Interest Rate Scenario Generation for Stochastic Programming

Author: Omri Ross s041369

September 14, 2007

Supervisors: Professor Jens Clausen, PhD Student Kourosh Marjani Rasmussen

The Section of Operations Research

Informatics and Mathematical Modelling

The Technical University of Denmark (DTU)

Table of Contents

List of Figures

To Elad...

Acknowledgements

Writing this thesis has been a memorable voyage alive with abundant interesting ideas and challenging tasks. It would not have been the same without the numerous insightful discussions, erudite advice and mentoring of my supervisors Professor Jens Clausen and PhD student Kourosh Marjani Rasmussen to whom I am greatly thankful and indebted.

This journey was supported and encouraged by many who have kindly assisted me. In particular, I would like to thank the treasury team at Nykredit, especially Kenneth Styrbæk and Michael Ager Carlsen, for sharing information and insight about the problem discussed in this paper. I would also like to convey my thanks to Dr. Michal Kaut, Professsor Ronald Hochreiter, Professor Rolf Poulsen, Snorri Pall Sigurdsson and Arngrimur Einarsson for the supporting information, data and in depth discussions they provided along the way. In addition, I would like to express my gratitude to The Josef and Regine Nachemsohn Fund for partial financial support provided along this exciting trip. Sincere appreciation to Nechama Golan for reviewing my writing, who with boundless energy, kept me awake asking for clarification and making useful corrections. Finally yet importantly, I want to thank my family Tzlil, Elad, Ofra and Arie for their endless support and my wonderful girlfriend Helle Gleie Damgaard, who have supported, inspired and tolerated me throughout.

Preface

This thesis fulfills the final requirement in order to obtain a Master of Science degree in Computing and Mathematics from the Technical University of Denmark (DTU). It was carried out in the Operations Research Section of the Informatics and Mathematical Modelling Department of DTU from October 1, 2006 through September 4, 2007 under the supervision of Professor Jens Clausen and PhD student Kourosh Marjani Rasmussen.

Chapter 1

Executive Summary

Research Motivation and State of the Art

Representing uncertainty in models for decision making under uncertainty poses a significant challenge. The mortgagor's selection problem is typical of the conflict most homebuyers experience when purchasing a house. In Denmark, a mortgagor can finance up to 80% of the property value by issuing mortgage-backed securities from a mortgage bank. The variety of mortgage-backed securities available in some countries (such as Denmark) leads to a great variety of finance options for a house buyer. Nielsen and Poulsen in [10] suggested a two-factor, arbitrage-free interest-rate model, calibrated to observable security prices, and implement on top of it a multi-stage, stochastic optimization program with the purpose of optimally composing and managing a typical mortgage loan. Rasmussen and Clausen in [11] formulated multi-stage integer programs of the problem, and used scenario reduction and LP relaxations to obtain near optimal results. Their research suggests both market and wealth risks of the problem and suggests a more efficient utility function. A Conditional Value-at-Risk (CVaR) model was suggested by Rasmussen & Zenios in [12], [13] as well as a thorough examination of the value received by most risk averse homeowners who consider a diversified portfolio of both fixed (FRM) and adjustable (ARM) rate mortgages.

All the different calculations done by these mathematical models are based on future prices of diverse bonds. These prices are heavily dependent on different future realizations of the interest rates. A more elaborate model of the interest rate scenario generation can be used to increase the quality of the solution.

This report explores and implements different scenario generation methods for representing the interest rates. The research is mainly based on moment matching approaches as represented in Højland and Wallace in [4] and followed by Højland, Kaut and Wallace in [5],[6] . These approaches are later used as part of a vector autoregressive with leg 1 (VAR1) interest rate model to create other interest rate models that are suitable for the financial industry. Thereby creating arbitrage–free scenarios that are consistent with literature regarding financial properties such as factor analysis of the term structure (as observed by Litterman & Scheinkman at [42], and explored by Rasmussen and Poulsen at [39], Dahl [43] and Zenios at [14]).

The use of interest rates scenarios generations that are explored in this report can be extended to be used with any financial framework. Moreover, the scenario generation approaches can be used for general stochastic programming models outside the financial industry.

Research done as part of this report

The project was done in collaboration with Nykredit Denmark as part of the creation of a scenario generator for the Danish mortgagor problem as described above. The actual writing of this report took into consideration that concepts needed to be explained one step at a time. This report is structured in the following manner: first, an introductory chapter in which the motivation and main concepts for using scenario generation for stochastic programming problems is presented. Different scenario generation methods and quality criteria are put forward in the next chapter so as to better understand the reasoning behind the choice of different scenario generation heuristics. This is important for setting the groundwork when searching for an appropriate scenario generation approach. The moment matching approach for scenario generation is then described in detail expounding on two heuristics for moment matching scenario generations in chapter 4. The challenge of modeling arises from the need to extend the existing scenario generation methodology to deal with financial challenges as part of interest rate scenario generation. In chapter 5 the interest rate risk is introduced and financial concerns associated with interest rate scenario generation, such as arbitrage detection, factor analysis of the term structure and smoothing of the yield curve are examined and comprehensively explored. A proposed solution that creates an accurate and consistent model for scenario generation from the mathematical standpoint, based on the latest stochastic programming trends that incorporates correctness of interest rate modeling from the financial perspective is shown in chapter 6. The chapter presents a general framework for interest rate scenario generation and introduces a concrete formulation of a model for interest rate scenario generation. Chapter 7 explores the results of running the suggested interest rate scenario generation model with a different variation based on different time points, scenario generation strategies, yield curve smoothing methods, and the like, as well as a comparison between a 1–factor Vasicek model to the 3–factor model presented in chapter 6. The results are very promising and show numerous possible advantages by using a specific scenario generation approach that is designed for interest rate scenario generation. The conclusion of the research findings and contributions finalize this master thesis report and briefly describe future aspirations of its author.

Personal Summary

Being involved in a financial engineering research project that is in use in the financial industry and deals with day-to-day practical issues in addition to theoretical research provided an excellent opportunity for this author to learn and apply knowledge acquired. The project included looking into alternative technologies as part of the scenario generation creation as well as their execution. Since the practical implementations are used by Nykredit, technical appendixes have been left out. Some of the discussions offered in this report were done on a research level, put into action on the practical level and then presented directly as results, excluding interesting implementation analysis. Having said that, a significant amount of time was spent utilizing different methodologies that unfortunately cannot be incorporated as part of this thesis.

I believe that today risk management is more relevant than ever. Take into consideration the most recent sub-prime mortgage crisis. The sharp rise in foreclosures in the sub-prime mortgage market, which began in the United States in 2006, has been blown into the global financial crisis of July 2007. Interest rates increased, newly popular adjustable rate mortgages and property values suffered declines from the demise of the housing bubble, leaving homeowners unable to meet financial commitments and lenders without a means to recoup their losses. Consequently, it is essential to provide a more thorough look into the future assessments of (mortgage) loan prices as well as interest rates when deliberating a long term obligation, such as a mortgage loan. That is because an adverse change in the market (as seen by the interest rate increases in the USA from approximately 1% at the beginning of 2003 to 5.25% in July 2007) can lead to customer defaults and human tragedies.

I appreciated the opportunity to perform meaningful research with very promising results in stochastic programming as well as demonstrating the practical use of stochastic programming and risk management in the contemporary finance industry.

Chapter 2

Introduction

This chapter aims to create an intuitive understanding of the role of a scenario generator as well as the structure of an optimization process that contains scenario generation. Later this chapter will cover the mathematical background and terminology used around scenario generation.
The first section discusses the need for a scenario generator in mathematical modeling. The following section discusses the role of a scenario generator as part of an optimization process. The next section introduces stochastic programming. This part is followed by a section introducing essential scenario generation terminology - scenario trees. This section is then followed by a short discussion on the complexity issues introduced by scenario tree generation. At its conclusion, the chapter ends with a short summary.

2.1 Why Should Someone Be Interested in Scenario Generation?

Some people believe that the only certain thing in life is death. Nevertheless, many decisions need to be taken by individuals or companies every day. Therefore, one can suppose that all our decisions hold a certain amount of uncertainty.

Operations Research is a field of applied mathematics that is used to help with decision making in complex real-world problems by modeling and solving them. In many cases the modeling process tries mathematically to capture the nature of the problem, i.e. the main processes, activities, dependencies, etc.

The problem specification usually describes the process (problem constraints), and then captures the success criteria or utility function (objective function). The model is then solved using a solver. However, the solution process is in many cases deterministic and if one agrees that uncertainty is assimilated in life, one would expect a good model to capture it.

Stochastic programming is used as a framework for modeling optimization problems that involve uncertainty. Stochastic programs need to be solved with discrete distributions. Usually, we are faced with either continuous distributions or data. Hence, we need to pass from the continuous distributions or the data to a discrete distribution suitable for calculations. The process of creating this discrete distribution is called scenario generation, and the result is a scenario tree.

More formally, stochastic programming is a branch of operations research that tries to suggest an approach to deal with uncertainty. Instead of suggesting an objective function such as $f(x)$ (in linear programming cx) in which the decision variable x is considered to have only one realisation as part of the objective function, the stochastic programming approach defines a stochastic variable $\xi \in \Omega$ and a new objective function $f(x, \xi)$. Therefore, the new objective function value is dependent on a different realization of ξ and therefore includes the effect of a stochastic process when evaluating the decision at the variable x. The purpose of a scenario generator is to discretize the distribution capture of all the various possible values of ξ and introduce uncertainty into the model. The output of the scenario generation is then used numerous times as the input for the optimization model.

It should be noted that as a general rule in operations research the value of your solutions is only as good as the data you put inside the model (a.k.a GIGO - Garbage In, Garbage Out). Having a proper way to capture uncertainty and generate scenarios are important milestones in the creation of a thorough stochastic programming solution.

2.2 Scenario Generator as Part of the Optimization Process

A scenario tree captures the uncertainty for a multi-stage stochastic programming problem and the process of building this input tree is called scenario generation.

A scenario generator receives as its input data what is believed to represent the distribution of an uncertain process that needs to be captured. The scenario generator creates scenarios that are possible future outcomes of the processes/distribution. These scenarios are later used by another optimization problem (a multi-stage stochastic programming problem). A graphic presentation of this process is found at Figure 2.1. There are, of course, several properties that need to be found by the scenarios to determine the quality of the scenario generation. These issues will be discussed further in later chapters of this report. It should also be noted that not only raw historical data is used as input for the scenario generator but input can also be an expert opinion or other parameters used to calibrate the scenario generation process.

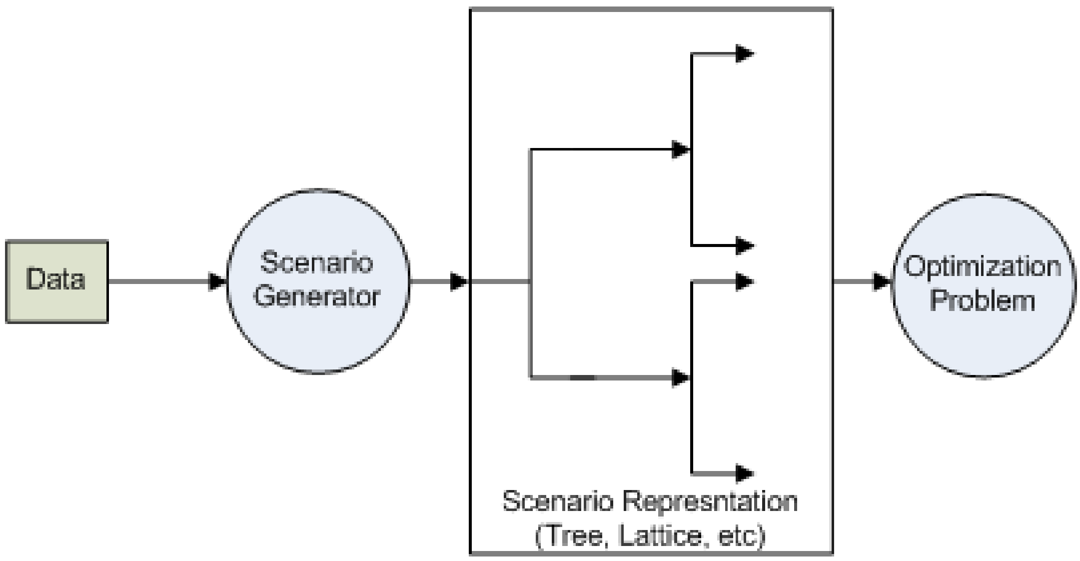


Figure 2.1: The Role of a Scenario Generator in Stochastic Programming Optimization Model

An example of a scenario generator can be a stochastic process that predicts the monthly electricity consumption of an apartment. This process input is the monthly historical time series of the electricity consumption in that apartment and its output is a guess for the electricity consumption the following month. Not only can a scenario generation be based on historical data as its input, but it can use a more complex function of its input. For example, consider a stock value that is analyzed and the yearly return of that stock is explored. The past returns are then formed into a function that has an expected value and standard deviation. The scenario generation can then receive as input the expectation and standard deviation of that function, and return as output different future scenarios. (For example the scenario generation can return three scenarios one is the expected return and the other two are the expected return plus/minus one standard deviation).

Remark 1: It should be mentioned that not all stochastic optimization applications use scenario generation to capture the underlying uncertainties in an optimization problem. The scenarios can also be assimilated as part of a general optimization problem. However, one reason to separate the scenario generation and the optimization is that it allows one to capture all the uncertainty of the optimization problem in one place only (the scenario generator) and in that way to better control the uncertainty by decoupling it from the optimization problem.

2.3 Stochastic Programming

As defined by the stochastic programming community - COSP at [25] - Stochastic programming is a framework for modeling optimization problems that involves uncertainty. Whereas deterministic optimization problems are formulated with known parameters. Real world problems almost invariably include some unknown parameters. When the parameters are known only within certain bounds, one approach to tackling such problems is called robust optimization. Here the goal is to find a solution which is feasible for all such data and optimal in some sense. Stochastic programming models are similar in style but take advantage of the fact that probability distributions governing the data are known or can be estimated. The goal here is to find some policy that is feasible for all (or almost all) the possible data instances. It maximizes the expectation of some function of the decisions and the random variables. More generally, such models are formulated, solved analytically or numerically, and analyzed in order to provide useful information for a decision maker.

The most widely applied and studied stochastic programming models are two-stage linear programs. Here the decision maker takes some action in the first stage, after which a random event occurs affecting the outcome of the first-stage decision. A recourse decision can then be made in the second stage that compensates for any bad effects that might have been experienced as a result of the first-stage decision. The optimal policy from such a model is a single first-stage policy and a collection of recourse decisions (a decision rule) defining which second-stage action should be taken in response to each random outcome. These results can later be extended into multi-stage stochastic programming.

More formally I have used the definitions as described by J.R. Birge and F. Louveaux in [2]. A deterministic linear program is defined as:

Minimize

$$z = c^T$$

Subject To:

$$Ax = b$$

$$x \geq 0$$

where x is an $(n \times 1)$ vector of decisions and c, A and b are known data of the sizes $(n \times 1)$, $(m \times n)$ and $(m \times 1)$. In this formulation all the first-stage decisions are captured by the variable x.

Let us look now at a two-stage problem with fixed recourse by G.B. Dantzig at [3] and Beale at

[1]:

Minimize

$$z = c^T x + E_\xi[minq(\omega)^T y(\omega)]$$

Subject To:

$$Ax = b$$

$$T(\omega)x + Wy(\omega) = h(\omega)$$

$$x \geq 0, y(\xi) \geq 0 \tag{2.1}$$

The first-stage decisions are represented by a familiar vector x which is an $(n \times 1)$ vector of decisions and c,A and b are known data of the sizes $(n \times 1)$, $(m \times n)$ and $(m \times 1)$. However, this model considers a representation of a number of random events $\omega \in \Omega$. For a given realization ω the second-stage problem data $q(\omega), h(\omega)$ and $T(\omega)$ become known, where $q(\omega)$ is $n_2 \times 1$, $h(\omega)$ is $m_2 \times 1$ and $T(\omega)$ is $m_2 \times n_1$. Each component of q, h and T is thus a possible random variable. Piecing together all the stochastic components of the second-stage data and the vector $\xi^T(\omega) = (q(\omega)^T, h(\omega)^T, T_1(\omega), \ldots, T_{m_2}(\omega))$ is obtained. The optimization model now considers future scenarios that are dependent upon different values of ξ in order to make the first-stage decision x.

According to Kaut and Wallace in [8] stochastic programming has gained increasing popularity within the mathematical programming community. Present computing power allows users to add stochasticity to models that had been as difficult to solve as deterministic models only a few years ago. In this context, a stochastic programming model can be viewed as a mathematical programming model with uncertainty about the values of some of the parameters. Instead of single values, these parameters are then described by distributions (in a single-period case), or by stochastic processes (in a multi-period case), where ξ is a random vector, whose distribution must be independent of the decision vector x. Note that the formulation is far from complete we still need to specify the meanings of min and the constraints.

It is interesting to note that the special structure of the stochastic programming problems as different blocks of constraints are considered in different scenarios. These can be very useful for solving problems. When such a problem is created different solving heuristics that exploit this structure can perform better and faster than others. This is done by the several decomposition algorithms including the L-Shaped method.

Except for some trivial cases, the problem (2.1) can not be solved with continuous distributions. Most solution methods need discrete distributions. In addition, the cardinality of the support of discrete distributions is limited by the available computing power, together with a complexity of the decision.

In the following report the scenario generator is used in order to find different likely values for $\omega \in \Omega$. These values can later be solved in an optimization model and be used as scenarios.

In that sense, the scenario generation process discretizes the stochasticity of the problem.

As described by Hochreiter at [26]. The field of multi-stage stochastic programming provides a rich modeling framework for tackling a broad range of real-world decision problems. In order to numerically solve such programs - once they get reasonably large - the infinite-dimensional optimization problem has to be discretized. The stochastic optimization program generally consists of an optimization model and a stochastic model. In the multi-stage case, the stochastic model is most commonly represented as a multi-variate stochastic process. There are different ways to represent scenarios and a few of them will be considered in the following section. The most common technique to calculate a usable discretization is to generate a scenario tree from the underlying stochastic process.

2.4 Scenario Trees

By far most used way to represent scenarios is scenario trees. Each level in the tree represents a different time point and all the nodes for a particular time point represent the possible scenarios for that time point. An example can be seen in figure 2.2.

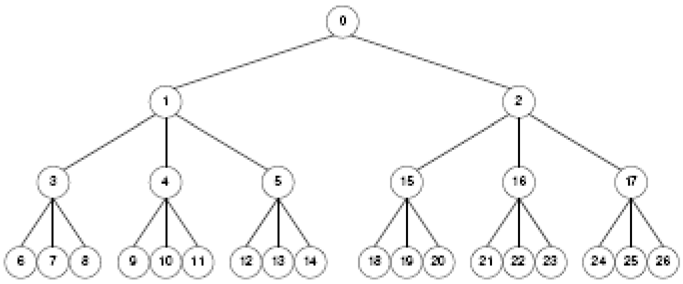


Figure 2.2: Example of a Scenario Tree

As can be seen in the figure, the number of child nodes at each level does not need to match the number of child nodes in another level. For example, node 0 in the picture has two child nodes while nodes 1 and 2 have three. The different levels do not necessary represent the same time gaps. In the example, level 0 can represent year 0, level 1 can represent the year 2 and level 2 can represents the year 10. In fact, in some complex scenario trees, as can be seen in figure 2.3, there are not even the same number of child nodes on the same level.

More formally, scenario tree formulation is found in the next subsection.

2.4.1 Scenario Tree Formulation

There are a number of mathematical representations for a scenario tree. A more formal mathematical formulation of a scenario tree based on Hochreiter at [26] is described in this subsection. First assume that a discrete-time continuous space stochastic process $(\xi_t)_{[t=0,1...T]}$ is given, where $\xi_0 = x_0$ represents today's value and is constant. The distribution of this process may be the result

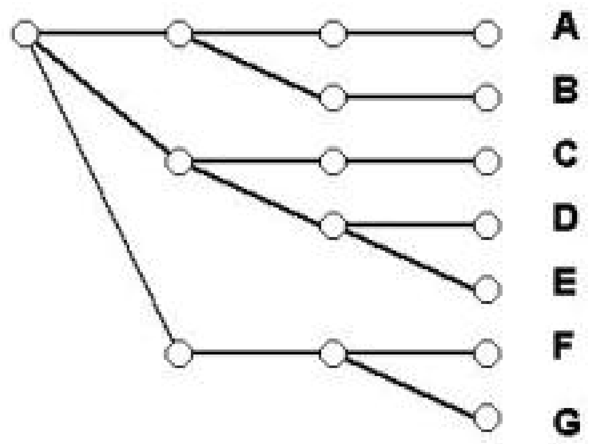


Figure 2.3: Example of a Complex Scenario Tree Structure

of a parametric or non-parametric estimation based on historical data. The state space may be univariate (the $\mathbb{R}^1$) or multivariate (the $\mathbb{R}^k$). We look for an approximate simple stochastic process $\tilde{\xi}_t$, which takes only finitely values and which is as close as possible to the original process (ξ_t) and at the same time has a predetermined structure as a tree. Denote the finite state space of $\tilde{\xi}_t$ by S_t, i.e.

$$\mathbb{P}\{\tilde{\xi}_t \in S_t\} = 1$$

Let $c(t) = \#(S_t)$ be the cardinality of S_t. We have that $c(0) = 1$. If $x \in S_t$, we call the branching factor of x the quantity

$$b(x, t) = \#\{y : \mathbb{P}\{\tilde{\xi}_{t+1} = y | \tilde{\xi}_t = x\} > 0\}$$

Obviously, the process $(\tilde{\xi}_t)_{t=0,\dots,T}$ may be represented as a tree, where the root is $(x_0, 0)$ and the node (x, t) and $(y, t+1)$ are connected by an arc, if $\mathbb{P}\{\tilde{\xi}_t = x, \tilde{\xi}_{t+1} = y\} > 0$. The collection of all branching factors $b(x, t)$ determines the size of the tree. Typically, we choose the branching factors beforehand and independent of x. In this case, the structure of the tree is determined by the vector $[b(1), b(2), b(3), \dots, b(T)]$. For example, a [5,3,3,2] tree has height 4 and $1 + 5 + 5 \cdot$

$3 + 5 \cdot 3 \cdot 3 + 5 \cdot 3 \cdot 3 \cdot 2 = 156$ nodes. The number of arcs is always equal the number of nodes minus 1. The main approximation problem is an optimization problem of one of the following types and is most often determined by the chosen scenario generation method:

The given-structure problem. Which discrete process $(\tilde{\xi}_t), t = 0, \ldots, T$ with given branching structure $[b(1), b(2), \ldots, b(T)]$ is closest to a given process $(\xi_t), t = 0, \ldots, T$? The notion of closeness has to be defined in an appropriate manner.

The free-structure problem. Here again the process $(\xi_t), t = 0, \ldots, T$ has to be approximated by $(\tilde{\xi}_t), t = 0, \ldots, T$, but its branching structure is free except for the fact that the total number of nodes is predetermined. This hybrid combinatorial optimization problem is more complex than the given- structure problem.

A summary of these methods developed before 2000 can be found in [27]. Methods published since include [4], [5] for moment matching strategies, [19], [28],[29] for probability metric minimization and [30], [31] for an integration quadratures approach.

2.4.2 Pro Et Contra - Arguments For and Against

- Arguments For

  + The use of scenario trees decouple the uncertainty from the optimization problem. The uncertainty is kept in the scenario tree which makes it possible to examine different approaches for scenario generation without changing the optimization problem. It also makes it possible to extract a successful scenario generation approach to be used on different optimization problems.

  + Scenario trees are very intuitive structures for stochastic programming problems.

  + Scenario trees keeps the path for the scenario. The tree structure allows you to connect different scenarios at different time points.

+ The use of the tree structure can allow an algorithm to examine only part of the tree so it can be used by recursive algorithms.

- Arguments Against

  - The biggest difficulty when using scenario trees is the exponential growth in the number of scenarios. If three scenarios are generated for every node at any level and there are 21 levels the number of scenarios generated will be $\sum_{i=0}^{20} 3^i$ about 5 Billion scenarios.

2.4.3 Other Scenario Tree Representations

Another common tree structure that can be used for scenario generation is a lattice tree. As can be seen in figure 2.4, a binomial lattice tree keeps the properties that different tree levels represent at different time periods and any specific node can be seen as a scenario. However, different paths

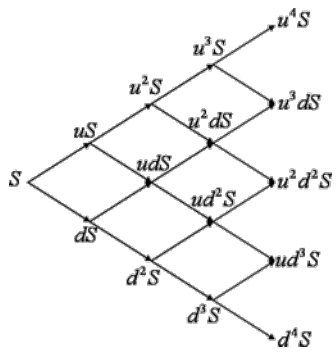


Figure 2.4: A Binomial Lattice Tree

can be used to receive the same scenario. When looking at the example in figure 2.4, u and d represent up and down scenarios respectively. The path u and d will find the same node as the path d and u afterwards. On the other hand, this approach does not lead to exponential growth in the number of scenarios.

2.5 Difficulties Related to Scenario Generations

There are at least two major issues in the scenario generation process:

- The number of scenarios must be small enough for the stochastic program to solve.
- The number of scenarios must be large enough to represent the underlying distribution or data in a good way.

For most reasonable cases, pure sampling will not be good enough. Certainly, with enough sample points, the second item above will be well taken care of, but most likely the first will not. If the sampling is stopped so that the corresponding stochastic program can be solved in reasonable time, its statistical properties are most likely not very good, and the problem we solve may not represent the real problem very well.
The main limitation for this problem is the vast number of scenarios. If we use k scenarios per time period and generate a scenario tree we will receive an exponential number of scenarios in k. The number of scenarios received for a $t \geq 0$ period scenario tree is the sum of scenarios for each time period between $i = 0, \ldots, t$ there are k^i scenarios and in total $\sum_{i=0}^{t} k^i$. When keeping in mind that a thorough scenario representation is dealing with at least 3-4 scenarios for each time frame it leads for very small periods of scenario representations. The exponential number means that having scenarios for more than 3-20 periods will be computationally impossible, the exact number is also dependent in the size of k. For example, the number of scenarios for k=6 and time frame of 8 periods is more than 2,000,000 scenarios which is a huge input for any problem. This is also the main limitation regarding the problem of scenario calculation.
When dealing with computing the problem of finding scenarios, we deal with a non-linear optimization problem as well. That makes the problem hard to solve and a non linear optimization problem with more than 2,000,000 variables is something that cannot really be solved by the tools available to us nowadays.

