1] + k + t, j) \cdot \prod_{j=1}^{m} \left(p_{x+[t_1]+k}^{(j)}\right)^t \cdot \left[-\ln \left(p_{x+[t_1]+k}^{(j)}\right]\right] dt,$$

$$A(t_1, j) = \prod_{j=1}^{m} \left(p_x^{(j)} p_{x+1}^{(j)} \cdots p_{x+[t_1]-1}^{(j)}\right) \cdot$$

$$\cdot \int_0^1 B([t_1] + t, j) \cdot \prod_{j=1}^{m} \left(p_{x+[t_1]}^{(j)}\right)^t \cdot \left[-\ln \left(p_{x+[t_1]}^{(j)}\right]\right] dt,$$

$$A(t_2, j) = \prod_{j=1}^{m} \left(p_x^{(j)} p_{x+1}^{(j)} \cdots p_{x+[t_2]-1}^{(j)}\right) \cdot$$

$$\cdot \int_0^1 B([t_2] + t, j) \cdot \prod_{j=1}^{m} \left(p_{x+[t_2]}^{(j)}\right)^t \cdot \left[-\ln \left(p_{x+[t_2]}^{(j)}\right]\right] dt.$$

We obtain:

$$A = \sum_{j=1}^{m} \left(\sum_{k=0}^{[t_2]-[t_1]-1} A(k, j) - A(t_1, k) + A(t_2, j)\right).$$

### 4.6 Numerical Example

Consider a basket consisting of two assets with the following parameters:

**Current asset prices:** $S_1(0) = S_2(0) = 1000$;

**Current maximum:** $\text{Max} = 1000$;

**Number of shares:** $\alpha_1 = .3$, $\alpha_2 = .7$;
Standard deviation vector: $\sigma_1 = \sigma_2 = .22$;

Asset correlation: $\rho = .1$;

Risk-free interest rate: $r = .1$;

Evaluation period: $t_1 = 0$, $t_2 = 10$ years.

Using the MATLAB function `pv.m`, with source code in Appendix C, we obtain the following values of the claim over the next 10 years:

<table>
<thead>
<tr>
<th>Sampling times</th>
<th>Claims</th>
</tr>
</thead>
<tbody>
<tr>
<td>Every year</td>
<td>27.1163</td>
</tr>
<tr>
<td>Every two years</td>
<td>22.1736</td>
</tr>
<tr>
<td>Every four years</td>
<td>18.2614</td>
</tr>
<tr>
<td>No sampling</td>
<td>10.2379</td>
</tr>
</tbody>
</table>

Figure 4.1: Claims Values over 10 Years

Therefore, for $1000$ invested today in the basket of two assets described above, the expected value of the claims covered by the insurance company for the next 10 years is about $27$ in the case of Ratchet and $10$ for ROP.
CHAPTER 5
CONCLUSION AND
FURTHER RESEARCH

As we have seen in the introductory chapter, using the explicit finite differences method for solving partial differential equations puts severe constraints on the size of the time step.

One way to overcome this problem is to use *implicit finite differences* schemes for solving partial differential equations, such as the *Alternating Direction Implicit* method. This approach proceeds as follows:

1. Introduce an “intermediate” value $V(i, j, k + \frac{1}{2})$;

2. Solve from time-step $k$ to the intermediate time-step $k + \frac{1}{2}$ using explicit differences in $S_1$ and implicit differences in $S_2$;

3. Having found the intermediate value $V(i, j, k + \frac{1}{2})$, step forward to time-step $k + 1$ using implicit differences in $S_1$ and explicit differences in $S_2$.
This method is stable for all time-steps and the error is $O(dt^2, dS_1^2, dS_2^2)$. The only problem is that the cross derivative term $\frac{\partial^2 V}{\partial S_1 \partial S_2}$ causes lots of problems in the basic implementation of the ADI, and the Black-Scholes equation cannot be changed into the equation:

$$\frac{\partial V}{\partial t} + a(s_1, s_2, t)\frac{\partial^2 V}{\partial s_1^2} + b(s_1, s_2, t)\frac{\partial^2 V}{\partial s_2^2} = 0,$$

unless using substitutions of the form:

$$s_1 = x_1 S_1 + x_2 S_2$$

$$s_2 = y_1 S_1 + y_2 S_2$$

with $x_1, y_1 > 0$ and $x_2, y_2 < 0$, in which case the boundary condition at $S_1 \to \infty$, $S_2 \to \infty$ becomes impossible to implement.