This limit will especially have an effect when dealing with the tests of models and their usage. The number of periods available is very low and for practical purposes it means that the models used here will only be able to make decisions in the near future.
For financial problems this is often not enough. An investment, such as buying a house or taking a mortgage loan, deal with a period of 20-30 years. While the decision regarding a loan can be done every month, in a model we will use periods of 4-5 years with decisions made every year. Then later the model will be able to run again at the end of this period and make some other decisions. However usually a person making a decision regarding real estate can only make a proper decision for a period of 4-5 years. Since so many microeconomics, macroeconomics, and other data can completely change the financial environment, for short term decisions these models can still be appropriate.

2.6 Summary

This chapter introduces the concept of scenario generation as well as the appropriate terminology and methods used in optimization problems that are based on stochastic programming. Scenario trees are then introduced and the complexity problems that are presented when scenario generation applications are discussed. This introduction chapter build the foundation for further scenario generation applications that are built in the following chapters.

Chapter 3

Review Of Scenario Generation Methods

This chapter begins by examining measures for scenario quality, followed by a wide overview of the most used scenario generation methods. The approaches are heavily based on Zenios at [14] and Kaut and Wallace at [8].

3.1 Introduction

This chapter gives an overall overview of different approaches of scenario generation. The common belief in the academic world is that there is no one general scenario generation approach that can be applicable for all stochastic programming problems. A good scenario generator is usually very problem specific. Moreover, the lack of a standard for scenario generation makes it very difficult to compare different techniques.

This chapter approaches these issues by identifying good scenario generation properties and give an overview of different scenario generation techniques. This chapter starts by suggesting scenario qualities that should be examined. While this report will mainly deal with moment matching scenario generation approaches, this chapter will go through the definitions of other approaches with a few examples.

3.2 Quality of Scenarios

Zenios at [14] defined three main criteria for identifying the quality of scenario generation - Correctness, Accuracy and Consistency. These criteria are explained below:

Correctness -

- Scenarios should contain properties that are prevalent from the academic research point of view. For example, the term structure should exhibit mean reversion and changes. The term structure consists of changes in level, slope and curvature as examined in academic research.

- Scenarios should also cover all relevant past history. Furthermore, scenarios should account for events that were not observed, but are plausible under current market conditions.

Accuracy -

- As in many cases, scenarios represent a discretization of a continuous process. Accumulating a number of errors in the discretization is unavoidable. Different approaches can be used to ensure the sampled scenarios still represent the underlying continuous distribution function.

- Accuracy is ensured when, for instance, the first and higher moments of the scenarios match those of the underlying theoretical distribution. (Moments and property matching are often used in order to ensure that the scenarios keep the theoretical moments of the distribution they represent.)

- The accuracy demand can lead to a large number of scenarios generated. That is in order to create a fine discretization of the continuous distribution and to achieve the accuracy considered appropriate and acceptable for the application at hand.

Consistency -

- When scenarios are generated for several instruments (e.g. bonds, term structure, etc.), it is important to see that the scenarios are internally consistent.

- For example scenarios in which an increase in the interest rate together with an increase in bond prices are inconsistent. Even though in a stand-alone scenario the same increase in interest rates or an increase in bond prices are both consistent scenarios.

- Taking into consideration the correlation between different financial instruments can be used to ensure scenarios' consistency.

In order to examine these fundamentals, I tend to think about using a clock to keep track of time. Accuracy is guaranteed when the clock's battery is fully charged and the time is displayed correctly. Consistency is achieved if the clock shows the correct time day after day. Correctness is confirmed when a news broadcast on the hour is shown on the clock as that precise hour, assuming that the radio/television station's clock is calibrated for accuracy. (Note: Many radio and television stations use an official government clock that is adjusted for accuracy according to an atomic clock.)

3.3 Overview of Scenario Generations Methodologies

Alternative methodologies for scenario generations will be discussed in this chapter all fit into one of the three categories as can be seen in figure 3.1. Bootstrapping is obviously the simplest approach to be used and it is only performed by sampling of the already observed data. A second approach models historical data using statistical analysis. A probability distribution is fitted to the data and sample scenarios are then drawn from that distribution. A third approach develops continuous time theoretical models with parameters estimated to fit the historical data. These models are then discretized and simulated to generate scenarios. These approaches can be seen in Figure 3.1

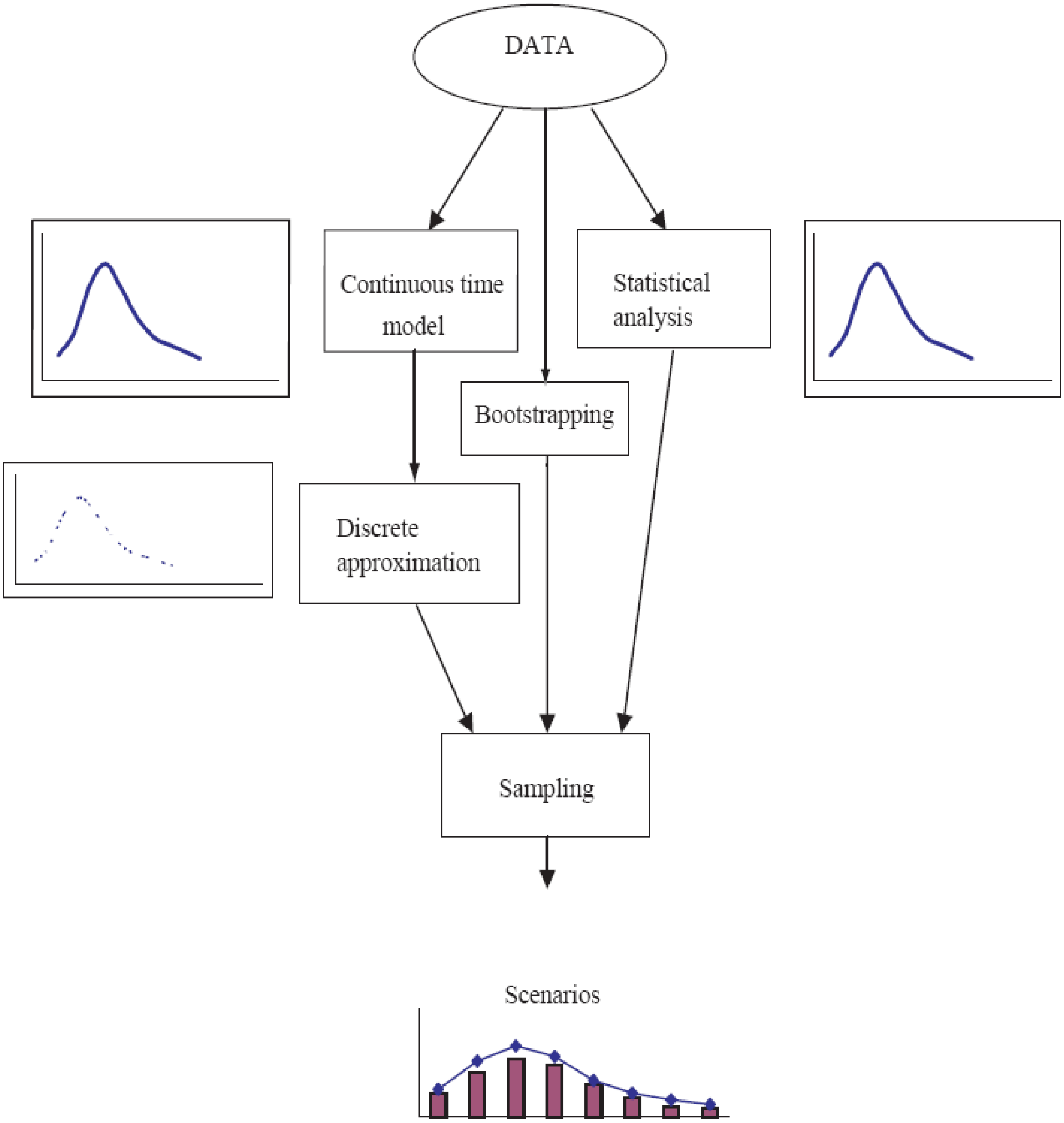


Figure 3.1: Scenario Generation Methodologies: Bootstrapping, Statistical Analysis of Data and Discrete Approximation of Continuous Time Models (taken from Zenios at [14])

The rest of this section looks into examples of these methodologies while examining the criteria of the quality of the scenarios as shown in the previous section.

3.3.1 Conditional Sampling

These are the most common methods for generating scenarios. At every node of a scenario tree, we sample several values from the stochastic process $\{\xi_t\}$. This is done either by sampling directly from the distribution of $\{\xi_t\}$, or by evolving the process according to an explicit formula:

$$\xi_{t+1} = z(\xi_t, \epsilon_t)$$

Traditional sampling methods can sample only from a univariate random variable. When we want to sample a random vector, we need to sample every marginal (the univariate component) separately, and combine them afterwards. Usually, the samples are combined all-against-all, resulting in a vector of independent random variables. The obvious problem is that the size of the tree grows exponentially with the dimension of the random vector: if we sample s scenarios for k marginals, we end-up with s^k scenarios.

Another problem is how to get correlated random vectors (a common approach can be seen at [32] [33] [34]) to find the principal components (which are independent by definition) and sample those, instead of the original random variables. This approach has the additional advantage of reducing the dimension, and therefore reducing the number of scenarios.

There are several ways to improve a sampling algorithm. Instead of a pure sampling, we may, for example, use integration quadratures or low discrepancy sequences (see [35]). For symmetric distributions [36] uses an antithetic sampling. Another way to improve a sampling method is to re-scale the obtained tree to guarantee the correct mean and variance (see [37]).

When considering the quality of a sampling method, the strongest candidate for the source of the problem is a lack of scenarios, as we know that, with an increasing number of scenarios, the discrete distribution converges to the true distribution. Hence, by increasing the number of scenarios, the trees will be closer to the true distribution and consequently also closer to each other. As a result, both the instability and the optimality gap should decrease. That will ensure the accuracy condition.

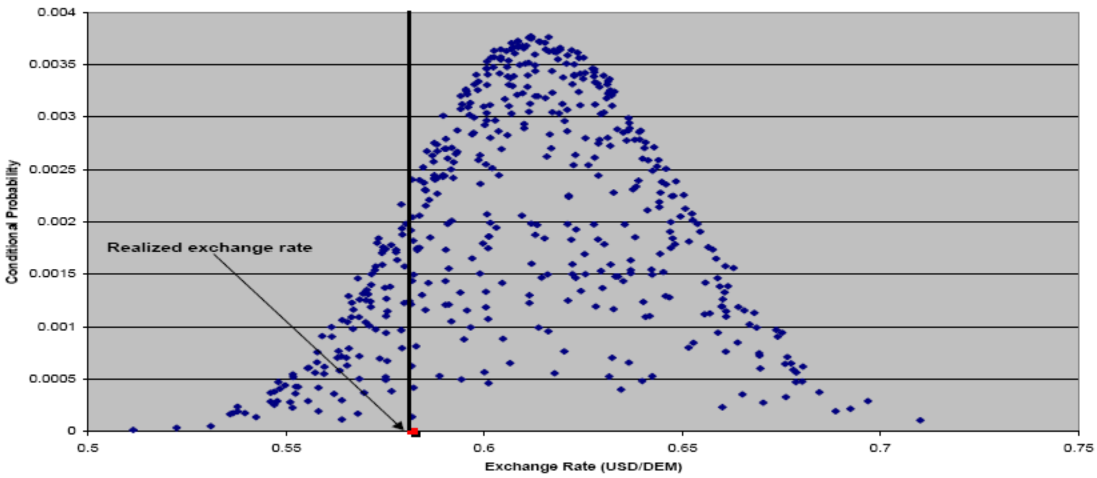
As an example of the use of this method consider to generate exchange rate scenarios, conditioned on scenarios of interest rates. These joint scenarios of interest rate and exchange rates are used in the management of international bond portfolios. Figure 3.2 illustrates the conditional probabilities for several exchange rate scenarios. On the same figure the exchange rate that was realized ex-post on the date for which the scenarios were estimated is plotted. Note that the same exchange rate value may be obtained for various scenarios of interest rates and samples. The figure plots several points with the same exchange rate value but different conditional probabilities.

3.3.2 Bootstrapping Historical Data

The simplest approach for generating scenarios using only the available data without any mathematical modeling is to bootstrap a set of historical data. In that context each scenario is a sample of returns of the assets obtained by sampling returns observed in the past. In order to generate scenarios of returns, for 10 years, a sample of 120 monthly returns from 10 years is used. This process can be repeated to generate several scenarios for return over 10 years. This approach preserves the observed correlation. However, this approach will not satisfy the correctness demand of scenario generation since it will never suggest a monthly return in a scenario that was never observed. When sampled correctly the scenarios satisfy accuracy and consistency as these scenarios satisfy real life observations.

3.3.3 Moment Matching Methods

In many situations when the marginal distribution for the scenario generation process is not known a moment matching approach is preferable. A moment matching scenario generation process would usually explore the first three or four moments (mean, variance, skewness, kurtosis) of the scenario generation process as well as the correlation matrix. These methods can be ex-



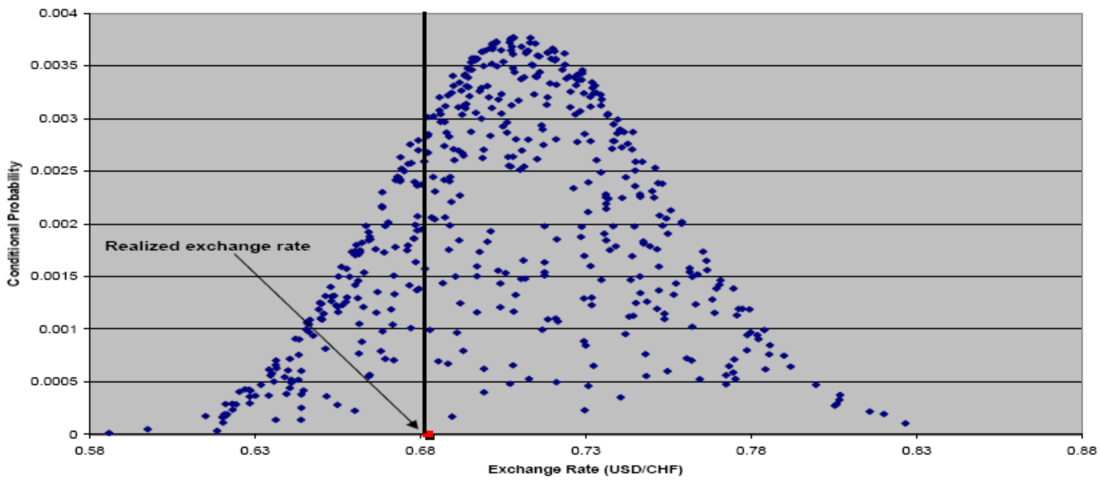


Figure 3.2: Exchange Rate Scenarios and Their Conditional Probabilities for the DEM and CHF Against the USD (taken from Zenios at [14])

tended to other statistical properties. (such as percentiles, higher co-moments, etc.) The moment matching scenario generator will then construct a discrete distribution satisfying the selected statistical properties.

These approaches have a wide impact on the industry, as they are very intuitive and easily implemented (see Johan Lyhagen at [49]).

A moment matching approach ensures accuracy by definition as it matches statistical moments. Matching the covariance matrix ensures scenario consistency. However, correctness is not ensured since the approach is general and does not reflect the academic knowledge which is problem specific.

3.3.4 Statistical Analysis: Time Series Modeling for Econometric Models

Time series models relate the value of variables at given points in time to the value of these variables at previous time periods. Time series analysis is particularly suitable for solving aggregated asset allocation problems when the correlation among asset classes is very important. When time series analysis is extended to model the correlations with some macroeconomics variables, such as short rates or yield curves, the resulting simulation model can be used to describe the evaluation of the corresponding problem (for example an Asset Liability Management (ALM), pricing or interest rates problem). Vector autoregressive (VAR as opposed to VaR - Value at Risk) models are used extensively in econometric modeling. A VAR model for scenario generation will later be described as part of this thesis.

3.3.5 Optimal Discretization

Pflug at [19] describes a method which tries to find an approximation of a stochastic process (i.e. scenario tree) that minimizes an error in the objective function of the optimization model. Unlike the methods from the previous sections, the whole multi-period scenario tree is constructed at once. On the other hand, it works only for univariate processes. For multistage problems, a scenario tree can be constructed as a nested facility location problem (as was shown by Hochreiter and Pflug at [47]). Multivariate trees may be constructed by a tree coupling procedure.

3.4 Summary

Successful applications of financial optimization models hinge upon the availability of scenarios that are correct, accurate and consistent. Obtaining such scenarios is a challenging task. There are none available. This chapter introduced a number of measures for scenario qualities as well as an overview of used scenario generation approaches. Other approaches for scenario generation include Markov Chain Monte Carlo (MCMC), Hidden Monte Carlo and Vector Error Correlation Methods (VECM), for solving differential equations and using discrete lattice approximations in continuous models. A short overview of the most common scenario generation approaches can also be seen at [8].

The conclusion of this chapter is not that moment-matching is a good scenario generation method for every stochastic program. There is no dominant strategy for scenario generation, however, the moment matching approach does ensure accuracy and consistency. In the rest of this report an interest rate scenario generator based on moment matching is suggested, described and tested. Since a well defined scenario generation should satisfy the correction criteria, the moment matching scenario generation process is extended to capture correctness criteria for interest rate modeling.

As a first step, a better understanding of moment-matching scenario generation will be further described and examined in the next chapter.

I would like to state the promising research on moment-matching scenario generation done by Højland and Wallace at [4], Højland, Kaut and Wallace at [5], as well as the research on optimal discretization by Pflug at [19] followed by Pflug and Hochreiter at [47] do provide appropriate answers for both accuracy and consistency of scenarios. However, in order to deal well with scenario correctness more research should be performed to identify academic properties on the specific domain of different scenario generation classes (e.g. bond pricing, house pricing, interest rates, etc.). This report will examine some of the academic properties described in the research about interest rates and fit it into the scenario generation process.

Chapter 4

Moment Matching

While the previous chapter examined different scenario generation approaches, this chapter emphasizes moment matching approaches. The first section examines different statistical properties. The second section describes the algorithm by Højland and Wallace at ([4]) as an operations research problem. The algorithm is discussed in detail. The third section looks into a heuristic for moment-matching scenario generation based on a paper by Højland, Kaut and Wallace ([6]), followed by a summary.

4.1 Statistical Properties

In this section we will be looking deeply into statistical analysis of scenario generation models. These properties are later matched in order to find future scenarios.

The scenario generation methods that will be considered are based on different statistical properties that capture the behaviour of the stochastic process for which the scenarios are generated.

In this section the most common statistical properties are considered. It is important to note that other statistical properties or general properties can also be considered with moment matching scenario generation approaches. This flexibility is one of the main reasons why many real life scenario generation applications (see for example [52] and [48]) are based upon moment matching.

4.1.1 Matching Statistical Moments

The most common statistical properties to be considered are the moments of the stochastic processes. There are several approaches to calculate the moments either based on a sample or based on a mathematical definition. For each of the models both approaches are considered. Since all of our calculations are discrete times, only the discrete variable definitions are mentioned.

When a capitalized variable is used (as X) the corresponding variable refers for a vector or a matrix. While non capitalized letters (as x_i) refers for discrete single values.
The simple notation used in our definitions is shown below:

- X,Y are considered as random variables.
- The notation x_i or y_i is considered as the i^{th} possible value of the random variables X and Y $(i = 1, 2, ...)$ accordingly.
- The portabilities for each of random variable values x_i is p_i i=1,2,..
- The portabilities for the intersection of the random variable values x_i and y_j is determined as p_{ij} i=1,2,.. j=1,2,..

4.1.2 Expectation

The expectation is the first central moment. It simply represents the weighted sum of the random variable values, i.e. the arithmetical mean. When a sample is considered the random variable values are the sampled values.
Mathematical definition:

$$E(X) = \sum_i p_i X_i$$

The sample definition of expectation is as follows:

$$E(X) = \overline{x} = \frac{\sum_{i=1}^{N} X_i}{N}$$

The definitions are simple and therefore examples are exempted. The notations $\overline{x}$ will be used later on in the paper referring to this definition.

4.1.3 Standard Deviation

The standard deviation is the root mean square (RMS) deviation of the values from their expectations.

For example, in the population {4, 8}, the mean is 6 and the deviations from mean are {-2, 2}. Those deviations squared are {4, 4} the average of which (the variance) is 4. Therefore, the standard deviation is 2. In this case 100% of the values in the population are at one standard deviation of the mean.

The standard deviation is the most common measure of statistical dispersion, measuring how widely spread the values in a data set are. If the data points are close to the mean, then the standard deviation is small. Also, if many data points are far from the mean, then the standard deviation is large. If all the data values are equal, then the standard deviation is zero.

The mathematical definition is therefore:

$$\sigma(X) = \sqrt{E(X^2) - E(X)^2}$$

When the expected statistical definition is:

$$\sigma(X) = \sqrt{\frac{\sum_{i=1}^{N}(X_i - \overline{x})^2}{N}}$$

However, this definition is usually not the one used when a sample standard deviation is used because it leads for a bias estimator of the standard deviation. In statistics the difference between an estimator's expected value and the true value of the parameter being estimated is called the bias. An estimator or decision rule having a nonzero bias is said to be biased.

Lets consider the first definition suggested for the sample standard deviation and calculate its expectation. When the previous definition is used it can be shown that:

$$E(S^2) = \frac{n-1}{n}\sigma^2 \neq \sigma^2$$

That, in turn, leads to a biased definition of the variance.

In order to avoid this problem, the unbiased estimator of the sample standard deviation is defined to be:

$$s = \sqrt{\frac{\sum_{i=1}^{N}(X_i - \overline{x})^2}{N-1}}$$

In a similar manner, the definition of the estimators for the third and forth moments (i.e. skewness and kurtosis) are also changed to be kept unbiased. Later, when these moments are discussed only the unbiased definitions will be shown and this discussion will not be repeated.

In practice one often assumes that the data is measured from a normally distributed population. Figure 4.1 shows the different dispersions for normal distribution. The standard deviation in this case is widely used for the calculation of confident interval measures the probability of one specific sample of the population being in a specific range of values. That can also be seen in figure 4.1. It should be noted that if it is not known whether the distribution is normal, Chebyshev's inequality can always be used for the creation of a confident interval. For example, at least 50 %

of all values are within 1.4 standard deviation from the mean.

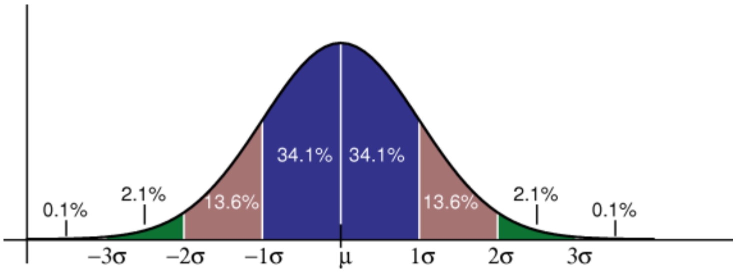


Figure 4.1: Standard Deviation Spread Over a Normal Distribution

4.1.4 Skewness

Skewness is the measure of the asymmetry of the probability distribution. Roughly speaking, a positive skewness represents a long or fatter right tail in Comparison to the left tail, while a negative skewness represents the opposite situation. Therefore a symmetrical distribution (for example the normal distribution in Figure 4.1) has a skewness of zero. An example of nonzero skewness can be seen in figure 4.2.

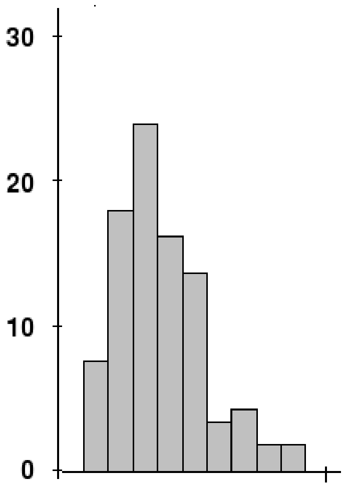


Figure 4.2: Nonzero Skewness

The skewness is the standardized third moment over the mean. When μ_3 is the third moment over the mean and σ is the standard deviation, the skewness (Sometimes referred as skew or skew(X)) is defined as:

$$skew = \frac{\mu_3}{\sigma^3}$$

The theoretical skewness is defined as:

$$skew(X) = \frac{E[(X - E(X))^3]}{\sigma^3}$$

When the definition of s is the unbiased estimator for the standard deviation, the unbiased estimator for the skewness is then:

$$skew = \frac{N}{(N-1)(N-2)} \Sigma_{i=1}^{N} (\frac{x_i - \overline{x}}{s})^3$$

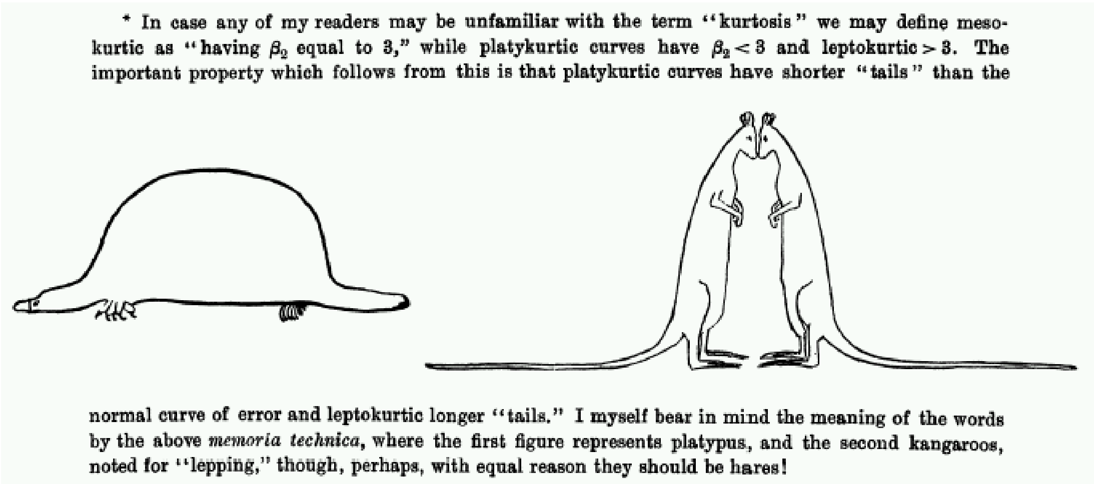
4.1.5 Kurtosis

The kurtosis (symbolized as kurt or kurt(x)) is the forth standardized central moment.

$$kurt = \frac{\mu_4}{\sigma^4}$$

The kurtosis is a measure of the peakedness of the probability distribution. The kurtosis of the normal distribution is 3. Therefore, in many cases the kurtosis is defined as Kurt(x) - 3, in order to easily compare the peakedness to the one of the normal distribution. A high kurtosis occurs when a high percentage of the variance is due to infrequent extreme deviations from the mean. On the other hand, a low kurtosis occurs if the variance is mostly due to frequent modestly-sized deviations for the mean.

In his "Errors of Routine Analysis" Biometrika, 19, (1927), p. 160, a student provided a mnemonic device that is shown in figure 4.4. In the figure it can be seen that the platypus on the left hand size represents a frequent modestly-sized variation in the distribution and therefore a low kurtosis, while the two kangaroos on the right hand side represent extreme deviations with a long tail and therefore a high kurtosis.



* In case any of my readers may be unfamiliar with the term "kurtosis" we may define mesokurtic as "having β_2 equal to 3," while platykurtic curves have $\beta_2 < 3$ and leptokurtic > 3. The important property which follows from this is that platykurtic curves have shorter "tails" than the normal curve of error and leptokurtic longer "tails." I myself bear in mind the meaning of the words by the above *memoria technica*, where the first figure represents platypus, and the second kangaroos, noted for "lepping," though, perhaps, with equal reason they should be hares!

Figure 4.3: Student's Kurtosis Explanation

The theoretical kurtosis is defined as:

$$kurt(X) = \frac{E[(X - E(X)^4]}{\sigma^4}$$

With the definition of s as the unbiased estimator for the standard deviation, the unbiased estimator for the kurtosis - 3 is then:

$$kurt(X) - 3 = (\frac{N(N+1)}{(N-1)(N-2)(N-3)}\Sigma_{i=1}^{N}(\frac{x_i - \overline{x}}{s}^{4})) - 3\frac{(N-1)^2}{(N-2)(N-3)}$$

4.1.6 Correlation Matrix

In probability theory and statistics, correlation – also called correlation coefficient – indicates the strength and direction of a linear relationship between two random variables. In general statistical usage, correlation refers to the departure of two variables from independence, although correlation does not imply causation. In this broad sense there are several coefficients, measuring the degree of correlation, adapted to the nature of data.

A number of different coefficients are used for different situations. The best known is the Pearson product-moment correlation coefficient, which is obtained by dividing the covariance of the two variables by the product of their standard deviations. Despite its name, it was first introduced by Francis Galton.

The correlation coefficient $\rho_{X,Y}$ between two random variables X and Y with expected values μ_X and μ_Y and standard deviations σ_X and σ_Y is defined as:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E((X-\mu_X)(Y-\mu_Y))}{\sigma_X \sigma_Y}$$

where E is the expected value operator and cov means covariance. Since $\mu_X = E(X), V(X) = \sigma_X^2 = E(X^2) - E(X)^2$ and likewise for Y, we may also write

$$\rho_{X,Y} = \frac{E(XY) - E(X)E(Y)}{\sqrt{E(X^2) - E(X)^2}\sqrt{E(Y^2) - E(Y)^2}}$$

The correlation is defined only if both standard deviations are finite and both of them are nonzero. It is a corollary of the Cauchy-Schwarz inequality that the correlation cannot exceed 1 in absolute value.

The correlation is 1 in the case of an increasing linear relationship, -1 in the case of a decreasing linear relationship, and some value in between in all other cases, indicating the degree of linear dependence between the variables. The closer the coefficient is to either -1 or 1, the stronger the correlation between the variables.

If the variables are independent then the correlation is 0, but the converse is not true because the correlation coefficient detects only linear dependencies between two variables. Here is an example: suppose the random variable X is uniformly distributed on the interval from -1 to 1, and $Y = X^2$. Then Y is completely determined by X, so that X and Y are dependent, but their correlation is zero (From symmetry property $\forall n\ E(X^n) = 0$ in the chosen interval); they are uncorrelated. However, in the special case when X and Y are jointly normal, being independent is equivalent to being uncorrelated.

A correlation between two variables is diluted in the presence of the measurement error around estimates of one or both variables, in which case disattenuation provides a more accurate coefficient .

4.2 Generating Scenario Trees for Multistage Problems

The paper [4] by Højland & Wallace in 2001 develops a scenario generation technique for multivariate scenario trees, based on optimization. The following subsections will present in more detail the mathematical approach used in this model.

This section will describe the one period approach since this is the version used as part of the construction later described in chapter 6.

4.2.1 Motivation

If random variables are represented by multidimensional continuous distributions, or by discrete distributions with a large number of outcomes, computation is difficult since the models explicitly or implicitly require integration over such variables. To avoid this problem, we normally resort to internal sampling or procedures that replace the distribution with a small set of discrete outcomes in real life applications.

Internal sampling is used in many models of stochastic decomposition (see for example, Higle and Sen from 1991 at [53] and importance sampling by Infager 1994 at [54]).

The standard approach for approximating a continuous distribution is the following:

- Divide the outcome region into intervals

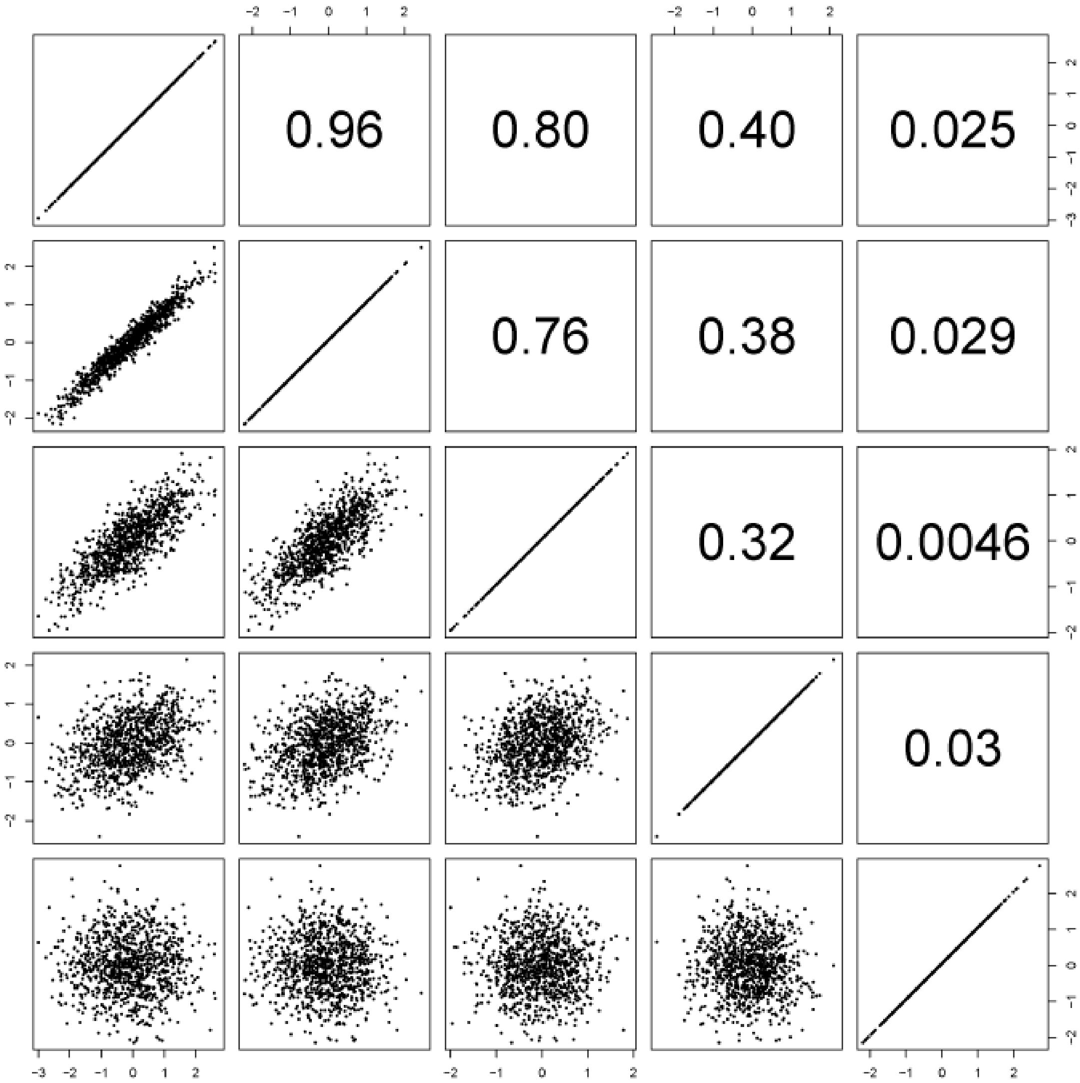


Figure 4.4: Simple Example of Linear Correlation. 1000 Pairs of Normally Distributed Numbers are Plotted Against One Another in Each Panel (bottom left), and the Corresponding Correlation Coefficient Shown (top right). Along the Diagonal, Each Set of Numbers is Plotted Against Itself, Defining a Line with Correlation +1. Five Sets of Numbers were Used, Resulting in 15 Pairwise Plots.

- Select a representing point for each interval
- Assign a probability to each point

An example of this kind of approach is the "bracket mean" method. In that method, intervals are found by dividing the outcome region into N equally probable intervals. The representative point in each interval is the mean point of the corresponding interval with the assigned probability of 1/N. However, as pointed out by Miller and Rice (1983) at [55], "bracket mean" methods always underestimate the even moments and usually underestimate the odd moments of the original distribution. That of course raise questions in regards to the accuracy of this approach.

The moment-matching approach presented here illustrates a different approach to scenario generation. Rather than discretization of a continuous process or sampling approach, this approach suggests exploring the statistical outcome of the process and "reverse engineering" it into a new stochastic process that comply with these properties.

This approach is very flexible with regards to user specification. Users can specify the structure of the outcomes to be constructed and which distribution properties are relevant for a specific problem.

The rest of this section will present the mathematical model as well as evaluation of this method.

Description of model data

This method produces a scenario tree. The nodes in the scenario tree depict states of the world at a particular point in time. The model presented will be looking at a one stage model. In stochastic programming decisions are made at the child nodes. The arcs of the scenario tree represent realizations of the uncertain variables. The scenario tree branches off for each possible value of a random vector $x = (x_1, \ldots, x_l)$, in each time point.

The data is:

- Sets (indices)
  - Scenarios: $n \in 1, \ldots, N$
  - Statistical properties: $S_l \in S_1, \ldots, S_L$
- Data:

  - $S_{l,VAL}$ The matched value of the statistical property S_l
  - p_n probability of scenario n (of course $\sum_n p_n = 1$)
  - w_l weight of the statistical property S_l

- Free Variable:
  - Assignment variable: x represents the vector of random variables of scenarios of the stochastic process that is matched. x_n is the value of the ransom variable for scenario $n \in 1, \ldots, N$

- Functions:
  - $f_l(x)$ the function representing the calculation of the statistical property S_l as a function of x

4.2.2 Mathematical Description of the Model

A measure of distance between the different statistical properties is been minimized. (For the purpose of this report the square norm is used as a measure of distance.)

Min:

$$\sum_{s_l=s_1}^{s_L} \sum_{n=1}^{N} w_l(f_l(x_n) - S_{l,VAL})^2 \tag{4.1}$$

subject to:

$$x_n \in \mathbb{R} \qquad\qquad n \in 1, \ldots, N \tag{4.2}$$

The levels of freedom of the model can be extended by making the probabilities of each scenarios p_l be a variable as well. However, even though it might look as an increase in the level of freedom, it leads to a further increase in complexity of the model by adding a constraint and making the objective into a more complicated non-linear optimization. Therefore, it is usually not recommended.

4.2.3 Pro Et Contra - Arguments For and Against

This subsection supplies a short overview of the qualities of this approach:

- Arguments For

  + The presented methodology is applicable for many decision problems under uncertainty. The paper [4] by Højland & Wallace in 2001 describes the generation of a scenario tree. However, it can be adjusted to other structures as defined by different decision problems.

  + This approach can easily be extended to other properties that are problem specific. That is a vital property, since it is believed that there is no general scenario generation approach.

    By adding the statistical properties as moments or covariance to the objective problem of a scenario generator these properties can be matched while keeping other constraints that are problem specific.

  + The approach is easily implemented in comparison to more mathematically correct approaches as optimal discretization as suggested by R. Hochreiter and G. Ch. Pflug and scenario tree generation as a multidimensional facility location problem at [17], for example.

- Arguments Against

  - Non linear optimization.

    The optimization problem is generally not convex, therefore, the solution might be (and probably is) local.

However, for our purposes it is in many cases satisfactory to have a solution with distribution properties equal to or specified by the statistical properties at all.

- As shown by Pflug and Hochreiterr at [47] (2003) there could exist different theoretical distributions with the same moments (This is examined in the next subsection.)

As raised by Pflug and Hochreiterr at [47] (2003), in the moment matching approach by Højland and Wallace that is described in this chapter (and at [4]) the first and the second moments are modified before the new approximation is calculated for each node of the tree. It is noteworthy that the approximation is done in a multivariate fashion, i.e. in one step no matter how many variables (e.g. asset, classes, etc.) are estimated for each node. Although this method performs better than random sampling and adjusted random sampling for stochastic asset liability management problems (see Kouwenberg at [48]), it is obviously awkward to use moment-matching in terms of reliability and credibility of the approximations. The accuracy criteria of moment-matching approximation is problematic as seen in the following example.

4.2.4 Different Distribution with the Same Moments

Matching statistical moments, especially matching the first four moments of a probability distribution introduced by Høyland and Wallace is a widespread method, however, moment-matching may lead to strange results as is illustrated below:
The following four distributions coincide in all of their first four moments.

1. A uniform distribution in the interval [-2.44949, 2.44949]

2. The mixture of two normal distributions N(1:244666; 0:450806) and N(-1:244666; 0:450806) with equal weights 0.5

3. The discrete distribution

Value	-2.0395	-0.91557	0	0.91557	2.0395
Probability	0.2	0.2	0.2	0.2	0.2

4. The discrete distribution

Value	-3.5	-1.4	0	1.4	3.5
Probability	0.013	0.429	0.1162	0.429	0.013

These distributions are shown in Figure 4.5. In the left graph distribution 1,2,4 are shown as 1,2,3 in the right graph respectively. Visual inspection shows that these distributions do not have much in common.

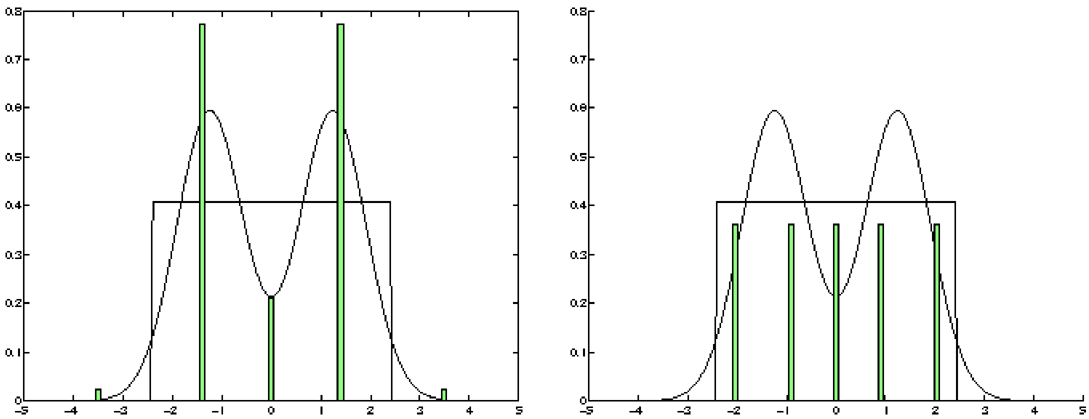


Figure 4.5: Four Distributions with Identical First Four Moments (taken from [47])

Even though the results shown in this subsection raise doubt in the moment-matching approach it is essential to see that scenario generation is used to provide input for the optimization problem. This result does not necessarily means that the scenario generation method is invalid. This mainly means that stability analysis in addition to the scenario generation method should be done in order to examine whether it satisfies the accuracy criteria or not.

4.3 A Heuristic for Moment Matching Scenario Generation

This section will present the heuristic presented by Høyland, Kaut and Wallace at [5] and will be based on their paper. A basic prototype of the algorithm was received and further implemented and tested as part of this thesis.

4.3.1 Motivation

The heuristic developed tries to address some of the following issues.

- In the general form of the algorithm presented by Høyland and Wallace at 4.2, outcomes of all the random variables (assets) are generated simultaneously. Such an approach becomes slow when the number of random variables increases. In this section we have generated one marginal distribution at a time and created the joint distribution by putting the marginal distributions together in the following way: All marginal distributions are generated with the same number of realizations, and the probability of the i^{th} realization is the same for each marginal distribution. The i^{th} scenario, that is, the i^{th} realization of the joint distribution, is then created by using the i^{th} realization from each marginal distribution, and given the corresponding probability. We then applied various transformations in an iterative loop to reach the target moments and correlations.
- As presented at 4.2.4 when using the approach suggested at 4.2 there might be several distributions that can match the moments and be achieved as solutions for a given number of scenarios. The heuristic presented here will start looking for a solution from a normal distribution. That in return will most likely lead to a scenario which represents a real life distribution.

The presented algorithm is inspired by ([58],[57],[56]). Fleishman at ([58]) presents a cubic transformation that transforms a sample from a univariate normal distribution to a distribution satisfying some specified values for the first four moments. Vale and Maurelli at ([56]) address the multivariate case and analyse how the correlations are changed when the cubic transformation is applied. The algorithm assumes that we start out with multivariate normal distribution. The initial correlations are adjusted so that the correlations after the cubic transformation are the desired ones. The heuristic is only approximate with no guarantee regarding the level of the error.

There are, however, two major differences between the two algorithms. One is in the way they handle the (inevitable) change of distribution during the transition to the multivariate distribution while they modify the correlation matrix. In order to end up with the right distribution, the presented heuristic modifies the starting moments. The other major difference is that the previous algorithm starts with parametric marginal distributions whereas the presented heuristic starts with the marginal moments.

4.3.2 The Heuristic

The general idea of the algorithm is as follows:

- Generate n discrete univariate random variables – each satisfying a specification for the first four moments.
- Transform them so that the resulting random vector is consistent with a given correlation matrix.
- The transformation will distort the marginal moments of higher than second order. Hence, we need to start out with a different set of higher moments, so that we end up with the right ones.

Notation

- n - Number of random variables
- s - Number of scenarios
- $\tilde{X}$ - General n-dimensional random variable
  - $\tilde{X} = (\tilde{X}_1, \tilde{X}_2, \dots, \tilde{X}_n)$.
  - Every moment of $\tilde{X}$ is a vector of size n.
  - The correlation matrix of $\tilde{X}$ is a matrix of size n × n.
- $\mathbb{X}$ - Matrix of s scenario outcomes X has dimension n × s.
- $\mathbb{X}_i$ - Row vector of outcomes of the $i^t h$ random variable $\mathbb{X}_i$ has size s.
- $\mathbb{P}$ - Row vector of scenario probabilities given by the user
- $\tilde{\chi}$ - Discrete n-dimensional random variable given by $\mathbb{X}$ and $\mathbb{P}$
- TARMOM - Matrix of target moments (4 × n)
- R - Target correlation matrix (n × n)

The Core Algorithm

The core algorithm runs as follow: Find the target marginal moments from stochastic processes, from statistics or by specifying them directly. Generate n discrete random variables with these moments. Create the multivariate random variable by combining the univariate variables, as explained in [5]. Transform this variable so that it has the desired correlations and marginal moments. If the random variables $\tilde{\chi}_i$ and i were independent, we would end up with $\tilde{Y}$ having exactly the desired properties.

The algorithm is divided into two stages - the input phase and the output phase. In the input phase we read the target properties specified by the user and transform them into a form needed by the algorithm. In the output phase we generate the distributions and transform them into the original properties.

The Input Phase

In this phase we work only with the target moments and correlations. We do not yet have any outcomes. This means that all operations are fast and independent of the number of scenarios. Our goal is to generate a discrete approximation $\tilde{Z}$ of an n-dimensional random variable $\tilde{Z}$ with moments TARMOM and correlation matrix R. Since the matrix transformation needs zero means and variances equal to 1, we have to change the targets to match this requirement. Thus, instead of $\tilde{Z}$ we will generate random variables $\tilde{Y}$ with moments MOM (and correlation matrix R), such that $MOM_1 = 0$, and $MOM_2 = 1$. $\tilde{Z}$ is then computed at the very end of the algorithm as:

$$\tilde{Z} = \alpha\tilde{Y} + \beta$$

It can be shown that the values leading to the correct are:

$$\alpha = TARMOM_2^{\frac{1}{2}}$$

$$\beta = TARMOM_1$$

$$MOM_3 = \frac{TARMOM_3}{\alpha^3}$$

$$MOM_4 = \frac{TARMOM_4}{\alpha^4}$$

The final step in the input phase is to derive moments of independent univariate random variables $\tilde{X}_i$ such that $\tilde{Y} = L\tilde{X}$ will have the target moments and correlations. To do this we need to find the Cholesky-decomposition matrix L, i.e. a lower-triangular matrix L so that

$$R = LL^T$$

The input phase then contains the following steps (figure 4.6):

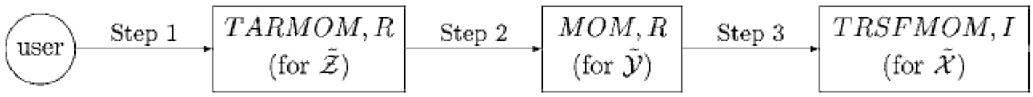


Figure 4.6: Input Phase

1. Specify the target moments TARMOM and target correlation matrix R of $\tilde{Z}$
2. Find the normalized moments MOM for $\tilde{Y}$.
3. Compute L and find the transformed moments TRSFMOM for $\tilde{\chi}$.

The Output Phase

In this phase we start by generating the outcomes for the independent random variables. Next, we transform them to get the intermediate-target moments and target correlations, and finally obtain the moments specified by the user. Since the last transformation is a linear one, it will not change the correlations. All the transformations in this phase are with the outcomes, so the computing time needed for this phase is longer and increases with the number of scenarios.

The output phase then contains the following steps (figure 4.7):

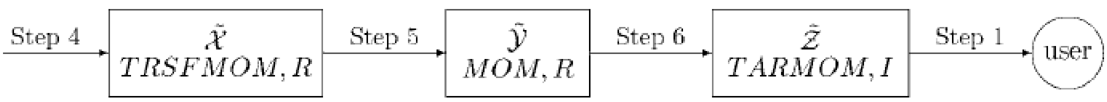


Figure 4.7: Output Phase

4. Generate outcomes $\mathbb{X}_i$ of 1-dimensional variables $\tilde{\chi}_i$ (independently for $i = 1, \ldots, n$).

5. Transform $\tilde{\chi}$ to the target correlations: $\mathbb{Y} = L\mathbb{X}$

6. Transform $\tilde{Z}$ to the original moments: $\mathbb{Z} = \alpha\mathbb{Y} + \beta$

Assumptions

There are two assumptions on the specified correlation matrix R.

1. R is a possible correlation matrix, i.e. that it is a symmetric positive semidefinite matrix with 1's on the main diagonal. While implementing the algorithm there is no need to check positive semi-definiteness directly, as we do a Cholesky decomposition of the matrix R at the very start. If R is not a positive semi-definite, the Cholesky decomposition will fail.

2. The random variables are not collinear, so that R is a nonsingular, hence a positive definite matrix. For checking this property we can again use the Cholesky decomposition because the resulting lower-triangular matrix L will have zero(s) on its main diagonal in a case of collinearity.

Possible Extensions

Regarding the algorithm, the procedure will lead to the exact desired values for the correlations and the marginal moments if the generated univariate random variables are independent. This is, however, true only when the number of outcomes goes to infinity and all the scenarios are equally probable. However, with a limited number of outcomes, and possibly distinct probabilities, the marginal moments and the correlations will therefore not fully match the specifications. To be able to secure that the error is within a pre-specified range, an iterative algorithm was developed, which is an extension of the core algorithm. The extension can be seen in more detail at [5].

4.3.3 Pro Et Contra - Arguments For and Against

– Arguments For

  + Start by a normal distribution and the results look more like a distribution.
  + This paper presents an algorithm that reduces the computing time for the scenario generation substantially. Testing shows that the algorithm finds trees with 1000 scenarios representing 20 random variables in less than one minute.
  + A potential divergence or convergence to the wrong solution is easy to detect. Hence, we never end up using an incorrect tree in the optimization procedure.