We are still looking for an implicit finite differences method that can be easy implemented in the case of a basket option.

The finite differences methods are suitable for solving financial problems with two or three random factors. For more than that number, the Monte Carlo simulation becomes a better method, which also works in the case of complex payoffs.

The discrete lookback options are more difficult to hedge than regular options because the delta of the option is discontinuous at all sampling times.

An alternative approach that may be used to hedge a position in exotic options is *Static Option Replication*. This approach involves searching for a portfolio of actively traded options that approximately replicate the option position. Shorting this position provides the hedge. The basic principle used
is: if two portfolios are worth the same on a certain boundary, they are worth the same at all interior points of the boundary.

Unfortunately, we were unable to find a general form of a portfolio that can be used for static replication in the case of lookback options.
APPENDIX A

THE SOURCE CODE OF

LOOKBASKET.M

function
f=lookbackbasket(S1,S2,M,sigma1,sigma2,r,q1,q2,ro,T,alpha1,
alpha2,tbeg,tinc)

%The lookbackbasket function returns the value of a lookback put
%on a basket with two assets using the finite-difference method.

% S1 Current price of the first asset
% S2 Current price of the second asset
% M Current max of the basket price
% sigma1 SD for the first asset
% sigma2 SD for the second asset
% r Risk-free interest rate
% q1 Divided rate of the first asset
% q2   Divided rate of the second asset
% ro   Correlation coefficient between the two assets
% T    Time to maturity
% alpha1 Number of shares of the first asset in the basket
% alpha2 Number of shares of the second asset in the basket
% tbeg  First sampling time
% tinc  Time between two consecutive samplings

n=floor((T-tbeg)/tinc)+2;
I=100; % The number of points in the mesh for s1
J=100; % The number of points in the mesh for s2
p=10;
K=p*n; % The number of points in the mesh for t
ds1=4/I; ds2=4/J; dT=T/K;
a=zeros(I+1);
for i=1:(I+1)
    a(i)=0.5*(sigma1^2)*((i-1)*ds1)^2;
end;

b=zeros(I+1);
for i=1:(I+1)
    b(i)=(r-q1)*((i-1)*ds1);
end;

c=-r;
d=zeros(J+1);

for j=1:(J+1)
    d(j)=0.5*(sigma2^2)*(((j-1)*ds2)^2);
end;

e=zeros(I+1,J+1);

for i=1:(I+1)
    for j=1:(J+1)
        e(i,j)=ro*sigma1*sigma2*((i-1)*ds1)*((j-1)*ds2);
    end;
end;

f=zeros(J+1);

for j=1:(J+1)
    f(j)=(r-q2)*(j-1)*ds2;
end;

v=zeros(I+1,J+1,K+1);

for i=1:(n+1)
    w(i)=(i-1)*p;
end;
slin1=linspace(0,4,I+1);
slin2=linspace(0,4,J+1);

%rule 3 Boundary condition at s1=0, s2=0
for k=2:(K+1)
    v(1,1,k)=exp(-r*(k-1)*dT);
end;

z=v(1,1,K+1);
%end rule 3

%start main cycle
for l=1:n

% rule2: The final condition and The "jumping"condition
for i=1:(I+1)
    for j=1:(J+1)
        if (l==1)
            v(i,j,1)=max(1-(alpha1*(i-1)*ds1+alpha2*(j-1)*ds2),0);
        end;
        if (l>1)
            mx=max(1,alpha1*(i-1)*ds1+alpha2*(j-1)*ds2);
            v(i,j,w(l)+1)=mx*interp2(slin1,slin2,u(:,:,w(l)+1),
                (i-1)*ds1/mx,(j-1)*ds2/mx);
        end;
    end;
end;

end;
end;
end;
%end rule 2

for k=(w(l)+1):w(l+1)