– Arguments Against

  - Cannot guarantee convergence. However, experience shows that it does converge if the specifications are possible and there are enough scenarios. The algorithm was run 25 times and the convergence of the algorithm can be seen at figure 4.8. Lines represent average errors after every iteration. Bars represent the best and the worst cases. The dashed lines represent errors in moments after the matrix transformation of the solid line errors in correlations after the cubic transformation.
  - One Stage algorithm. multi-stage the algorithm is not trivial.
  - Complicated to implement.

4.4 Summary

Moment matching scenario generation approach can be very useful as part of a general scenario generation approach. However, as such moment matching in itself does not necessarily fit the consistency criterion of a scenario generator as described at section 3.2. That is because moment matching as described in this chapter is a mathematical method

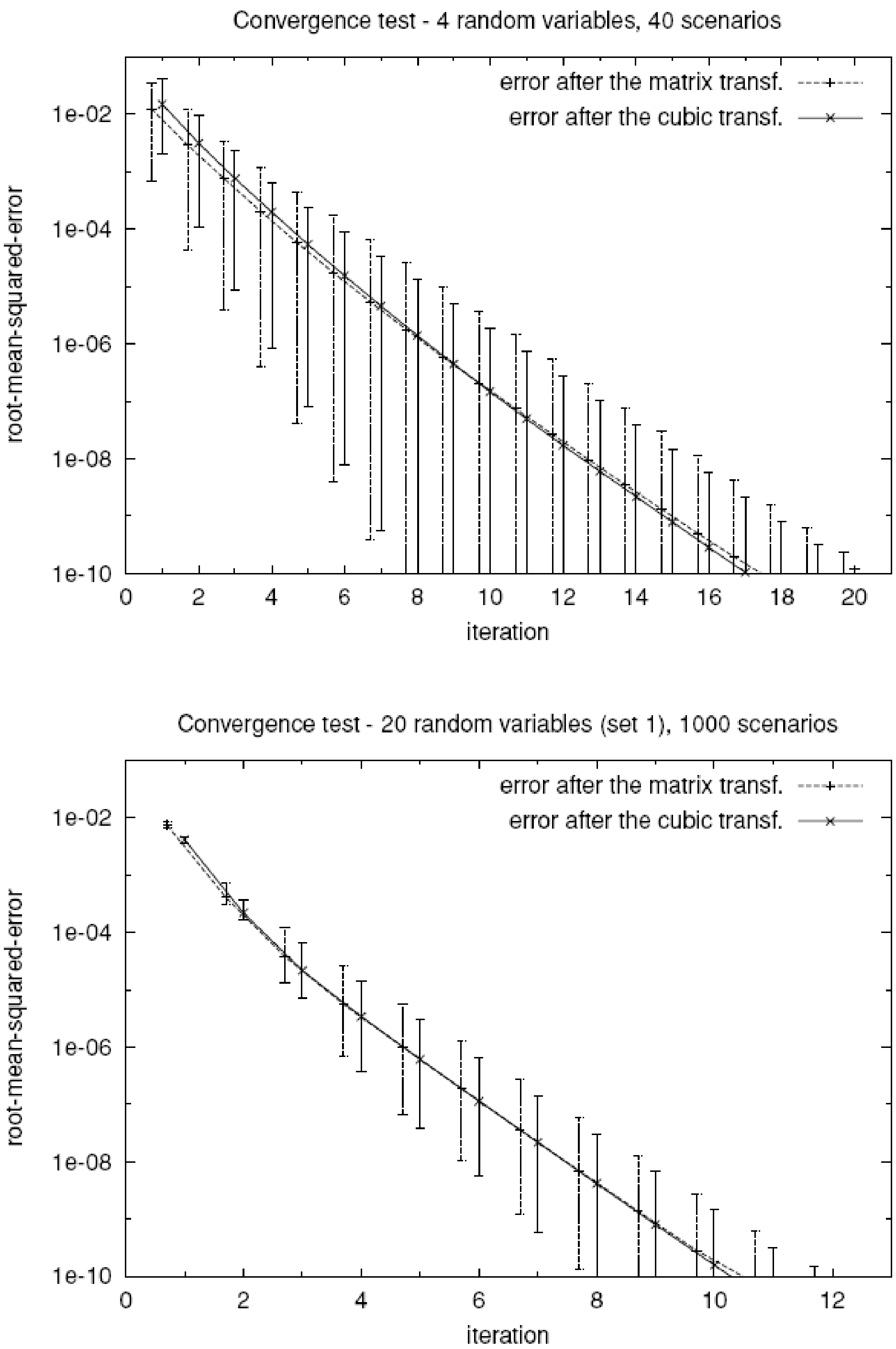


Figure 4.8: Convergence of the Iterative Algorithm (from [5])

and as such does not suggests any financial insight directly. The next chapter will suggest different measures or properties that are essential when building a valid interest rate scenario generation and later at chapter 6 a VAR1 model will be described as a propose for a yield curve scenario discretization model.

During this thesis work, a scenario generator was attempted to be built which would be solely be based upon moment matching. However, it did not led to promising results. Since we do not believe that a sole moment matching approach suggests useful solutions, (examples of such approaches were not given). Nevertheless, such examples can be seen at ([4], [5]). I implemented an example of a moment matching multi-stage stochastic programming approach that was used in this research by Svitlana Sukhodolska as part of her master's thesis project ([40]). These sources give examples for pure moment matching approaches while later on in this report a yield curve scenario generation based on moment matching will be presented. The next chapter will describe the appropriate property of a good interest rate scenario generator. This chapter is the direct consequence of the poor results achieved when creating a scenario generator without building a model based on a thorough understanding of the domain of the solutions.

Chapter 5

Interest Rate Scenario Generation

While the previous three chapters dealt mainly with creating the mathematical background associated with scenario generation. In addition, it described some of the most used scenario generation approaches in general and dealt in more detail with different moment-matching approaches.

As mentioned, a general scenario generation approach that can deal with all sets of problems is believed to not have been found yet. Since this report deals with interest rate scenario generation, this chapter will elaborate on the components that are essential for looking into interest rate scenario generation. As such, it is heavily dependent on the research of Zenios at [14] in financial engineering. The report by Rasmussen and Poulsen at [39] presenting factor analysis of the term structure in Denmark and identifying consistency criteria for an event tree of the yield curve. The subject of arbitrage detection over scenario trees is based on the comments provided by Klassen for moment-matching scenario generation at [38] and an alternative method for arbitrage removal that is further suggested.

5.1 Interest Rate Risk

A scenario generation model for the interest rate is a risk management tool. In order to obtain good qualitative measures of interest rates more thorough interest rate risk fundamentals should be observed as can be seen at [14] and [40], for example.

Interest Rate risk is the potential loss if the price of a security changes over time due to adverse movements of the general levels of interest rates. This risk affects fixed-income as well as all other securities with price dependencies, including interest rates, among other possible factors.

The general level of interest rates is determined by the interaction between supply and demand for credit. If the supply of credit from lenders rises relative to the demand from borrowers, the interest rate falls as lenders compete to find borrowers for their funds. On the one hand, if the demand rises relative to supply, the interest rate will rise as borrowers are willing to pay more for increasingly scarce funds. The principal force of the demand for credit comes from the desire for current spending and investment opportunities. Supply of credit on the other hand, comes from willingness to defer spending. Moreover, central banks are able to determine the levels of interest rates - either by setting them directly or by influencing the money supply - in order to achieve their economic objectives. For example, in the UK, the Bank of England sets the base rate charged to other financial institutions. When it is raised, these institutions follow suit and raise rates to their customers, making it more expensive to borrow and thereby slowing down economic activity. The base rate (also known as the official interest rate) will influence interest rates charged for overdrafts, mortgages, as well as savings accounts. Furthermore, a change in the base rate will tend to affect the price of property and financial assets such as bonds, shares and the exchange rate. The central bank influences the availability of money and credit by adjusting the level of bank reserves and by buying and selling government securities. These tools influence the supply of credit, but do not directly impact the demand for it. Therefore, central banks in general are not able to exert complete control over interest rates.

Inflation is also a factor. When there is an overall increase in the level of prices, investors require compensation for the loss of purchasing power, which means - higher nominal interest rates. As agents are supposed to base their decisions on real variables, it is the equilibrium between real savings and real investments that will determine the real interest rate. Hence, if this equilibrium remains the same, movements in the nominal interest rate should reflect movements in the prices or in expected future prices.

Another important factor is credit risk, which is a possibility of a loss resulting from the inability to repay the debt obligation. The larger the likelihood of not being repaid, the higher the interest rate levels are.

Time is also a factor of risk and it consequently has an influence on the level of interest rates.

It is common to distinguish between short-term rates - for lending periods shorter than one year - and long-term rates for longer periods. Long-term rates are typically decomposed into two factors: the expected future level of short-term rates and a risk premium to compensate investors for holding assets over a longer time frame. As a result, yields on long-dated securities are in general (but not always) higher than short-term rates.

Figure 5.1 captures all the detrimental risk factors influencing the interest rate levels, summarizing the above study in accordance.

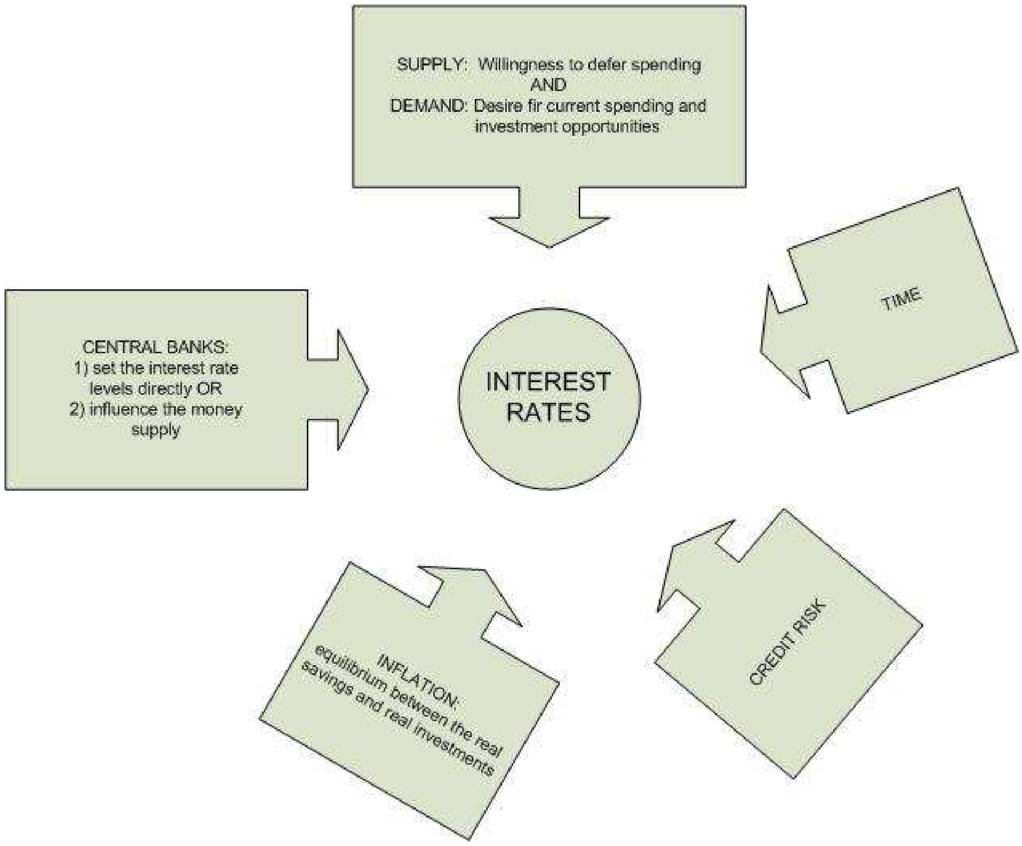


Figure 5.1: Detrimental Factors of Interest Rate Risk (from [40])

5.2 Arbitrage and Arbitrage Tests

5.2.1 Overview of Arbitrage

In this section arbitrage will be considered in detail as well as the way to integrate arbitrage tests as part of an optimization model.

In finance, arbitrage is the practice of taking advantage of a price differential between two or more markets: a combination if matching deals are struck that capitalize upon that imbalance, the profit being the difference between the market prices.

However, when used by academics, an arbitrage is a transaction that involves no negative cash flow at any probabilistic or temporal state and a positive cash flow in at least one state; in simpler terms, a risk-free profit. A person who engages in arbitrage is called an arbitrageur. The term is mainly applied to trading in financial instruments, such as bonds, stocks, derivatives and currencies. If the market prices do not allow for profitable arbitrage, the prices are said to constitute an arbitrage equilibrium or arbitrage free market. An arbitrage equilibrium is a precondition for general economic equilibrium.

When looking into arbitrage usually two different arbitrage types are considered.

- Arbitrage type 1 which represents buying a portfolio of instruments at price 0 that will create non-negative future cash flows and a positive cash flow at no less than one future point.

- Arbitrage type 2 which represents buying a portfolio at a negative price (profit at time of buying) that will create non-negative future cash flow.

In some situations, it is straightforward to turn the identified arbitrage opportunity of the first type into an arbitrage opportunity of the second type. In general, however, the existence of an arbitrage opportunity of the first type does not imply the existence of an arbitrage opportunity of the second type, or vice versa. Therefore, the two types are treated separately.

The following subsections will show the motivation for performing arbitrage detection in an ALM problem as well as describe arbitrage in a more academic manner as an operations research problem in order to develop a model to remove arbitrage.

5.2.2 Motivation - The Importance of Arbitrage Test in ALM Problems

Klaassen at [24] emphasized the importance of precluding arbitrage opportunities when the scenario-generation method of Høyland and Wallace from [4] is applied to asset allocation problems under uncertainty. The presence of arbitrage opportunities will unrealistically bias optimal asset allocations. He has shown that arbitrage opportunities can either be detected ex-post by checking for solutions to sets of linear equations, or precluding ex-ante by adding constraints to the optimization program that is formulated to generate the scenario tree. This process is described in more detail in the continuation of the section. In addition to the research, a simple heuristic is presented to remove arbitrage from the scenario tree. This heuristic is now used as part of the VAR1 interest rate scenario generation process that will be described in the next chapter.

The importance of precluding arbitrage opportunities in scenario trees of asset returns for portfolio optimization problems under uncertainty has been illustrated in Klaassen [46]. If arbitrage opportunities are present, the optimal solution will exploit these to the maximum extent possible. It is unlikely, however, that the arbitrage opportunities will arise in reality, and hence the optimal solution will reflect spurious profit opportunities.

The rest of this section will introduce the two types of arbitrage and their corresponding dual problems that can be used as part of the arbitrage removal process. Finally, the arbitrage removal processes will be further discussed and the heuristic to preclude arbitrage will be shown.

5.2.3 Arbitrage of Type 1

Description of Model Data

The arbitrage detection problem: primal problem The data is:

- Sets (indices)
  - Time: $t \in 1, \ldots, T$
  - Bonds [1]:$k \in 1, \ldots, K$
  - Nodes of the event tree or state of the world[2]: $n \in 1, \ldots, N$
- Data:
  - $R^n_{k,t+1} \geq 0$ Represents the return of Bond k between time periods t and $t+1$ for node n.
- Free Variable:
  - Assignment Variable: $x_{k,t}$ represents the holding of the bond k at time t.

Mathematical description of the model

Ingersoll at [45] (1987) describes an arbitrage opportunity of the first type as one that exists between the time periods t and $t+1$ if there is an asset allocation $x_t = (x_{1,t}, \ldots, x_{K,t})$ such that:

[1] Can be generalized to other financial instruments but is thought of as bonds for the purpose of this report

[2] In stochastic programming problems and event trees, there are many possible states of the world between two time periods

$$\sum_{k=1}^{K} x_{k,t} = 0, \tag{5.1}$$

$$\sum_{k=1}^{K} x_{k,t} R_{k,t+1}^{n} \geq 0, \qquad \forall n \in 1, \ldots, N, \tag{5.2}$$

$$\sum_{k=1}^{K} x_{k,t} R_{k,t+1}^{n} > 0, \qquad \exists n \in 1, \ldots, N, \tag{5.3}$$

The Model - Arbitrage of type 1 as an operations research model

The following is a representation of the model as a maximization problem over scenario tree:

Max:

$$\sum_{t=1}^{T} \sum_{n=1}^{N} x_{k,t} R_{k,t+1}^{n} \tag{5.4}$$

subject to:

$$(5.1), (5.2), x_{k,t}, \in \mathbb{R} \tag{5.5}$$

Correctness of the optimization problem

Intuitively the objective is to find holding of bonds that will maximize the total return (obj function 5.4) under several conditions. The first condition at equation 5.1 ensures that no money is invested in the portfolio represented by x at any time point but the portfolio is balanced for selling and buying for the total cash flow of zero. The second condition at equation 5.2 is looking at the return at each time point and ensures that the total return of the portfolio is non-negative at every time point.

Theorem: The optimization problem is unbounded if, and only if, arbitrage opportunity of type 1 exists.

Proof: $\Rightarrow$ Assuming the optimization problem is unbounded then the portfolio selection that is presented by the solution is an arbitrage opportunity of type 1. Since it costs nothing and has a positive return at no less than one future point.

$\Leftarrow$ Assuming an arbitrage opportunity of type 1 exists then there exists a combination of buying and selling of a portfolio x for price 0 that will yield a return larger or equal than zero for each of the future time points. That x is a valid solution for the optimization problem that will yield an objective value > 0 lets call that value c. The buying of x can then be scaled by any factor $\lambda \geq 0$ and yield another feasible solution. and will yield an objective value of

$$\sum_{n=1}^{N}\sum_{k=1}^{K}\lambda x_{k,t}R^{n}_{k,t+1} = \lambda\sum_{n=1}^{N}\sum_{k=1}^{K}x_{k,t}R^{n}_{k,t+1} = \lambda c$$

For $\lambda \to \infty$ it concludes that

$$\lim_{\lambda\to\infty}\lambda c = \infty$$

Since λx is a valid solution since it satisfies 5.1 (just multiply the equation by λ) and since it satisfies 5.2 since it is a multiplication by a positive constant $\lambda > 0$ the equations also holds for λx

Therefore, there exists a series of solutions that converge to ∞ and the problem is unbounded.

□

The dual problem

It can easily be shown, after multiplying equation 5.2 at minus one that the equivalent dual problem is:

Min:

$$0 \tag{5.6}$$

subject to:

$$\pi_0 - \sum_{n=1}^{N} \pi_n R^n_{k,t+1} = \sum_{n=1}^{N} R^n_{k,t+1}, \qquad \forall k \in 1,\dots,K, \tag{5.7}$$

$$\pi_0 \in \mathbb{R}, \pi_n \geq 0 \qquad \forall n \in 1,\dots,N \tag{5.8}$$

Conversely, if this dual program does have a feasible solution, strong duality implies that for any feasible asset allocation x_t:

$$\sum_{n=1}^{N} \sum_{k=1}^{K} x_{k,t} R^n_{k,t} \leq 0$$

Thus, no arbitrage opportunity of the first type exists.

5.2.4 Arbitrage of type 2

Mathematical description of the model

Using the same notations as for arbitrage of the first type Ingersoll at [45] (1987) describes an arbitrage opportunity of the second type that exists between the time periods t and $t+1$ if there is an asset allocation $x_t = (x_{1,t}, \dots, x_{K,t})$ such that:

$$\sum_{k=1}^{K} x_{k,t} < 0 \tag{5.9}$$

$$\sum_{k=1}^{K} x_{k,t}(1 + R_{k,t+1}^{n}) \geq 0, \qquad \forall n \in \{1, \ldots, N\}, \tag{5.10}$$

The Model - Arbitrage of type 2 as an operations research model

The model is:

Min:

$$\sum_{k=1}^{K} x_{k,t} \tag{5.11}$$

subject to:

$$(5.10), x_{k,t} \in \mathbb{R} \tag{5.12}$$

Correctness of the optimization problem

Intuitively the objective for the problem at 5.11 aims to receive a positive return at time zero by the pick of a bonds portfolio. A negative objective presents a positive return for the investor at time 0. (Finding a portfolio for a negative price is receiving money at time 0.) The condition at equation 5.10 shows that the cash flow received by the investor at future time points is non-negative in the same matter as equation 5.2 in the first arbitrage model.

Therefore, if this linear program has a solution with a negative objective value, subsequently there is an arbitrage opportunity of the second type. The linear program will in fact be unbounded as we can multiply the asset allocation x_t by an arbitrary positive constant without violating the constraints (The correctness of this proof can be shown in a similar way as showing the arbitrage of type 1 and therefore will be ignored.). This model is unbounded if, and only if, arbitrage of type 2 exists.

Hence, according to the duality theory, in this case, the dual of this linear program will not have a feasible solution.

The dual problem

The following dual problem is defined:

Max:

$$0 \tag{5.13}$$

subject to:

$$\sum_{n=1}^{N} \nu_n(1 + R^n_{k,t+1}) = 1 \qquad \forall k \in 1, \ldots, K \tag{5.14}$$

$$\nu_n \geq 0 \qquad \forall n \in 1, \ldots, N \tag{5.15}$$

Conversely, if this dual program does have a feasible solution, strong duality implies that for any feasible asset allocation x_t must have:

$$\sum_{k=1}^{K} x_{k,t} \geq 0$$

Thus, no arbitrage opportunity of the second type exists.

5.2.5 Conclusion

As described in this section the existence of arbitrage should be addressed during the scenario generation process according to the three methods suggested in this report.

1. **Rerun the model with different starting point**

   As implied from the dual problems. One can check for the existence of solutions to these equations after a scenario tree is generated (for example by using moment-matching as described in previous chapters based in Høyland and Wallace). In a multi-period problem, one has to check for solutions to the sets of linear equations in each node n at every date t of the scenario tree before the model horizon. A useful result is that if the set of equations (5.14) has a strictly positive solution , then no arbitrage opportunities of either the first or second type are present (see Ingersoll 1987, p. 57). If an arbitrage opportunity is encountered, one can apply the scenario-generation method again (using a different starting point) in the hope that an arbitrage-free scenario tree is found. One may also have to increase the number of scenarios in the tree.

2. **Add arbitrage removal constraints**

   Alternatively, One could add equations (5.7),(5.8) and (5.14),(5.19) as constraints to the nonlinear optimization program of Høyland and Wallace. One will then preclude arbitrage opportunities of both types in the scenario tree that is generated. As asset returns $R^n_{k,t+1}$ are variables in the optimization program of Høyland and Wallace (2001), as presented in chapter 3, equations (5.7) and (5.14) represent nonlinear constraints if added to this optimization program. This will therefore complicate the numerical optimization of the nonlinear programming model.

3. **Run arbitrage removal on the results**

   An alternative can also be an arbitrage removal process that can be run after the optimization in order to remove arbitrage. This heuristic does not impose optimality of the results, since the arbitrage removal effects the interest rate scenarios.

When examining the three approaches presented above the third approach was chosen for this project since the second approach adds non-linear constraints to a problem that was not linear to begin with; it was rejected. In view of the fact that the first approach suggests arbitrage detection and rerun of the process if arbitrage is found with a different starting point and this process needs to be repeated for each subtree of the multi-stage interest rate tree, it was decided to be too time consuming and hard to implement. (It is not trivial to detect another good starting point in order to impose a non-arbitrage solution for each case.)

The arbitrage removal process is discussed in more detail in the following section.

5.3 Arbitrage Removal

This section will introduce conditions for arbitrage detection and arbitrage removal as an optimization problem. This section will try to define this problem as an operations research problem and would use financial theory only for creating intuition for results presented here. Nevertheless, this section is based on solid financial theory on arbitrage and asset pricing. (These issues can be further seen at Lando and Poulsen at [9] for example).

5.3.1 Arbitrage Free Asset Pricing on an Event Tree

This section does not intend to provide a solid financial background for examining arbitrage detection and arbitrage pricing. However, a few of the theories would be mentioned on a very wide perspective in order to give some kind of intuition for the process.

Without looking deeply into arbitrage free theory it should be mentioned that the results that are used in this chapter are based on the following theory and prepositions taken from Lando and Poulsen at ([9])

Theorem 2: The security market is arbitrage free if and only if there exists a strictly positive vector $d \in \mathcal{R}^T_{++}$ such that $\pi = C \cdot d$, where d is a vector of discount factors.

The key to this theorem is:

Lemma 1 (Stiemke's lemma): Let A be an $n \times m$ matrix. Then precisely one of the two following statements is true:

1. There exists $x \in \mathcal{R}^m_{++}$ such that $Ax = 0$.
2. There exists $y \in \mathcal{R}^n$ such that $y^T A > 0$.

Theorem 3: Assume that (π, C) is arbitrage free. Then the market is complete if and only if there is a unique vector of discount factors.

A market is complete if any desired payment stream can be generated by an appropriate choice of portfolio.

Proposition 18: The security market model is arbitrage free if and only if the one period model is arbitrage free.

The usefulness of this local result is that we often build multi period models by repeating the same one period structure. We may then check absence of arbitrage and completeness by looking at a one period submodel instead of the whole tree. Then the detection process can be repeated recursively throughout the event tree.

5.3.2 An Example of Arbitrage Removal in a Tree

In order to detect and remove arbitrage in the tree. The interest rates are transfered to bond prices. Below an axle of arbitrage detection and removal is presented:

Consider the subtree in Figure (5.2). Loans 1 to 3 are fixed rate mortgages (FRMs) whereas loan 4 is an adjustable rate mortgage (ARM) with annual re–financing. The prices in the

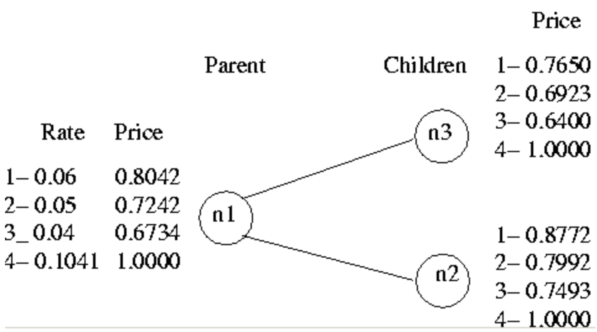


Figure 5.2: A Subtree with Information on Rates and Prices.

children nodes are already decided. We would like to check whether the tree is arbitrage free.

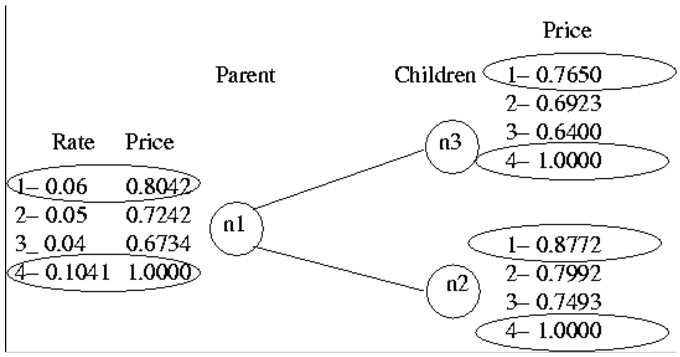


Figure 5.3: A Subtree with Information on Rates and Prices.

For the purpose of this example we choose the loans 1 and 4 as arbitrage–free pricing references to represent the short and the long end of the interest rate scale. We will show in the following how to change the prices of loans 2 and 3 in order for the subtree to become arbitrage–free.

Let the vector π denote the price vector for loans 1 and 4:

$$\pi_i = \begin{pmatrix} 0.8042 \\ 1.0000 \end{pmatrix} \qquad i \in \{1, 4\},$$

and let the matrix D denote the cashflow matrix for loans 1 and 4:

$$D_{ip} = \begin{pmatrix} 0.8250 & 0.9372 \\ 1.1041 & 1.1041 \end{pmatrix} \quad i \in \{1,4\}, p \in \{1,2\},$$

where every element of the matrix is defined as: $D_{ip} = r_i + Price_{ip}$.

In order for the price vector π to be arbitrage free the discount vector ψ which is the solution to the equation

$$\sum_{p=1}^{2} D_{ip} \cdot \psi_p = \pi_i \quad \forall i \in \{1,4\}$$

must be positive. Solving this linear system of equations we get:

$$\psi_p = \begin{pmatrix} 0.397827 \\ 0.507888 \end{pmatrix} \qquad p \in \{1,2\},$$

which is obviously positive meaning that loans 1 and 4 are priced arbitrage free in the subtree.

the risk neutral probabilities (martingale measures) are found using the following relation:

$$q_p = \frac{\psi_p}{\Psi} \quad \forall p \in \{1,2\},$$

where Ψ is defined as:

$$\Psi = \sum_{p=1}^{2} \psi_p.$$

The risk neutral probabilities thus become:

$$q_p = \begin{pmatrix} 0.439241 \\ 0.560759 \end{pmatrix} \quad p \in \{1, 2\}.$$

A 1–period tree with p states need only two instruments to find the martingale measures. Using these martingale measures we can find the arbitrage–free price of all the other instruments available in the tree.

Let D^{rest} denote the cash flow matrix for the loans 2 and 3:

$$D^{rest}_{ip} = \begin{pmatrix} 0.7423 & 0.8492 \\ 0.6800 & 0.7893 \end{pmatrix} \quad i \in \{2, 3\}, p \in \{1, 2\}.$$

The arbitrage–free prices of loans 2 and 3 are thus found as follows:

$$\pi^{rest}_i = \frac{\sum_{p=1}^{2} D^{rest}_{ip} \cdot q_p}{1 + r^f} \quad \forall i \in 2, 3,$$

where r^f denotes the risk free rate of return in the 1–period tree in question. Since loan 1 provides the same cash flow in both states we can use this rate as the risk free rate, so we get the arbitrage–free prices for loans 2 and 3:

$$\pi^{rest}_i = \begin{pmatrix} 0.726606 \\ 0.671398 \end{pmatrix} \quad i \in \{2, 3\}.$$

5.3.3 Removing Arbitrage as an Operations Research Problem

As described no arbitrage is a necessary condition for markets to be efficient.

Based on these theories the arbitrage detection can be described as operations research problem. The operations research algorithm will try to find the positive vector φ_k for all bonds $1, \ldots, K$ That satisfies the fact that the prices of the child nodes ($PC_{k,n}$) [3] and the price of the parent node ($PPrice_k$) following this formula:

$$PriceP_k = \sum_k \varphi_k PC_{k,n} \forall n$$

An operations research problem can be created to detect and remove arbitrage at once.

One can define a new variables $PC^{+}_{k,n}$ and $PC^{-}_{k,n}$ detecting the deviation in measure of the square norm between the calculated $PC_{k,n}$ and the arbitrage–free $PC^{'}_{k,n}$.

Min:

$$\sum_{k=1}^{K} \sum_{n=1}^{N} (PC^{+}_{k,n} + PC^{-}_{k,n})^2 \tag{5.16}$$

subject to:

$$PriceP_k = \sum_k \varphi_k PC^{'}_{k,n} \forall n \in \{1, \ldots, N\} \tag{5.17}$$

$$PC^{'}_{k,n} = PC_{k,n} + PC^{+}_{k,n} - PC^{-}_{k,n} \forall k \in \{1, \ldots, K\}, \forall n \in \{1, \ldots, N\} \tag{5.18}$$

$$\varphi_k, PC^{+}_{k,n}, PC^{-}_{k,n} \geq 0 \forall k \in \{1, \ldots, K\}, \forall n \in \{1, \ldots, N\} \tag{5.19}$$

A few comments in regards for the described model

- The use of a quadric objective function.

  The problem could also be solved using a linear objective function. However, since the rates are computed using a method as moment matching. A solver is very likely to change one of the rates as much as possible and the next one and so on until it finds an optimal solution. This approach might lead for non optimal solution from

[3] n represents child scenarios in accordance with the terminology used in this report.

the practical view. That is because the rates where chosen using a scenario generation method as moment matching. The scenario generation method keeps track of inter scenario information, such as covariance. A major change in one scenario is very likely to lead for a problem in the correctness of the scenario generation. Therefore, a quadric objective function would ensure that it is optimal from the solver perspective to try and change the scenario so they would be as close as possible to the value calculated by the scenario generator. (the square norm is used as a measure of distance.). However, it would be possible to keep a piecewise linear function or a non quadric objective function. The solution that would be achieved would be arbitrage free but not necessarily lead for good scenarios.

– The arbitrage removal process should be run recursively starting at the the root node of the scenario tree and going forward. (In order to keep the structure of the tree after the arbitrage removal correctly.)

5.4 Factor Analysis of the Term Structure

5.4.1 Motivation

As shown by Zenios at [14] the yields of short and long maturity bonds are not perfectly correlated as can be seen at figure 5.4 . Small and parallel shifts are insufficient for describing changes of the term structure in modern fixed income markets. Therefore, one that considers an interest rate model should make sure of finding a solution to take care of the shape risk created, such as a factor analysis model for the term structure.

Luckily financial observation obtains three eigenvalues that accounts for most of the changes in the term structure of the interest rate. These are - parallel shifts in level, changes in steepness and convexity. These changes might be different from market to market and from pe-

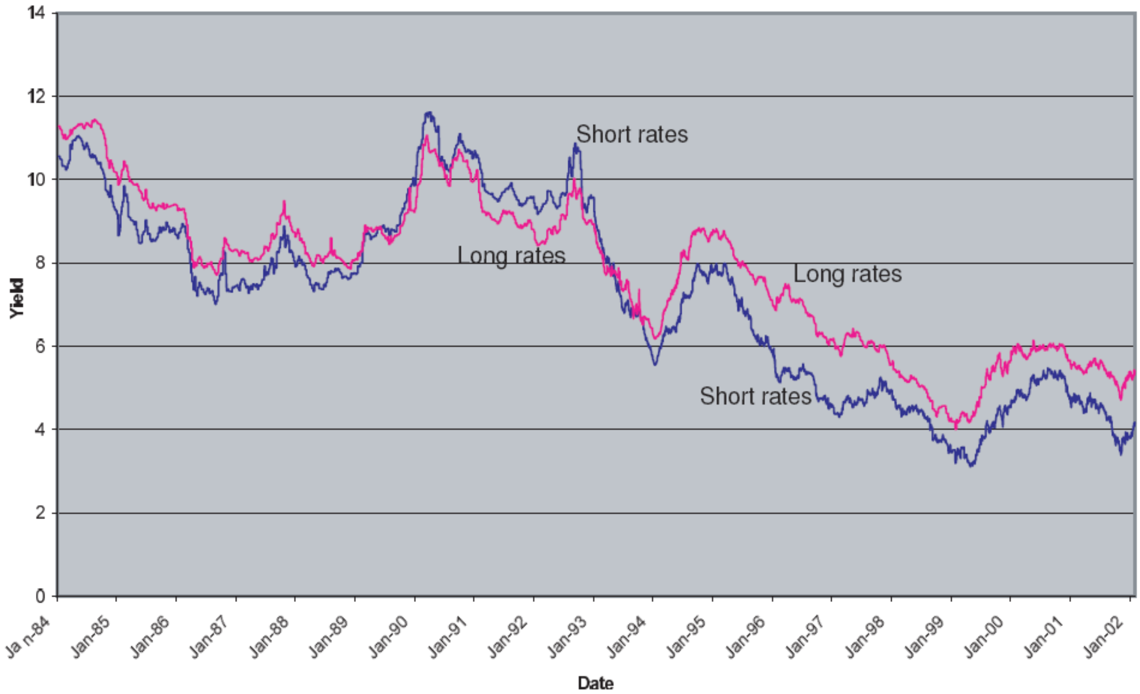


Figure 5.4: The Yields of Short and Long Maturity Bonds are not Perfectly Correlated Giving Rise to Shape Risk (from Zenios at [14])

riod to period. For example, the factor loading of the Italian BTP market is shown at figure 5.5

5.4.2 Principal Component Analysis (PCA)

Factor analysis, also known as principal component analysis (PCA), is a statistical technique to detect the most important sources of variability among observed random variables. Factor analysis may be used on a historic time series of a multidimensional random variable to decide how much variability is explained by different factors or principal components and to order them accordingly. In linear algebraic terms it is an orthogonal linear transformation that transforms data to a new coordinate system in such a way that the greatest variance lies on the first coordinate, called the first principal component, the sec-

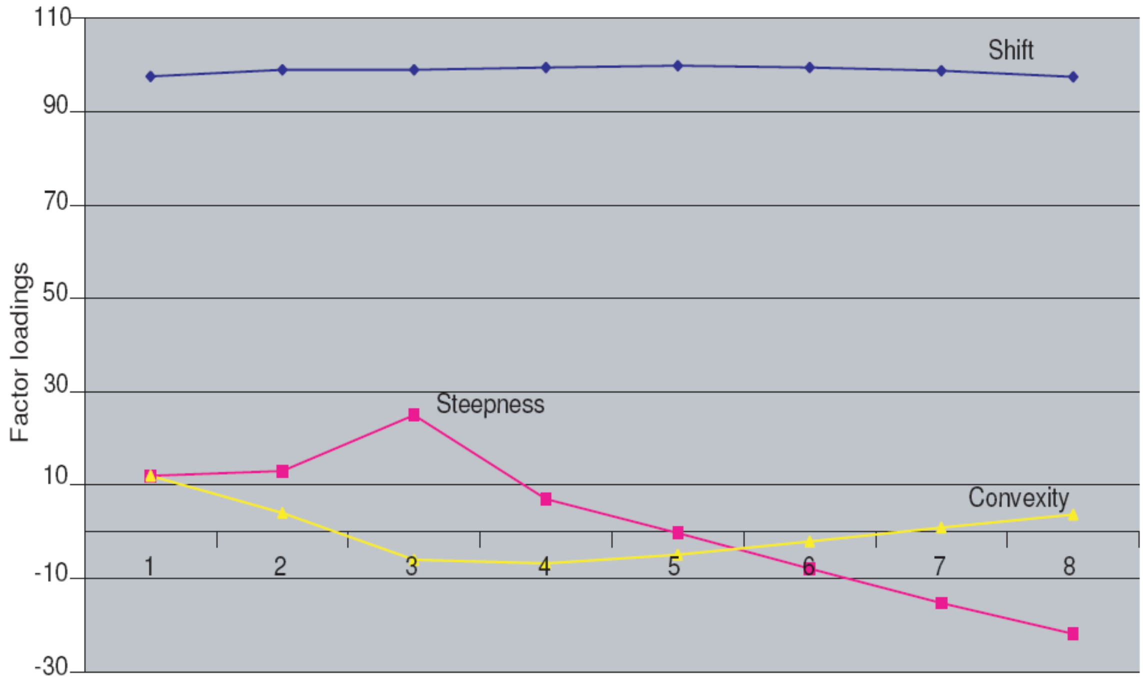


Figure 5.5: Factor Loading Corresponding to the Three Most Significant Factors of the Italian BTP Market (from Zenios at [14])

ond greatest variance on the second principal component and so on. It is used for reducing the dimensionality of a data set while keeping its characteristics. This is done by keeping only some numbers of the first principal component while ignoring the remaining ones that only explain an insignificant proportion of the variance.

Definition: Principal Components of the term structure. Let $\tilde{r} = (\tilde{r}_t)_{t=1}^{T}$ be the random variable presenting the spot rates, and Q be the $T \times T$ covariance matrix. An eigenvector of Q is a vector $\beta_j = (\beta_{jt})_{t=1}^{T}$ such that $Q\beta_j = \lambda_j \beta_j$ for some constant λ_j called an eigenvalue of Q. The random variable $f_j = \sum_{t=1}^{T} \beta_{jt} r_t$ is a principal component of the term structure. The first principal component is the one that corresponds to the largest eigenvalue, the second to the second largest, etc.

As can be seen from the definition, in order to observe the most significant factors, a statistical analysis of the market should be performed. The report is mainly concerned with an implication of the term structure and the factor analysis model as shown at Rasmussen & Poulsen [39].

Litterman & Scheinkman (1991) at [42] and P. J. Knez & Scheinkman (1994) at [41] use factor analysis to show that three factors explain - at a minimum - 96% of the variability of excess returns on several American zero coupon yield curves in the period from 1985 to 1988. Dahl (1994) at [43] show similar results for the Danish data in the 1980's and Bertocchi & Zenios (2005) at [44] repeat the experiments for American and Italian data during 1990's with similar results.

These findings are used by some practitioners to improve duration hedging (immunization) by factor based duration hedging (factor immunization). The main shortcoming of these hedging techniques is that they are myopic and do not consider the re-balancing effects in long term fixed income portfolio investments. Rather than using factor analysis to shape risk hedging, we use factor analysis as a means of finding a sufficient number of factors to be used as the underlying factors of uncertainty for the proposed interest rate model of this paper. Factor analysis on the Danish yield curves for the period 1995–2006 was performed by Rasmussen & Poulsen at [39]. Similar to earlier works, it has been identified that three factors are enough to capture almost all variability (99.99%) for the Danish yield curves. Figure 5.6 shows the factor loadings as a function of maturities in years based on the rates from figure 5.7.

The first factor explains almost 95% of all variability. It can be interpreted as a slight change of slope for interest rates with maturities under 5 years together with a parallel shift for the rest of the curve. The second factor, explaining 4.7% of the variability, corresponds to a change of slope for the whole curve. However, the slope change for the first 10 years is much more pronounced. Finally, the third factor corresponds to a change of curvature in

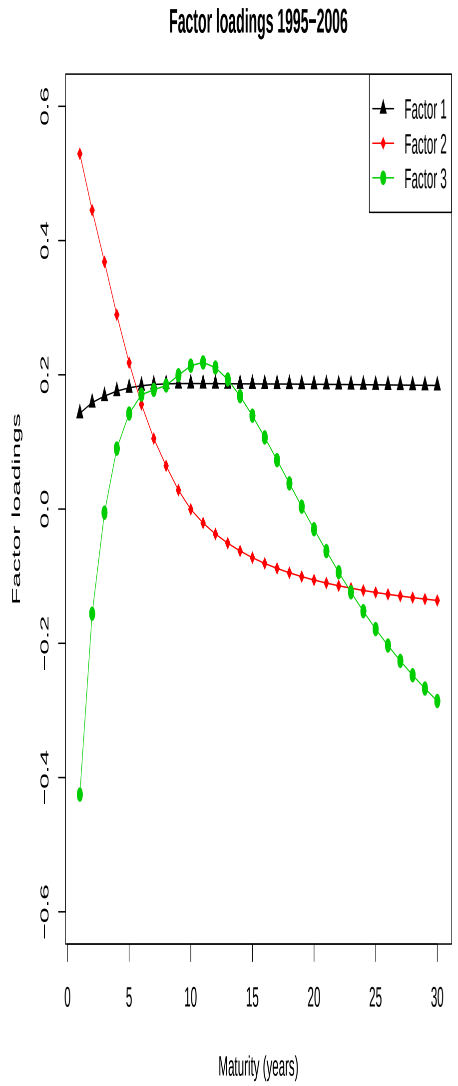


Figure 5.6: Factor Loadings of the Danish Yield Curves for the Period 1995 to 2006. (taken from Rasmussen & Poulsen at [39])

the yield curves. This factor explains only about 0.3% of the total variability.

From a statistical viewpoint we could suffice with level and slope as the main sources of variability. Nevertheless we do not reject the third factor, curvature, due to its economical appeal; changes of curvature are observed now and then, and a model not being able to

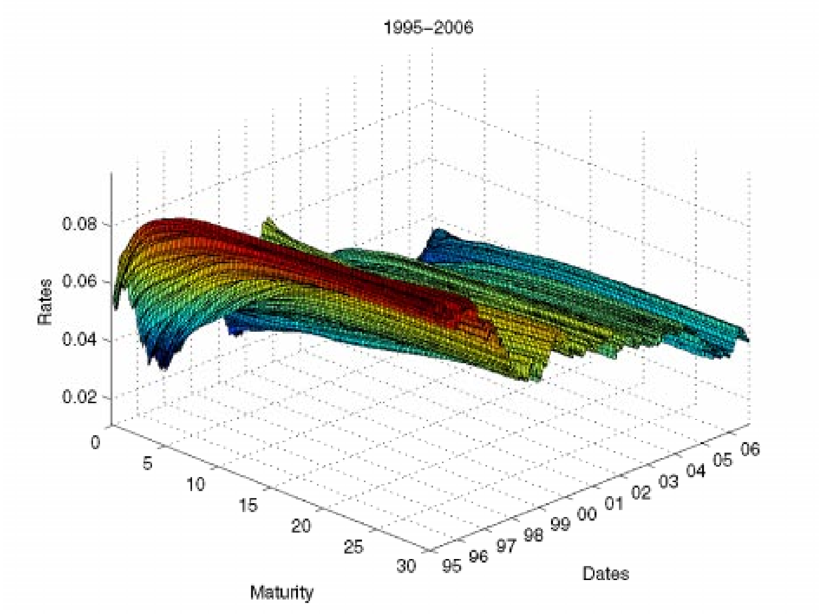


Figure 5.7: 3-Dimensional View of the Danish Yield Curve for the Period 1995-2006 (taken from Rasmussen & Poulsen at [39])

represent those changes properly has a potential of not capturing important movements in the interest rate market.

The interest rate model that was created with Nykredit was inspired by the results found in this section that was based on the definition of the following three factors:

1. Level: An arbitrary rate such as the one year rate,Y_1, may be used as a proxy for level.
2. Slope: A good proxy for the slope would be $Y_{30} - Y_1$ where Y_{30} stands for the 30 year rate. This expression is an approximation of the average slope of the yield curve.
3. Curvature: The expression $Y_5 - (\omega Y_1 + (1 - \omega)Y_{30})$, with Y_5 as the 5 year rate, may be used as a proxy for the curvature. ω is the weight corresponding to the proportion of the distance in between the middle to the long rates. It was chosen so that the curvature would be zero if the curve is a straight line, negative if the curve was convex and positive if the curve was concave.

In the rest of this report the terms level, slope and curvature are defined as above as the factors of the interest rate model in question that will be presented by the VAR (Vector Autoregressive Model) of the interest rate will be produced in the next chapter.

5.5 Smoothing the Term Structure

Of course models for term structure that perform prediction over the three factors described at previous section need to be able to extend their results for the all term structure. It is easy to see that the transformation between the 1-year, 5-year and 30-year rates to level, slope and curvature is easily invertible. However, after reverting back from the three factors to the 1-year, 5-year and 30-year rates a method should be used to further extended the rates to the complete term structure. This type of method is called a smoothing method.

This section suggests briefly two alternatives for performing smoothing the Nelson-Siegel (from Hurn, Lindsay and Pavlov at [67]) and an Affine Smoothing from (Bester 2004 at [59]).

These approaches are discussed briefly below:

Nelson-Siegel Smoothing:

The classical term-structure problem requires the estimation of the smooth yield curve $l = y(\tau)$ from observed bond prices. In recent years the method of choice has been to compute the implicit forward rates required to price successively longer maturity bonds at the observed maturities. These are called unsmoothed forward rates. The smoothed forward rate curve is then obtained by fitting a parametric functional form to these unsmoothed rates. One common choice proposed by Nelson and Siegel (1987 at [66]) is

$$f(u) = \beta_1 + \beta_2 e^{-\lambda u} + \beta_3 \lambda u e^{-\lambda u} \tag{5.20}$$

When β_1, β_2 and β_3 denoting the level, slope and curvature of the yield curve respectively.

Affine Smoothing:

Modern term structure modeling began with Vasicek (1977 at [60]) and Cox, Ingersoll, and Ross (CIR model at [61], 1985). Their work was later extended to the broader affine class of models (Duffie and Kan (1996) at [62]), which were classified in a convenient hierarchy by Dai and Singleton (2000 at [63]). Affine models are distinguished by the assumption that the spot rate and the instantaneous covariances of yields are linear in a finite set of diffusive state variables. These models have enjoyed a long and productive life in the finance literature, due in large part to their eminent tractability. They offer convenient forms for bond prices, yields, and forward rates, and are easily adapted to price interest rate derivatives (see Duffie, Pan, and Singleton (2000) at [64]). Unfortunately, affine models suffer from potentially serious empirical shortcomings. Chan, Karloy, Longstaff and Sandard et al. (1992 [65]) observe that the Vasicek and CIR models fail to capture the stochastic volatility in short-term interest rates.

The model presented in the next chapter uses Nelson-Siegel or affine smoothing in order to receive a complete yield curve.

5.6 Summary

Scenario generation methods are problem specific, while the previous chapters presented general methods for scenario generation. This chapter focused on the properties that will assure consistency of interest rate scenario tree generation. As experienced by our work and would be evaluated later on at this report. The importance of these properties can not be overestimated. As operations research solvers are design in order to exploit issues as incompleteness of the market (e.g. arbitrage opportunity). That in return would lead to unrealistic results when running an optimization model on top of non consistent scenario generation methods.

Up to this point the reader has followed an overview of the most used scenario generation techniques as well as some scenario generations quality measures. This chapter concludes with several measures that are essential for appropriate scenario generation.

- Arbitrage–free pricing
- Principal component analysis and factor analysis of the term structure.
- Smoothing of the term structure

These measures are found during the process of research as purposed in this thesis. They are also described in the paper by Rasmussen and Poulsen [39] that explores yield curve event tree construction for multi-stage stochastic programming problems.

The next chapter would describe a model for yield curve scenario generation.

Chapter 6

Develop a Three Factor VAR1 Interest Rate Scenario Generation Model

This chapter proposes an overall framework for building a yield curve event tree and testing whether or not the consistency criteria are respected.

There is a vast amount of literature on interest rate modelling (see James & Webber at [50] and Brigo and Mercurio at [51] for a review). These models can in general be categorized as being discrete or continuous, normal or log-normal, 1–factor or multifactor, and generally either more theoretically or more empirically inclined. What all such models have in common is the fact that they have been originally developed either for estimating current prices of interest rate sensitive assets, or for prediction purposes. None of the standard models therefore have been designed in order to construct yield curve event trees but rather fulfilling a lattice. That in turn leaves out some issues such as arbitrage free pricing by the natural construction of the lattice.

In section 3.2 the criteria for good scenario generation are described. It is also identified that a moment matching scenario generation, as well as most other mathematical scenario generation, does not imply correctness in the tree construction. However, in this chapter the constructed vector autoregressive with leg 1 (VAR1) model will deal with the issues identified in the previous chapter about building correct yield curve event tree and will attain a satisfactory scenario generation method for this problem.

The rest of this chapter will describe the model. It starts by describing a simple three factor VAR1 model that is representing the underlying stochastic process. A nonlinear discretization model of the stochastic process is then suggested. The discretization model is general but it is currently based on the moment matching scenario generation method as defined in chapter 4. The next chapter will perform test and analysis of this model and will discuss the results of the different model configuration. (As well as argue why a simple 1–factor interest rate model such as the Vasicek model is not appropriate for stochastic programming applications and why the proposed 3–factor model provides more reliable solutions.)

6.1 A Vector Autoregressive Model of Interest Rates

A vector autoregressive mode with lag 1 (VAR1) may be defined as:

$$x_{t+1} = \mu + A(x_t - \mu) + \epsilon_{t+1}$$

where x_t is an $n \times n$ matrix, μ is an $n \times 1$ vector and $\epsilon_{t+1} \sim \mathcal{N}_n(\overline{0}, \Omega)$ and Ω is an $n \times n$ matrix. In this formulation of the VAR1 model, μ is interpreted as the long term drift. A and μ are deterministic parameters which need to be calibrated based on historical data.

The conditional mean and covariance for the error term ϵ_{t+1} are given as:

$$E[\epsilon_{t+1}|x_t] = 0$$
$$E[\epsilon_{t+1}\epsilon_{t'+1}|x_t] = \Omega$$

Given the state of an uncertain variable at time x_t, the purpose of the model is to predict the state of the variable at time $t+1$, x_{t+1}. Based on the findings of the previous section we define the vector x_t as the proxies for level, slope and curvature $(l_t, s_t, c_t)^T$ of the yield curves.