%rule 1: The PDE
for i=2:I
    for j=2:J
        sum1=a(i)*(v(i+1,j,k)-2*v(i,j,k)+v(i-1,j,k))/(ds1^2);
        sum2=b(i)*(v(i+1,j,k)-v(i-1,j,k))/(2*ds1);
        sum3=c*v(i,j,k);
        sum4=d(j)*(v(i,j+1,k)-2*v(i,j,k)+v(i,j-1,k))/(ds2^2);
        sum5=e(i,j)*(v(i+1,j+1,k)-v(i+1,j-1,k)-v(i-1,j+1,k)+v(i-1,j-1,k))/(4*ds1*ds2);
        sum6=f(j)*(v(i,j+1,k)-v(i,j-1,k))/(2*ds2);
        v(i,j,k+1)=dT*(v(i,j,k)/dT+sum1+sum2+sum3+sum4+sum5+sum6);
    end;
end;
%end rule 1

%rule 4: Boundary condition at s1=0
for \( j=2:J \)
\[
\begin{align*}
\text{sum1} &= \frac{v(1,j,k)}{dT}; \\
\text{sum2} &= \frac{d(j) \cdot (v(1,j+1,k)-2 \cdot v(1,j,k)+v(1,j-1,k))}{(ds2^2)}; \\
\text{sum3} &= \frac{f(j) \cdot (v(1,j+1,k)-v(1,j-1,k))}{2 \cdot ds2}; \\
\text{sum4} &= c \cdot v(1,j,k); \\
v_{(1,j,k+1)} &= dT \cdot (\text{sum1} + \text{sum2} + \text{sum3} + \text{sum4}); \\
\end{align*}
\]
end;

%end rule 4

%rule 5: Boundary condition at \( s2=0 \)
for \( i=2:I \)
\[
\begin{align*}
\text{sum1} &= \frac{v(i,1,k)}{dT}; \\
\text{sum2} &= \frac{a(i) \cdot (v(i+1,1,k)-2 \cdot v(i,1,k)+v(i-1,1,k))}{(ds1^2)}; \\
\text{sum3} &= \frac{b(i) \cdot (v(i+1,1,k)-v(i-1,1,k))}{2 \cdot ds1}; \\
\text{sum4} &= c \cdot v(i,1,k); \\
v_{(i,1,k+1)} &= dT \cdot (\text{sum1} + \text{sum2} + \text{sum3} + \text{sum4}); \\
\end{align*}
\]
end;

%end rule 5

%rule 6 Boundary condition at \( s1=\text{max} \)
for \( j=1:J \)
\[
\begin{align*}
v_{(I+1,j,k+1)} &= \frac{I}{(I-1)} \cdot v_{(I,j,k+1)}; \\
\end{align*}
\]
end;

%end rule 6
%rule 7: Boundary condition at s2=max
for i=1:I
    v(i,J+1,k+1)=J/(J-1)*v(i,J,k+1);
end;
%end rule 7

%rule 8: Boundary condition at s1=max, s2=max
v(I+1,J+1,k+1)=I/(I+J-1)*v(I,J+1,k+1)+J/(I+J-1)*v(I+1,J,k+1);
%end rule 8

end;

u=zeros(I+1,J+1,K+1);
u=v;
end;

z=v(:,:,K+1);
f=interp2(slin1,slin2,z,S1/M,S2/M)*M;
APPENDIX B

THE SOURCE CODE OF

LKB.M

function f=lkb(stocks, sigma, T,inam,fi,t1,dt,r,corr)

%stocks Vector containing the current prices of the assets
%sigma Vector containing the standard deviations of the assets
%T Time to maturity
%inam Initial amount invested
%fi Vector containing the percentages invested in each stock
%t1 First sampling date
%dt Time between two consecutive samplings
%r Risk-free interest rate
%corr Correlation vector

global S;
global T;
global N;
global n;
global corr;

e0=ones(floor(n*(n+1)/2),1);
e=e0;  

N=10; n=length(stocks);
for k=1:N Y=randn(1,n);
for i=1:n s=0; for
   j=1:i ind=floor(i*(i-1)/2+j);
   s=s+e(ind)*Y(j);
end; X(i,k)=s;
end;
end;

for i=1:n S(i,1)=stocks(i);
for k=2:(N+1)
   S(i,k)=S(i,k-1)*exp((r-(sigma(i)^2/2)*T/N+sigma(i)*
   sqrt(T/N)*X(i,k-1));
end;
end;

maux=(T-t1)/dt;
if (floor(maux)==maux) M=floor(maux)-1;
end;

if (floor(maux)~=maux) M=floor(maux);
end;

tau(1)=t1;
for j=2:(M+1)
    tau(j)=t1+(j-1)*dt;
end;