An example of the VAR1 model with three factors looks like:

$$\begin{aligned}
l_{t+1} &= \mu_l + a_{ll}(l_t - \mu_l) + a_{ls}(s_t - \mu_s) + a_{lc}(c_t - \mu_c) + \epsilon_{l,t+1} \\
s_{t+1} &= \mu_s + a_{sl}(l_t - \mu_l) + a_{ss}(s_t - \mu_s) + a_{sc}(c_t - \mu_c) + \epsilon_{s,t+1} \\
c_{t+1} &= \mu_c + a_{cl}(l_t - \mu_l) + a_{cs}(s_t - \mu_s) + a_{cc}(c_t - \mu_c) + \epsilon_{c,t+1}
\end{aligned}$$

To estimate the parameters of the VAR1 model (μ, A, Ω) we can use the parameter estimation for a general linear regression model of the form:

$$y_i = \alpha + \beta x_i + \epsilon_i, \qquad \text{for all } i = 1, \cdots, n$$

Or in a matrix form:

$$\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

This can be rewritten as:

$$Y = \overline{X}\delta + \varepsilon$$

The VAR1 model can be rewritten in this form. Now we may use standard least square estimators as follows:

$$\hat{\delta} = (\overline{X}^T\overline{X})^{-1}\overline{X}^T Y$$

which minimizes the sum of least squares in the expression $||Y - \overline{X}\delta||^2$.

The estimator for the residuals (ε) is given as:

$$res = Y - \overline{X}\hat{\delta}$$
$$\hat{\Omega} = res^T res/(n-1)$$

The estimator $\hat{\delta}$ is then decomposed into μ and A from the VAR1 model and the estimator $\hat{\Omega}$ can be directly used as the estimator for Ω in the VAR1 model.

The VAR1 model so far may only be used for one–period predictions (same interval length as in the historical time series). But it may easily be extended to predict k periods ahead:

$$x_{t+k} = \mu + A^k(x_t - \mu) + \epsilon_{t+k}$$

where $\epsilon_{t+k} \sim \mathcal{N}_n(\overline{0}, \sum_{i=1}^{k} A^{i-1}\Omega(A^{i-1})^T)$

The reasons for choosing a VAR1 model as the underlying model of interest rate uncertainty are the following:

1. One can chose any factors or any number of factors to describe the variability. This gives us maximum flexibility with respect to our observations from a factor analysis of interest rates.
2. Time step flexibility. Varying time steps can be easily implemented.
3. Mean reversion is built into the VAR1 model.

The VAR1 model is discrete in time but continuous in states, so in order to use the model as a scenario generator for stochastic programs we need to discretize it in states as well. This can be done using a moment matching model (as described in chapter 4). The yield curve scenario discretization model is described in the next section.

6.2 Scenario Generation and Event Tree Construction

In Dynamic Stochastic Programming (DSP) literature for fixed income securities, simple models of interest rates often are used to represent the underlying interest rate uncertainty. In several applications lattice structures are either blown up into unique paths or sampled from, so as to account for the path dependency of DSP problems. See for example, Zenios at [14], and Rasmussen and Clausen at [11]. One immediate problem with such approaches is that the uncertainty space is not covered as efficiently as possible. This is mainly due to the recombining structure of the original trees.

Others (such as Nielsen and Poulsen at [10]) have used continuous interest rate models. Such models are either continuous both in time and state, or discrete in time and continuous in states. Discretizing in time is normally straight forward; it is a question of reformulating a differential equation into a difference equation. Discretizing in state, however, is often a more challenging issue. A number of nodes (in our case including yield curve information) have to be generated for each time point to give a discrete representation of the continuous distribution. There is no general consensus as to the best way of doing this discretization. According to one trend of research, the main focus is on generating discrete distributions which mimic the underlying continuous distribution as closely as possible. This is either done by sampling (as described in more detail in chapter 3), or moment matching approaches (as described in detail in chapter 4). Another trend of research states that the aim is not necessarily to get the closest discrete representation of the continuous distribution, but rather to find a discrete representation which results in a closer approximation to the "true" optimal solution of the stochastic program in question. Here the "true" optimal solution refers to the solution we will get if we were able to solve the stochastic program using the underlying continuous process directly. Indeed if we were able to do that, there would be no need to discretize the process in the first place. Nevertheless, it can be shown (see Pflug at [19]) that in general if the discrete process has the smallest distance (using the transport metric) to the underlying continuous process, then the SP solutions found will be guaranteed to be within certain bounds of the "true" SP solutions. (see also Pflug and Hochreiter at [47]). Although theoretically appealing, the guaranteed bounds are in many cases too large to have any practical interest (see Wallace and Kaut at [8]). Comparison and further development of specialized models and solution algorithms for these two streams of scenario discretization approaches are the subject of future research.

The following section will propose a yield curve scenario generation model which abides by accuracy, consistency and correctness. That is done by matching the moments and covariance using moment matching (for complying with accuracy and consistency). The solid financial background behind the model leads for correctness of the generated scenarios.

6.3 The Complete Model

6.3.1 Description of Model Data

We define the following sets, parameters and variables:

- f: Set of factors (level, slope and curvature), f' is alias for f.
- i: Set of zero coupon bonds (zcb's).
- i': A subset of the set i corresponding to the zcb–rates which define the three factors. We have chosen i' to be the set of 1, 5 and 30 year.
- t: Set of time points.
- s: Set of scenarios.
- $Mean_f$: Mean value for factor f. This value comes from the VAR1 model.
- $Covar_{f,f'}$: The covariance matrix of the error term taken from the VAR1 model.
- $Skewness_f$: Skewness of factor f. Assumed to be zero based on the normality assumption of the VAR1 model.
- τ_i^t: Time to maturity for zcb i at time t.
- $Y_{s,i}^{(VAR1)}$: The variable calculating the yields for scenario s and zero coupon bond i for the VAR1 model.
- $NSY_{i',s}$: The value of the Nelson-Siegel yields for scenario s and zero coupon bond i'.
- $R_{s,i}$: The interest rate of scenario s and zero coupon bond i.
- $CP_{i,s}^{\text{child}}$ is the child price at scenario s corresponding to zero coupon bond i.
- PP_i^{parent} is the parent price corresponding to zero coupon bond i.

– ϕ_i^{Const} is the positive vector with index i ensuring arbitrage–free pricing of all zero coupon bonds.

6.3.2 The Model

Note that the scenario generation model here is designed for single period. The model can be extended to a multi–period model with some minor changes.

$$\begin{aligned}\text{Minimize } & \sum_f (E(x)_f - Mean_f)^2 + \sum_f \sum_{f'} (\sigma(x)_{f,f'} - Covar_{f,f'})^2 + \\ & \sum_f (E3(x)_f - Skewness_f)^2 + \sum_s \sum_{i'} (Y_{i',s}^{(VAR1)} - NSY_{i',s})^2 \qquad (6.1)\end{aligned}$$

$$E(x)_f = \sum_s p_s x_{s,f} \qquad \text{for all } f \qquad (6.2)$$

$$\sigma(x)_{f,f'} = \sum_s p_s (x_{s,f} - E(x)_f)(x_{s,f'} - E(x)_{f'}) \qquad \text{for all } f, f' \qquad (6.3)$$

$$E3(x)_f = \frac{\sum_s (x_{s,f} - E(x)_f)^3}{(\sum_s (x_{s,f} - E(x)_f)^2)^{3/2}} \qquad \text{for all } f \qquad (6.4)$$

$$NSY_{i',s} = \varphi_{s,0} + \varphi_{s,1} e^{-\varphi_{s,3}\tau_{i'}^{\text{parent}}} + \varphi_{s,2}\tau_{i'}^{\text{parent}} e^{-\varphi_{s,3}\tau_{i'}} \qquad \text{for all } s, i' \qquad (6.5)$$

$$R_{s,i} = \varphi_{s,0} + \varphi_{s,1} e^{-\varphi_{s,3}\tau_i} + \varphi_{s,2}\tau_i^{\text{parent}} e^{-\varphi_{s,3}\tau_i^{\text{parent}}} \qquad \text{for all } s, i \qquad (6.6)$$

$$Y_{s,1}^{(VAR1)} = x_{s,1} \qquad \text{for all } s \tag{6.7}$$

$$Y_{s,30}^{(VAR1)} = x_{s,2} + Y_{s,1}^{(VAR1)} \qquad \text{for all } s \tag{6.8}$$

$$Y_{s,5}^{(VAR1)} = \frac{4}{29}Y_{s,30}^{(VAR1)} + \frac{25}{29}Y_{s,1}^{(VAR1)} + x_{s,3} \qquad \text{for all } s \tag{6.9}$$

$$R_{s,i} \geq 0.005 \qquad \text{for all } s, i \tag{6.10}$$

$$PP_i^{\text{parent}} = e^{-r_i \tau_i^{\text{parent}}} \qquad \text{for all } i \tag{6.11}$$

$$CP_{i,s}^{\text{child}} = e^{-R_{i,s} \tau_i^{\text{child}}} \qquad \text{for all } s, i \tag{6.12}$$

$$PP_i^{\text{parent}} = \sum_s \phi_i^{\text{Const}} CP_{i,s}^{\text{child}} \qquad \text{for all } i \tag{6.13}$$

$$\phi_i^{\text{Const}} \geq 0 \qquad \text{for all } i \tag{6.14}$$

The complete model includes the following points:

- Matching the moment of the vector autoregressive model that is described at section 6.1 for a possible future time point (which depends on the value of k).
- The VAR1 model is matching the realizations of the vector of factors - slope, level and curvature.
- The objective function at equation 6.1 matches the first three moments of the VAR1 model, as well as matches the Nelson-Siegel smoothing of the yield curve.

- Equations 6.2, 6.3, 6.4 matches the mean, variance and skewness of the VAR1 model. (From computational behaviour it might be considered a good idea to keep all these three equations as part of the objective function rather than as constraints, however, from the modularity of the model it is described in this way.).
- Equations 6.11, 6.12 and 6.13, deal with the arbitrage removal process.
- The rest of the equations transforms the factors back into the yield curve, defines the values for the Nelson-Siegel smoothing and calculates the rates of the yield curve.

6.4 Difficulties in Solving the One Period Model

Even though, intuitively solving the model described in previous section as a one stage model can lead for a better control over the solution since the optimal solution would balance an arbitrage free solution, with smoothing and keeping the correct moments of the outcome scenarios, the high complexity of the non-linear model (quadric objective function with non-linear constrains) leads for an undesired outcome of the optimization model.

Therefore, the model was broken down to an iterative procedure as described below:

1. Calculate the error term suggested by the VAR1 model at 6.1 using moment matching (match the first three or four moments plus covariance).
2. Calculate the complete VAR1 model with the error term.
3. Create the yield curve by a smoothing method (Nelson-Siegel or an affine smoothing method).
4. Remove Arbitrage

Of course the iterative model can no longer ensure that the smoothing and the arbitrage removal process does not effect the optimality of the moment matching used as part of the VAR1 model calculation. On the other hand, a one stage complex optimization model might lead to worse results when a local optimal solution is found. A comparative study of the two models suggests a very interesting mathematical and practical study.

6.5 Variations of The Model

The basic model and the iterative approach that are described in previous sections are based on concrete implementation that can be changed or extended. This section provides a short summary about the possible variations using this implementation in order to extend or to test the model:

- Moment matching was implemented using the algorithm suggested by Højland and Wallace [4]. Moment matching could also be done differently using the algorithm suggested by Højland, Kaut and Wallace at [6], as was carried out and tested in the next chapter. Other approaches for scenario generation can also be used as an alternative for moment matching. (Subject for further research.)
- The smoothing method can be changed. The affine smoothing and Nelson Siegel were tested in this work. Other smoothing approaches can also be implemented and tested.
- Instead of using an arbitrage removal process a non–arbitrage condition can be added to the scenario generation method as shown in section 5.2.5.
- The definitions of the factors as a linear combination of the rates in order to calculate the level, slope and curvature can be done differently. Testing different linear combination for finding these factors is a subject for future work.

6.6 Summary

A VAR1 model was suggested to perform scenario generation of the term structure. The scenario generation problem suggested was hard to solve, and that led for an iterative approach towards scenario generation. A different variations of the scenario generation were tested and the results will be shown in the next chapter.

Chapter 7

Fundamental Analysis of Results

The VAR1 model from the previous chapter was tested in order to uncover its capabilities and assess its qualities. The basic configuration was tested over a period of two years. The first results were computed based on the period up to August 2005 and the second test was done for the period up to May 2007. Moreover, the problem was tested with and without arbitrage removal as well as using an affine smoothing in comparison to smoothing introduced by Nelson-Siegel. The problem was also tested by comparing different moment matching approaches from chapter 4.

The basic configuration used in this chapter is the VAR1 model from the previous chapter with affine smoothing, arbitrage removal and moment matching approach by Højland & Wallace [4] observing the first three moments as well as the covariance matrix.

For the multi–period case, the 3–factor scenario generation approach was tested on different types of trees and was also compared to the 1–factor Vasicek model for scenario generation.

7.1 Looking at Different Amount of Scenarios

The following graphs present the results obtained when running a one period yield curve scenario generation. The moment matching is based on Højland and Wallace as described in section 4.2, affine smoothing before and after the arbitrage removal process. Figures 7.1, 7.2, 7.3 and 7.4 presented the 4, 8, 16 and 32 scenarios trees that were generated respectively.

In all the figures presented in this chapter:

- The black square represents the drift, which is the mean reversion factor of the model.
- The green circle represents the median point of all the scenarios generated.

The following observations where made when exploring these results:

- The growth in the number of scenarios definitely leads to a more complex tree that might allow a more thorough risk assessment of the optimization problem.
- Observe, for example, that the one year returns using 32 scenarios the range is from 0.008 to 0.044, while for 4 scenarios it ranges between 0.012 and 0.036. As expected, this indicates that the larger the number of scenarios being used, the better the risk management capabilities of the mode. What is more, the interest range for this period was within the results achieved by both the 32 and the 4 scenario forecast.
- It is also interesting to observe that the scenarios achieved by the 4 scenario model for both 1 and 6 year rates are almost symmetric to the median.
- It has also been observed when comparing the results before and after arbitrage removal that the arbitrage removal process does not effect the structure of the trees very much. That in return, gives us a good indication that the scenarios after the arbitrage removal process are in a reasonable distance from the original scenarios which represent the correct moments and correlation.

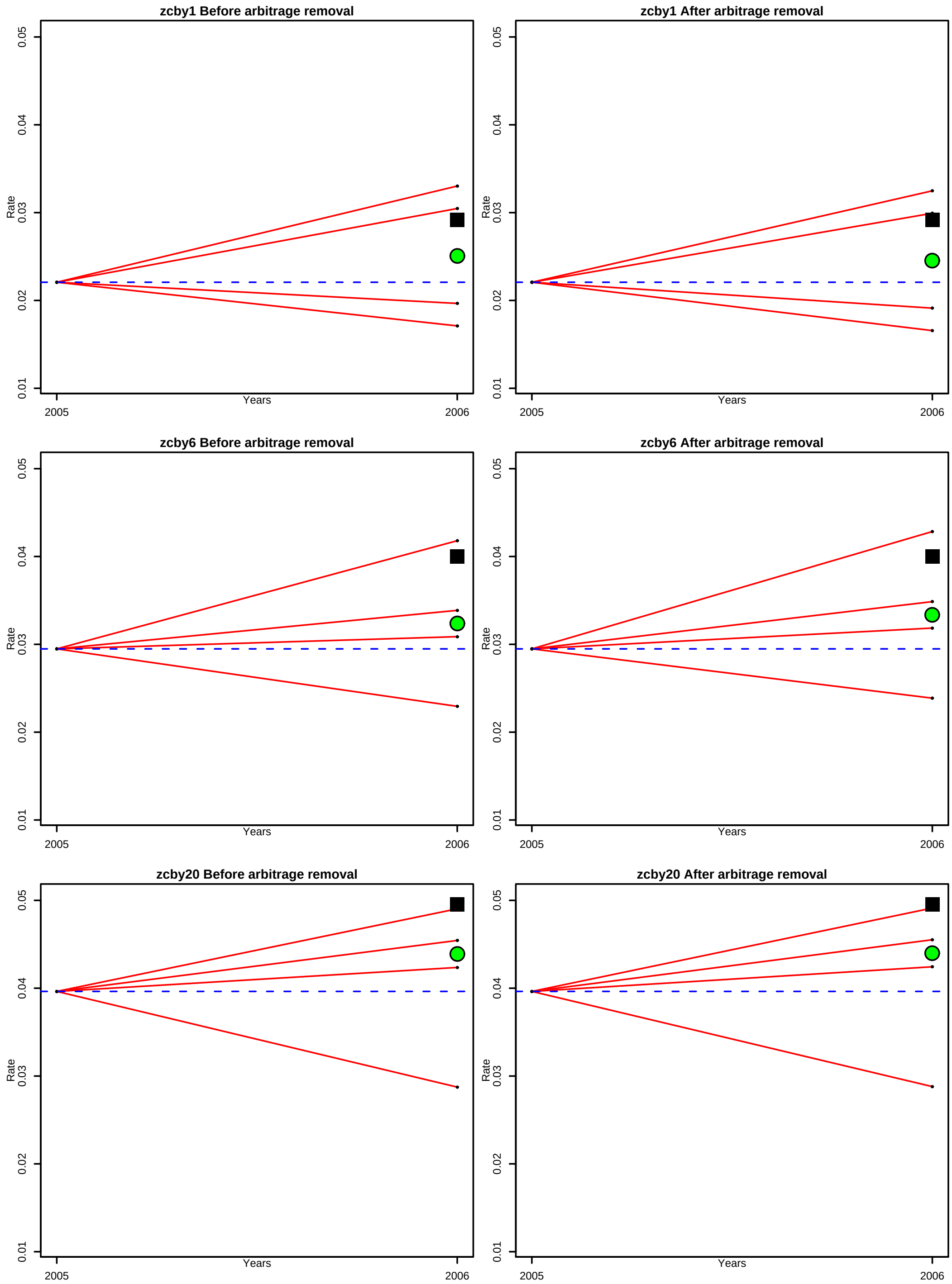


Figure 7.1: 4 Scenarios to Represent the Yields Curve of the 1, 6 and 20 Year Rates Before and After Arbitrage Removal. (Moment Matching is Based on 4.2 and Affine Smoothing is Used on Data Up To August 2005.)

7.2 Future Forecasting

The same forecasting was done for results until May 2007. The results are shown at figures 7.5, 7.6, 7.7 and 7.8 presented the 4, 8, 16 and 32 scenarios trees that were generated respectively.

As can be seen, the observation of the previous period still holds for this period. This indicates that the purposed approach reflects consistent results. It should be stressed again that allegedly 16 scenarios are needed in order to receive high quality.

7.3 Comparing Scenario Generation Approaches

A similar configuration to the basic configuration of this chapter can be created using the heuristic suggested by Højland, Kaut and Wallace at [5] (also explained in section 4.3) in which the first four moments where matched together with the correlation matrix. Figure 7.9 presents the results obtained when running a one period yield curve scenario generation with an affine smoothing, one period model having 16 scenarios.

The following observations where made when exploring these results:

- The results resemble a normal distribution more than the results achieved by the scenario generation used so far.
- The volatility of the long term interest rate is high in comparison with the short term interest rate. This creates a problem with the correctness of the scenario generation method as one would expect the volatility of the long term interest rate to be lower as observed in real life.
- The heuristic has been run several times in order to examine the difference in the results achieved. Figure 7.10, represents a run of the algorithm where the volatility

of the long term interest rate is lower by comparison to the short term interest rate. However, there is one extreme scenario for the 20-year rate. If we ignore that extreme scenario, the results still resemble a more normal distribution than the results achieved using the moment matching approach suggested by Højland and Wallace.

– When the data of August 2005 was used in figure 7.11, the results led to the understanding that an in depth stability analysis of this method should be done (together with the mortgagor problem, for example) in order to examine the stability of the solutions generated by these moment matching approaches. When the data of August 2005 was used in figure 7.11, the results led to the understanding that an in depth stability analysis of this method should be done (together with the mortgagor problem, for example) in order to examine the stability of the solutions generated by these moment matching approaches. Figure 7.11 presents a very unrealistic situation. In the observed figure, the 1-year rate volatility is lower than the one examined in the 6-year rate which is lower than the one in the 20-year rate. This outcome is not acceptable as far as the correctness criterion is concerned. The results of this figure seem to be very unambiguous. As opposed to what is actually happening in financial research, the results displayed in this figure maintain that for a longer maturity interest rate there is higher volatility.

As can be surmised in this section, using a different moment matching methodology as part of the VAR1 model seems to have the greatest influence on the observed results. Therefore, an in depth sensitivity analysis of the different moment matching methodology is suggested in order to better qualify the effectiveness of this approach.

7.4 Comparing Affine Smoothing with Nelson-Siegel

The following figures present the results obtained when running a one period yield curve scenario generation. The moment matching is based on Højland and Wallace as described in section 4.2 and Nelson-Siegel smoothing before and after the arbitrage removal process. Figures 7.12, 7.13, 7.14 and 7.15 present the 4, 8, 16 and 32 scenarios trees generated respectively.

The following observations where made when exploring these results:

- For 4 scenarios the difference between the affine smoothing at 7.1 and the Nelson-Siegel smoothing as presented at 7.12 is mainly on the 1-year rate in which the results of the affine model appear a bit more diversified.
- For 8 scenarios, the results are very similar.
- For 16 scenarios the 1-period results using the affine model create a more diversified scenario generation.
- For 32 scenarios the results are quite similar.
- Except for the 1-year rate for the 4 and 16 scenarios (in which the affine scenarios look a little bit more diversified) the results are quite similar. Therefore, it is concluded that the effect of using different smoothing strategy is not vital for the results received. It gives the impression that the results achieved by the affine model are slightly better in the sense that it is somewhat more diversified and can be better for risk management.

Therefore, the choice of different smoothing algorithm is of low priority since the results show that it does not have a lot of effect on the outcome. The results for the period of May 2007 are similar and can be found at Appendix 1.

7.5 Comparing Different Multi-Stage Scenario Generation Approaches – the Vasicek and the VAR1 Models

Three different tree structures were compared to generate scenarios for the period May 2007 until May 2012. The Vasicek model keeps the tree structure of 3–3–3–3–3, which is branched yearly at five time points. Whereas the VAR1 model contains 4 periods. This model has only four future time points which are 2008, 2009, 2010 and 2012. This is because it was decided to keep a similar number of total scenarios in order to have more comparative results. The VAR1 model was implemented with two different tree structures. One is symmetric a 4–4–4–4 tree and the other one is a left weighted tree of the 16–4–2–2 structure.

As observed by Dempster at [68], the recommended tree structure is likely to be a heavy left tree, i.e. trees that contains many more branching point at the early time point rather than at later stages. (For example, a tree structure of 32–4–2–2, where more branching is done on the first stage (32) in comparison to the last stage of 2 child nodes from each parent node). That is an intuitive result when one is taking into account the fact that a scenario generation might have stronger capturing capabilities for the first stages where the stochasticity is still within considerable range, since all the future scenarios in stochastic programming are made in order to find the right decision today. This also presents a telescopic view reflecting higher importance of decisions made earlier in the decision tree when used later in the optimization problem.

The results of the comparisons for 1-year rate, 6-year rate and 20-year rate can be seen in figures 7.16, 7.17, 7.18 respectively.

The following observations where made when exploring these results:

- Observing the one year rate from figure 7.16, both configurations of the VAR1 model suggest higher volatility in the rates than the one presented by the Vasicek model and are believed to more correctly predict the observed changes in interest rates. Moreover, it can also be seen that the left weighted tree does not predict very low interest rates (lower than 2.5%). That is in accordance to the observed behaviour in the Danish market. That in comparison to the 4–4–4–4 tree which suggests a few scenarios where the level of the interest rate is almost zero. For this period it is concluded that the VAR1 left weighted tree produces the best presentation followed by the 4–4–4–4 VAR1 tree.

– Observing the six and the twenty year rate from figures 7.17 and 7.18 the volatility of the Vasicek model is by no means acceptable and is much too low to be used for appropriate risk management. Both VAR1 approaches produce more valid scenarios. The 4–4–4–4 VAR1 model produces too high volatilities for the middle and long term interest rates while keeping a few scenarios with a predicted interest rate of almost 0%. That result is not optimal from the perspective of the correctness criteria, but it is still much more useful than the one presented by the Vasicek model. The left weighted tree suggests a more reasonable volatility and still does not expect the interest rate to go below 2.5%. That in return applies much better correctness. However, it seems to suffer from some gaps, i.e. a few intervals in which no interest rate scenarios are observed which is very surprising. This result might indicate that a different scenario generation approach might lead to better scenarios. Still the left weighted scenario tree presented the most useful results.

It should also be mentioned that as shown by Rasmussen & Poulsen [39] when comparing the 1–factor interest rate model such as the Vasicek model to the proposed 3–factor VAR1 model with the observed yield curve, a clear dominant indication exists leaning towards the VAR1 model.

7.6 Summary

This chapter examines several configurations of the VAR1 model. A study was done to test the different components that assimilate these processes – changing the scenario generation method, the smoothing method, testing over different time points and with and without the arbitrage removal process. This study conducted a comprehensive comparison between two configurations of the VAR1 model and the Vasicek model.

There are still, of course, numerous places in this method that can be improved. However, the overall picture generated by this VAR1 model is promising.

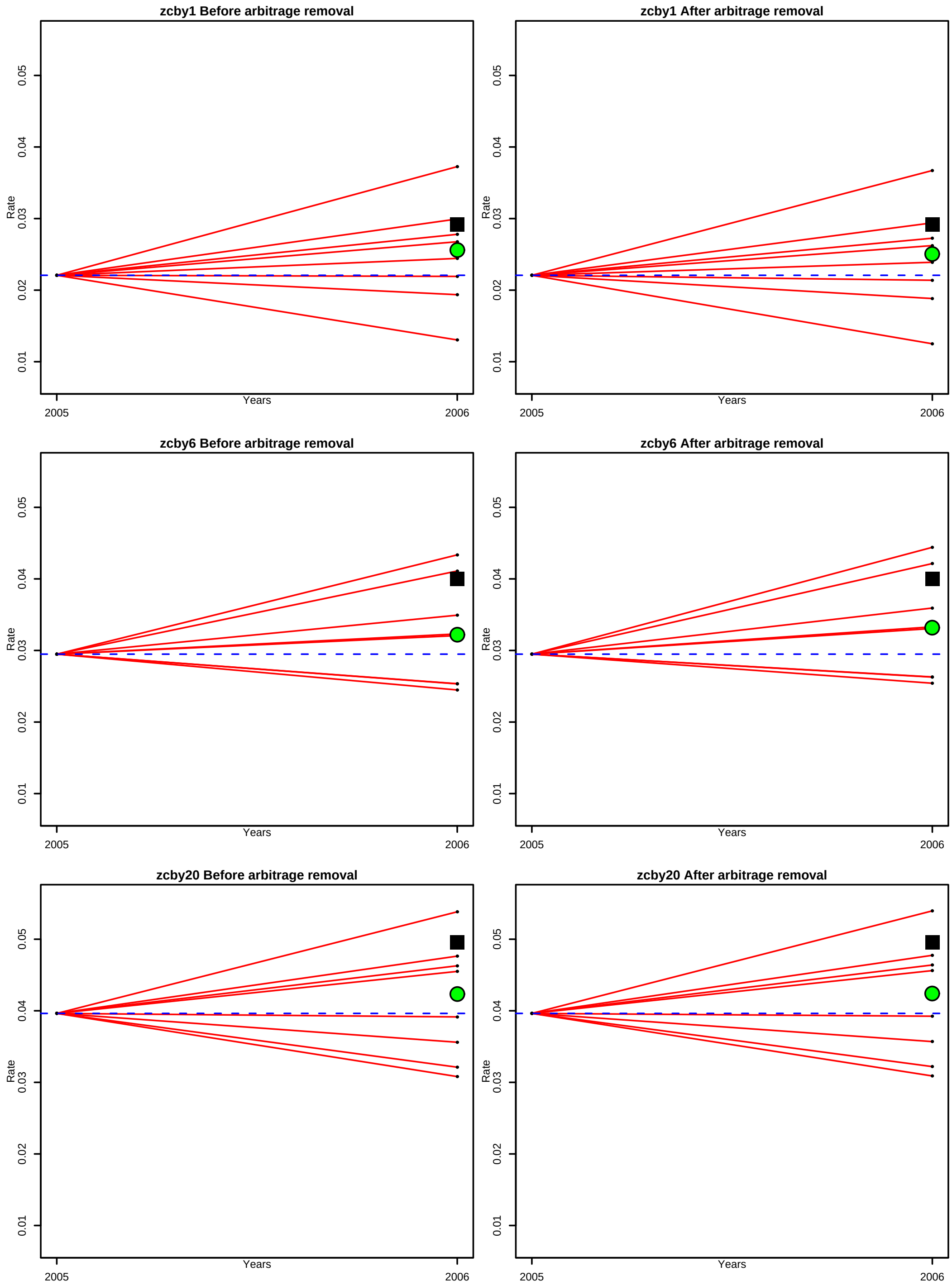


Figure 7.2: 8 Scenarios to Represent the Yields Curve of the 1, 6 and 20 Year Rates Before and After Arbitrage Removal. (Moment Matching is Based on 4.2 and Affine Smoothing is Used on Data Up To August 2005.)

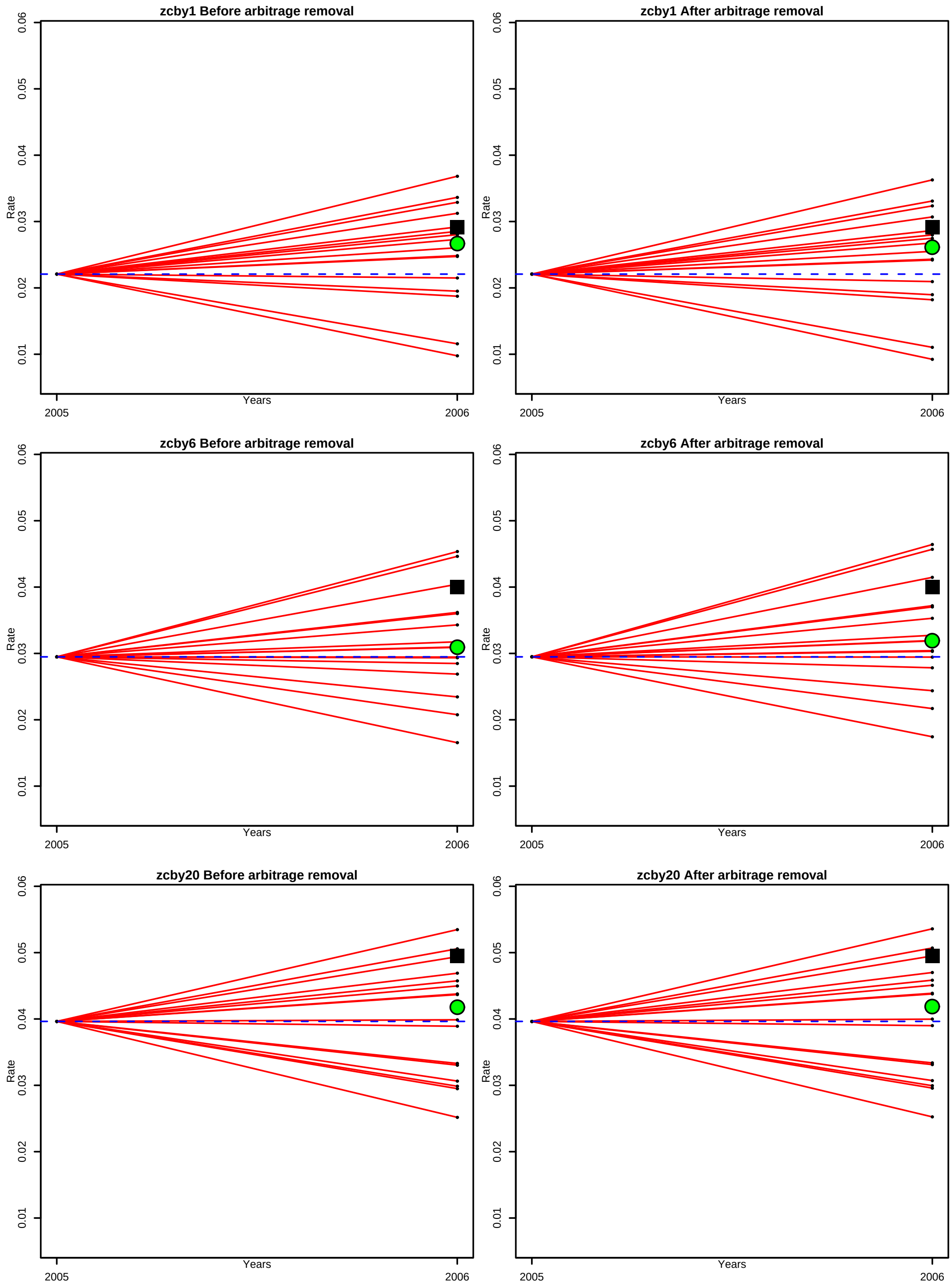


Figure 7.3: 16 Scenarios to Represent the Yields Curve of the 1, 6 and 20 Year Rates Before and After Arbitrage Removal. (Moment Matching is Based on 4.2 and Affine Smoothing is Used on Data Up To August 2005.)

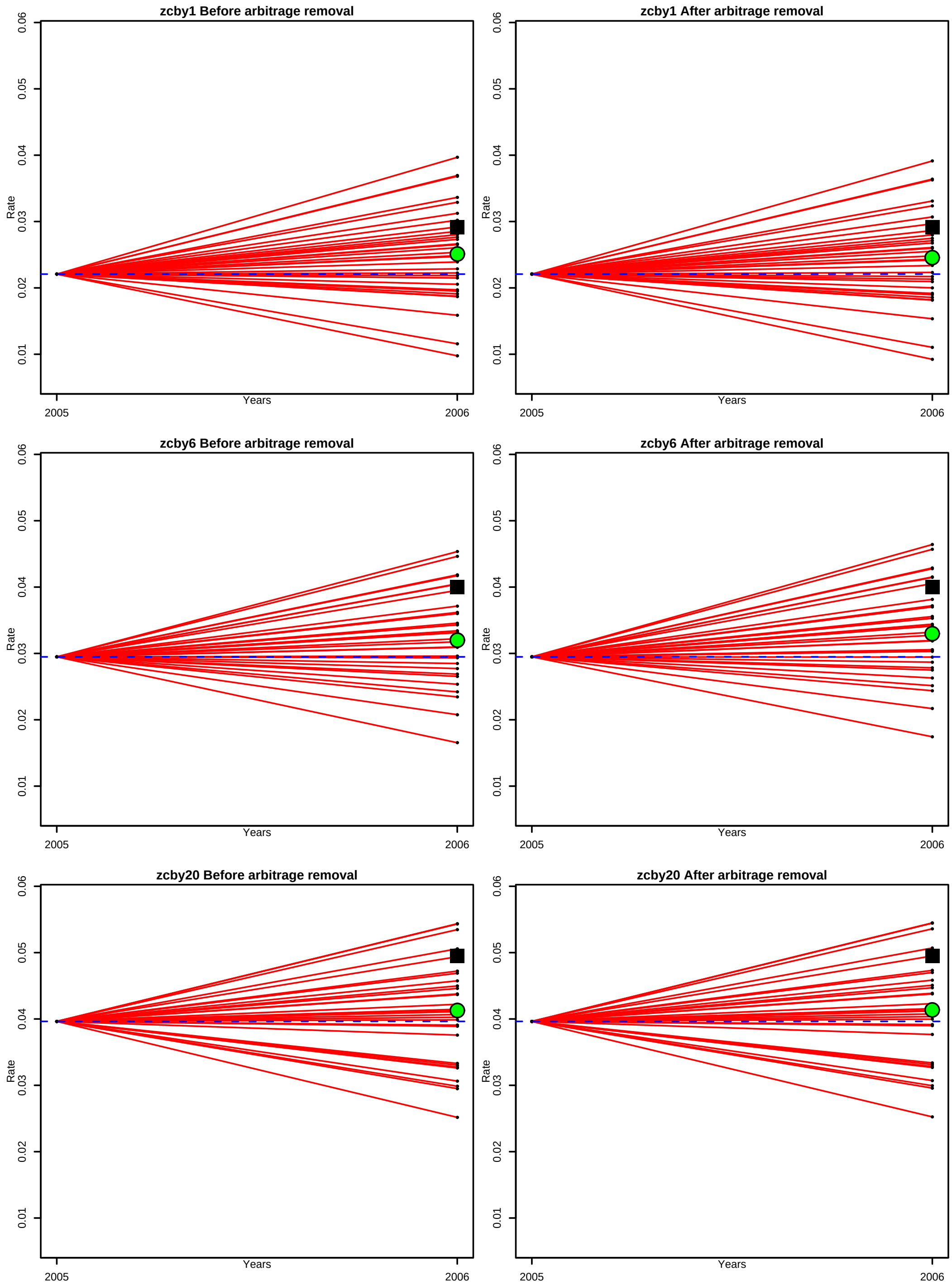


Figure 7.4: 32 Scenarios to Represent the Yields Curve of the 1, 6 and 20 Year Rates Before and After Arbitrage Removal. (Moment Matching is Based on 4.2 and Affine Smoothing is Used on Data Up To August 2005.)

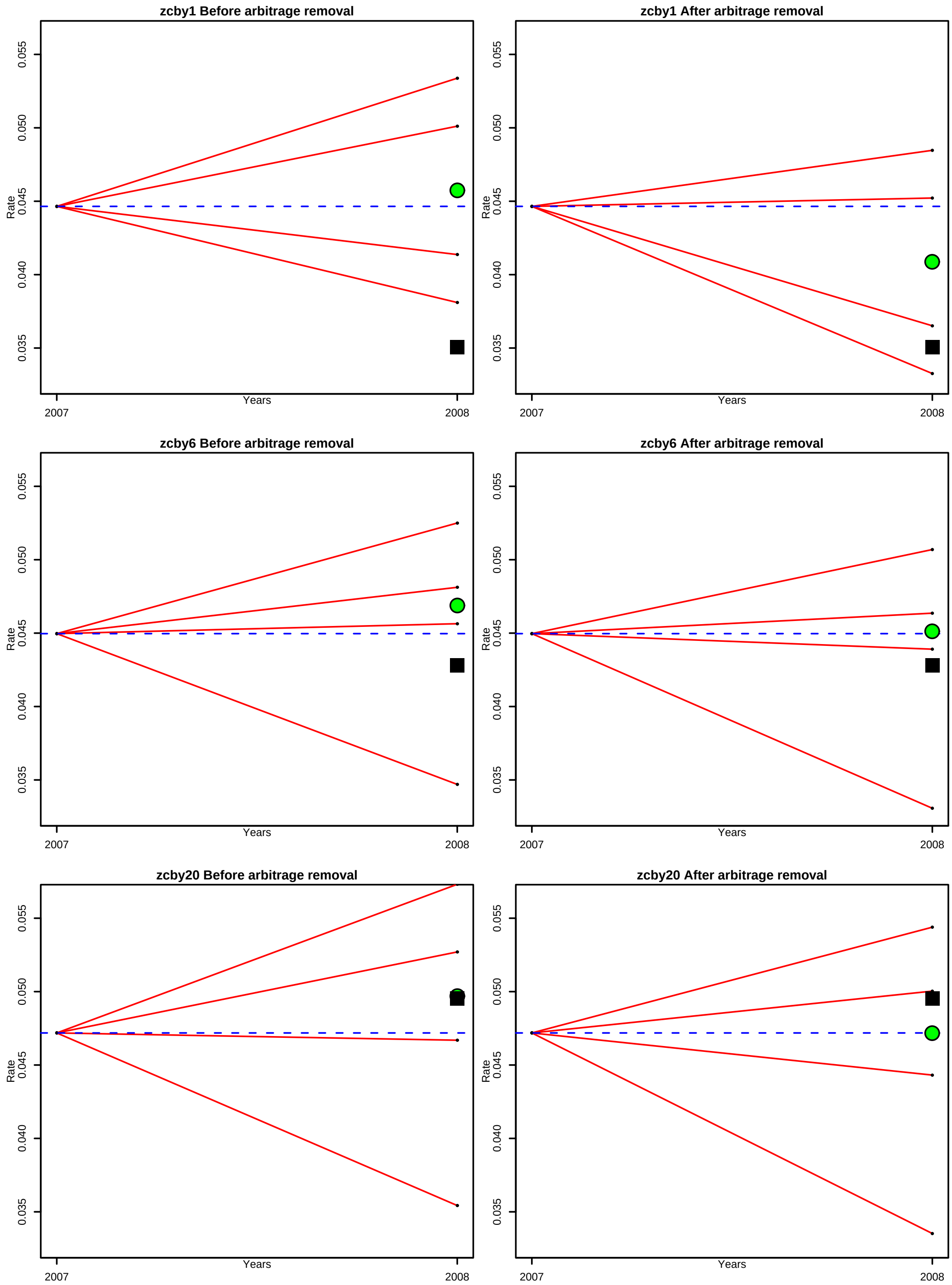


Figure 7.5: 4 Scenarios to Represent the Yields Curve of the 1, 6 and 20 Year Rates Before and After Arbitrage Removal. (Moment Matching is Based on 4.2 and Affine Smoothing is Used on Data Up To May 2007.)

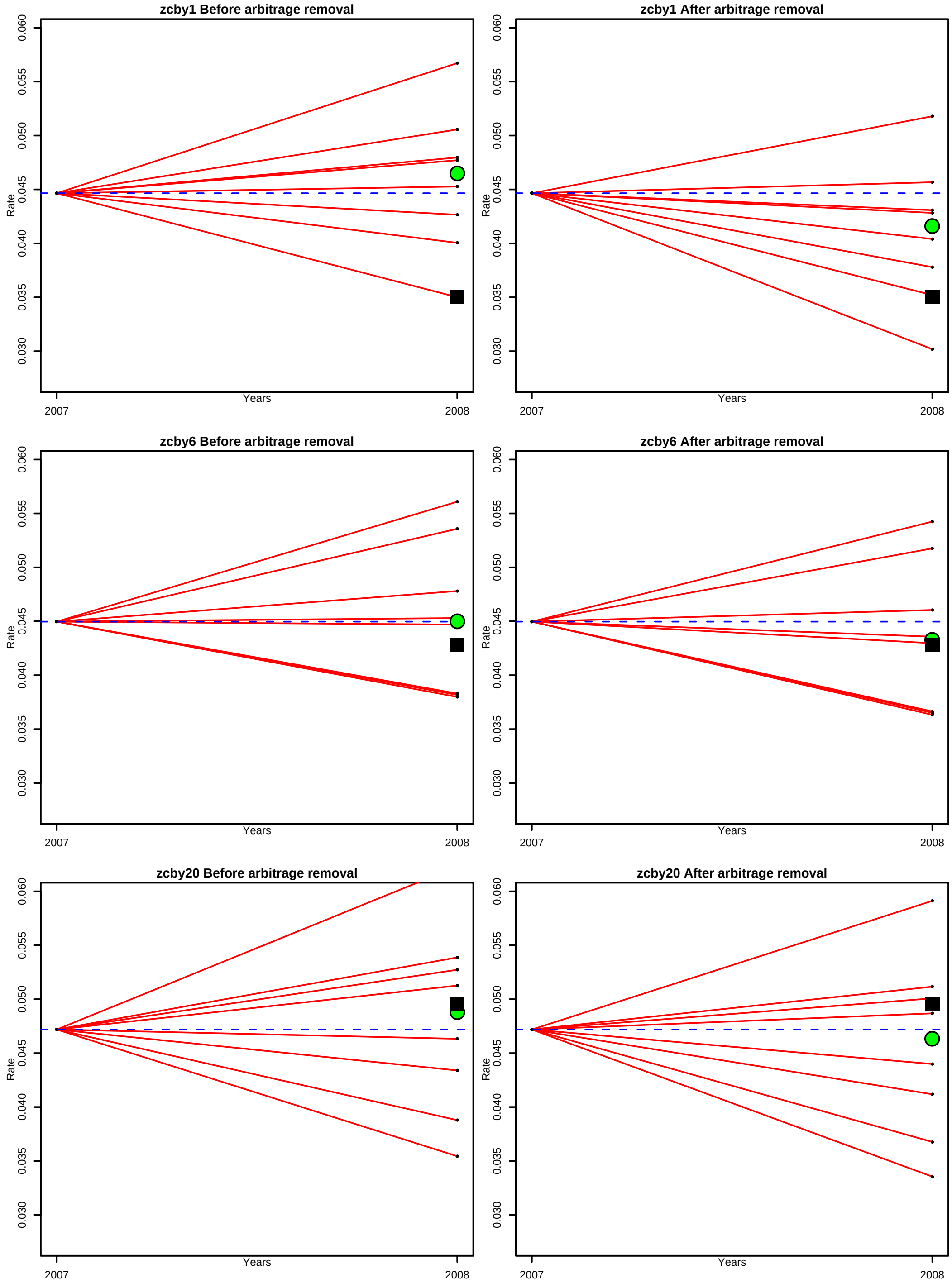


Figure 7.6: 8 Scenarios to Represent the Yields Curve of the 1, 6 and 20 Year Rates Before and After Arbitrage Removal. (Moment Matching is Based on 4.2 and Affine Smoothing is Used on Data Up To May 2007.)

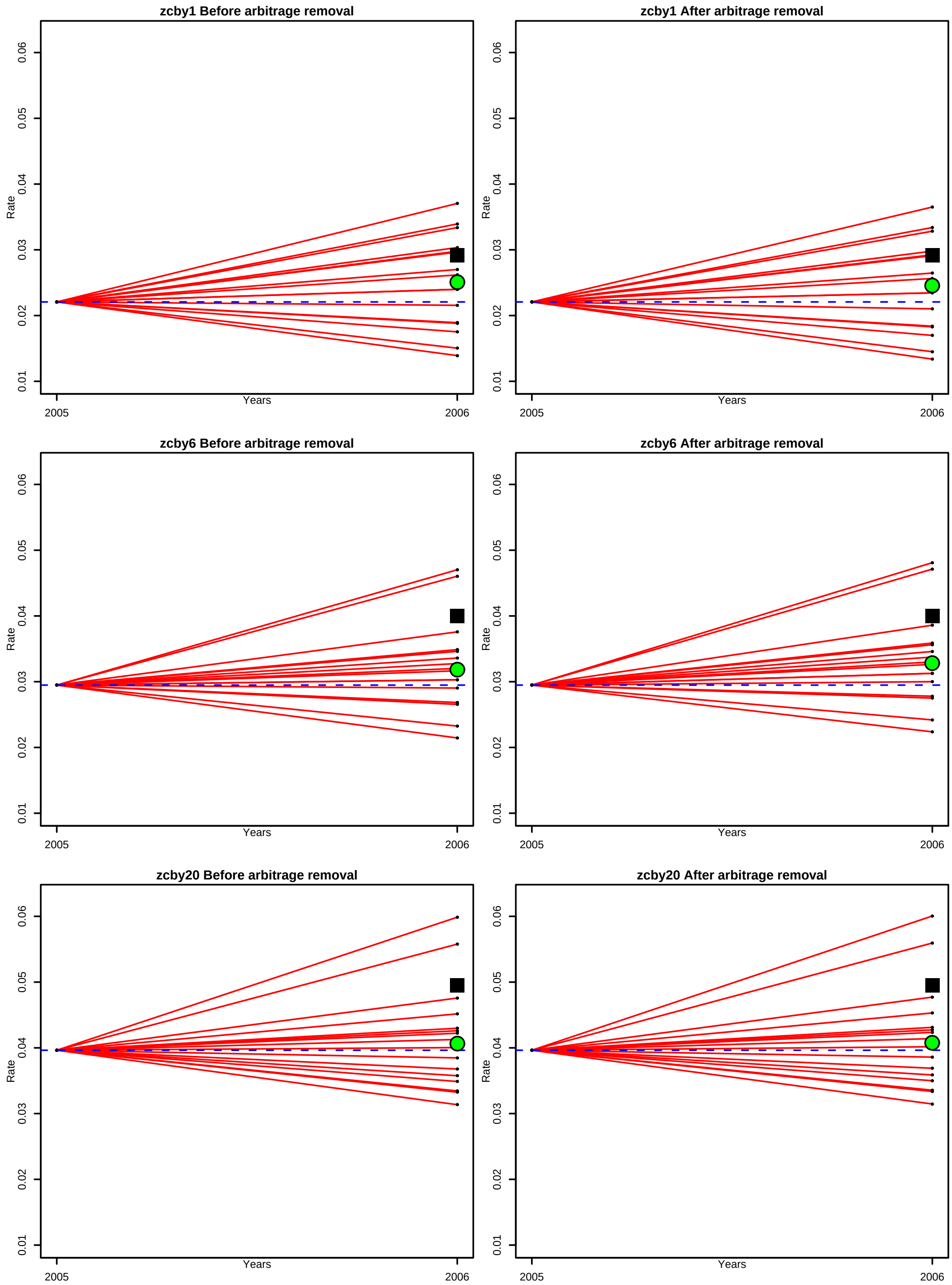


Figure 7.7: 16 Scenarios to Represent the Yields Curve of the 1, 6 and 20 Year Rates Before and After Arbitrage Removal. (Moment Matching is Based on 4.2 and Affine Smoothing is Used on Data Up To May 2007.)

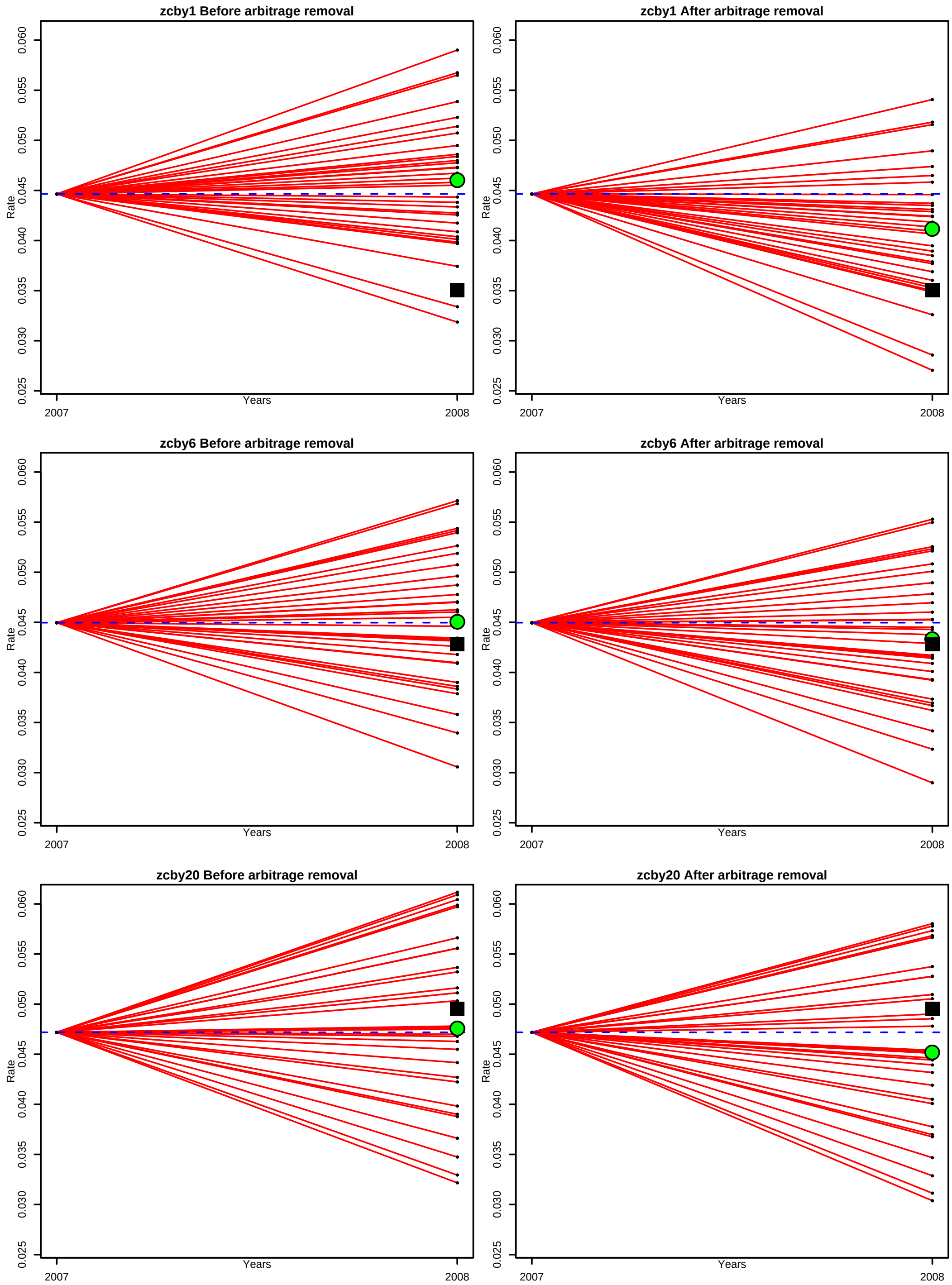


Figure 7.8: 32 Scenarios to Represent the Yields Curve of the 1, 6 and 20 Year Rates Before and After Arbitrage Removal. (Moment Matching is Based on 4.2 and Affine Smoothing is Used on Data Up To May 2007.)