for i=1:n
    a(i,1)=inam*fi(i)/S(i,1);
    for k=2:(M+2)
        s=0;
        for j=1:n
            s=s+a(j,k-1)*SS(j,tau(k-1));
        end;
        a(i,k)=fi(i)*s/SS(i,tau(k-1));
    end;
end;

for j=1:(M+1)
    s=0;
    for i=1:n
        s=s+a(i,j+1)*SS(i,tau(j));
    end;
end;
end;
v(j)=s;
end;
MAX=max(v);

s=0;
for i=1:n
    s=s+a(i,M+2)*SS(i,T);
end;

payoff=max(MAX-s,0);

optionvalue=exp(-r*T)*payoff; f=optionvalue;

function f1=SS(i,t)
global S;
global T;
global N;
m=floor(t*N/T);
f1=interp1(linspace(0,T,N+1),S(i,:),t);

function f2=randomk(e) g
global n;
global corr;
g=[];
for i=1:n
    s=0;
    for j=1:i
        s=s+(e(floor(i*(i-1)/2+j)))^2;
    end;
    gaux=s-1;
    g=[g;gaux];
end;

for i=1:(n-1)
    for j=(i+1):n
        s=0;
        for k=1:i
            ii=floor(i*(i-1)/2);
            ij=floor(j*(j-1)/2);
            s=s+e(ii+k)*e(ij+k);
        end;
        gaux=s-corr(n*(i-1)-floor((i-1)*i/2)+j);
        g=[g;gaux];
    end;
end;

f2=g;
APPENDIX C

THE SOURCE CODE OF PV.M

function res=pv(t1,t2)

%t1 Beginning of the evaluation period
%t2 End of the evaluation period
%N Length of the mortality vector
%m Mortality vector
%l Lapse vector
%pw Partial withdrawal vector

global k;
global t1;
global t2;
global px1;
global px2;
global px3;
N=10;
m=[0.000001 0.000001 0.000001 0.000001 0.000324 0.000301
0.000286 0.000328 0.000362 0.00039];
l= [0.02 0.03 0.05 0.07 0.10
0.12 0.20 0.17 0.17 0.17];
pw= [0.021 0.022 0.023 0.024 0.025
0.026 0.027 0.028 0.029 0.030];

for i=1:length(m)
    px1(i)=1-m(i);
end;

for i=1:length(l)
    px2(i)=1-l(i);
end;

for i=1:length(pw)
    px3(i)=1-pw(i);
end;

A=zeros(floor(t2)-floor(t1));

for k=0:(floor(t2)-floor(t1)-1)
    P1=1;
for i=1:(floor(t1)+k)
    P1=P1*px1(i);
end;

P2=1;
for i=1:(floor(t1)+k)
    P2=P2*px2(i);
end;

P3=1;
for i=1:(floor(t1)+k)
    P3=P3*px3(i);
end;

A(k+1)=P1*P2*P3*quad(@(intfun, 0, 1);
end;

P1=1;
for i=1:floor(t1)
    P1=P1*px1(i);
end;

P2=1;
for i=1:floor(t1)
    P2=P2*px2(i);
end;
P3=1;

for i=1:floor(t1)
    P3=P3*px3(i);
end;

k=0;
At1=P1*P2*P3*quad(@intfun,0,t1-floor(t1));
P1=1;

for i=1:floor(t2)
    P1=P1*px1(i);
end;

P2=1;

for i=1:floor(t2)
    P2=P2*px2(i);
end;

P3=1;

for i=1:floor(t2)
    P3=P3*px3(i);
end;

k=floor(t2)-floor(t1);
At2=P1*P2*P3*quad(@intfun,0,t2-floor(t2));
s=0;

for i=1:(floor(t2)-floor(t1))
    s=s+A(i);
end;

res=s-At1+At2;

function fp=intfun(t)

global k;
global t1;
global t2;
global px1;
global px2;
global px3;

fp=[];

for i=1:length(t)
c1=lookbackbasket(S1,S2,M,sigma1,sigma2,r,q1,q2,ro,floor(t1) +k+t(i),alpha1,alpha2,tbeg,tinc);
c2=(px1(floor(t1)+k+1)*px2(floor(t1)+k+1)*px3(floor(t1)+k+1)) ^t(i); c3=-log(px1(floor(t1)+k+1));
c=c1*c2*c3;
fp=[fp c];
end;
BIBLIOGRAPHY