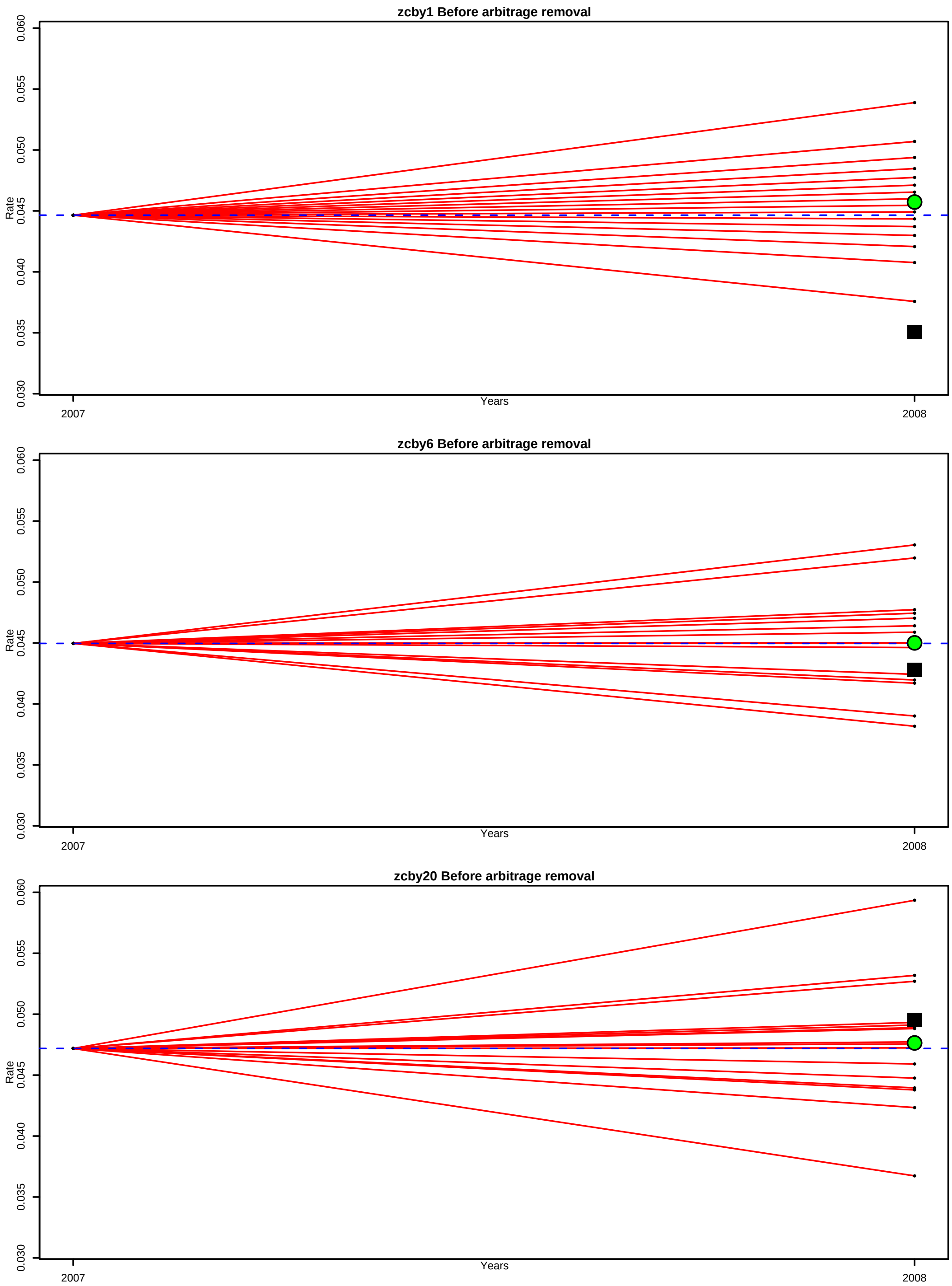


Figure 7.9: 16 Scenarios to Represent the Yields Curve of the 1, 6 and 20 Year Rates Before Arbitrage Removal. (Moment Matching is Based on 4.3 and Affine Smoothing is Used on Data Up To May 2007.)

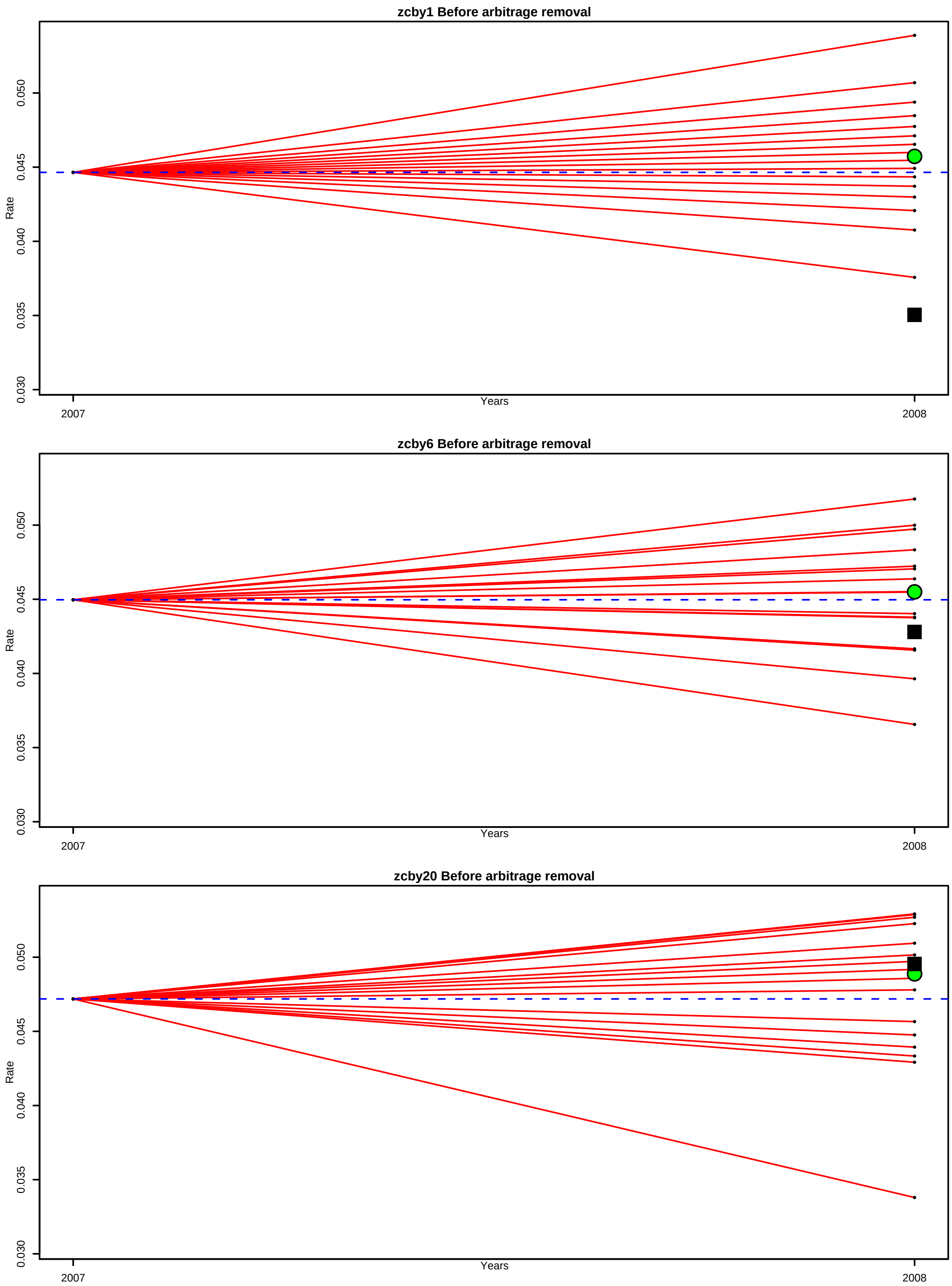


Figure 7.10: 16 Scenarios to Represent the Yields Curve of the 1, 6 and 20 Year Rates Before Arbitrage Removal. (Moment Matching is Based on 4.3 and Affine Smoothing is Used on Data up To May 2007.)

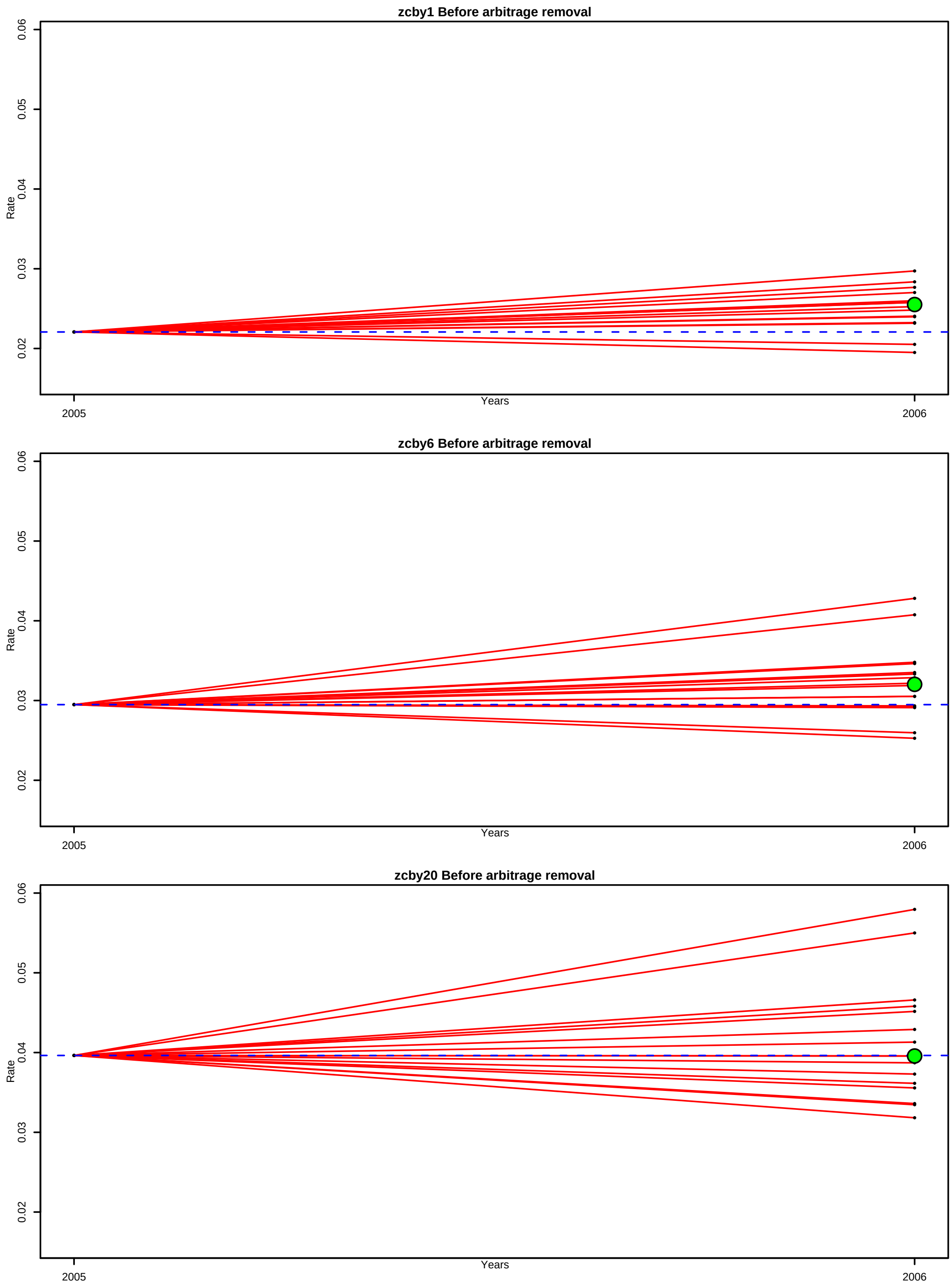


Figure 7.11: 16 Scenarios to Represent the Yields Curve of the 1, 6 and 20 Year Rates Before Arbitrage Removal. (Moment Matching is Based on 4.3 and Affine Smoothing is Used on Data Up To August 2005.)

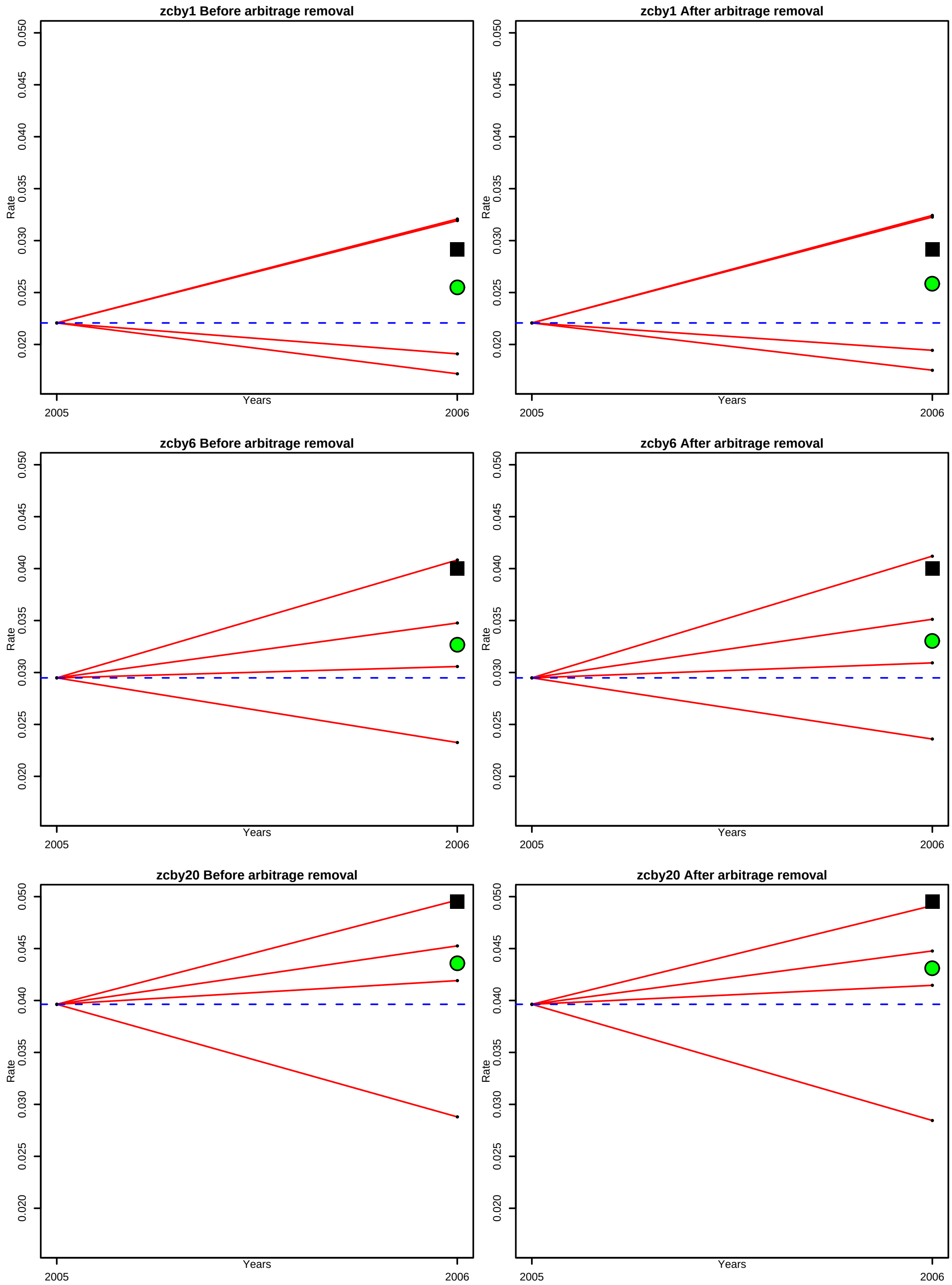


Figure 7.12: 4 Scenarios to Represent the Yields Curve of the 1, 6 and 20 Year Rates Before and After Arbitrage Removal. (Moment Matching is Based on 4.2 and Nelson-Siegel Smoothing is Used on Data Up To August 2005.)

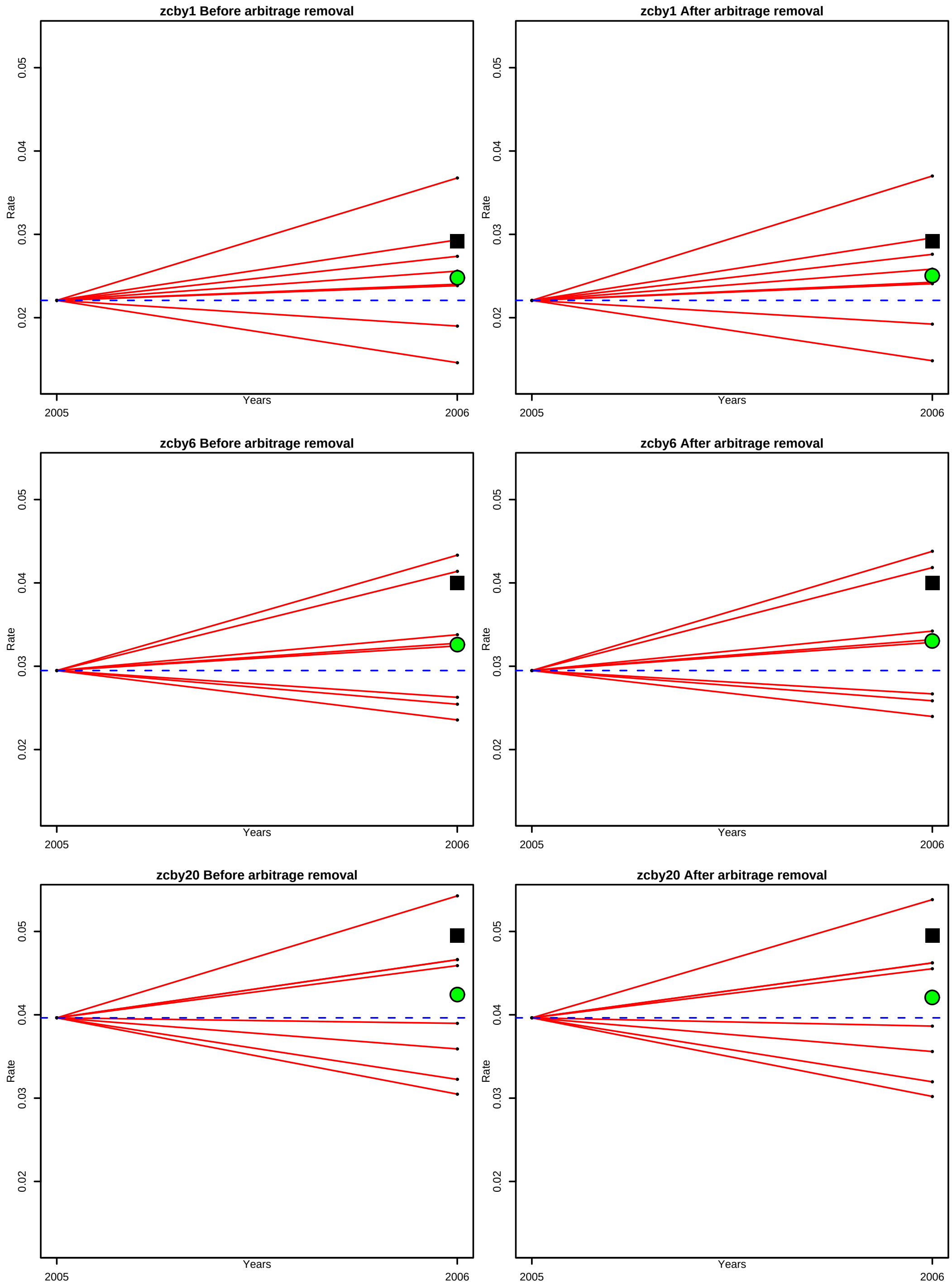


Figure 7.13: 8 Scenarios to Represent the Yields Curve of the 1, 6 and 20 Year Rates Before and After Arbitrage Removal. (Moment Matching is Based on 4.2 and Nelson-Siegel Smoothing is Used on Data Up To August 2005.)

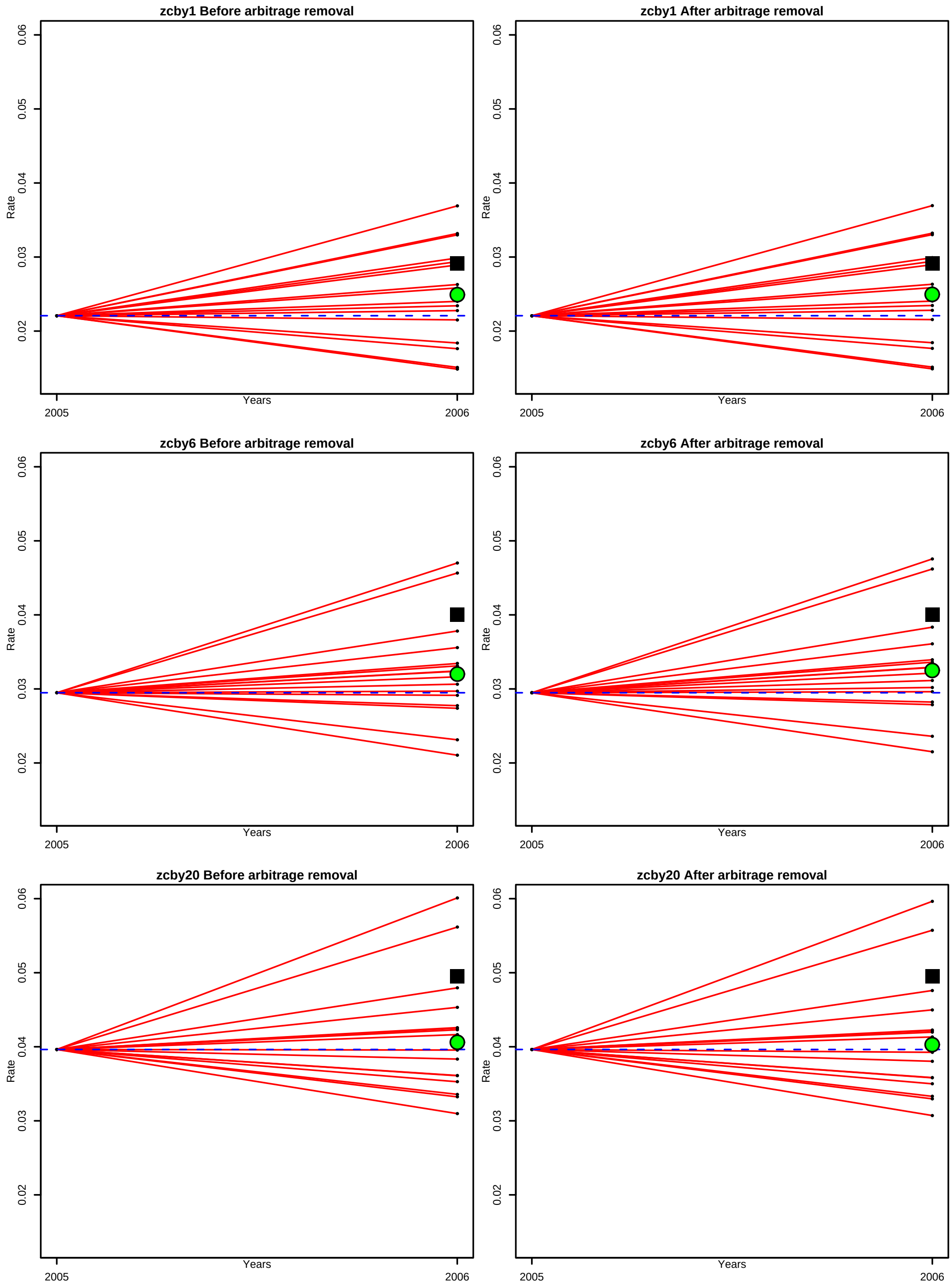


Figure 7.14: 16 Scenarios to Represent the Yields Curve of the 1, 6 and 20 Year Rates Before and After Arbitrage Removal. (Moment Matching is Based on 4.2 and Nelson-Siegel Smoothing is Used on Data Up To August 2005.)

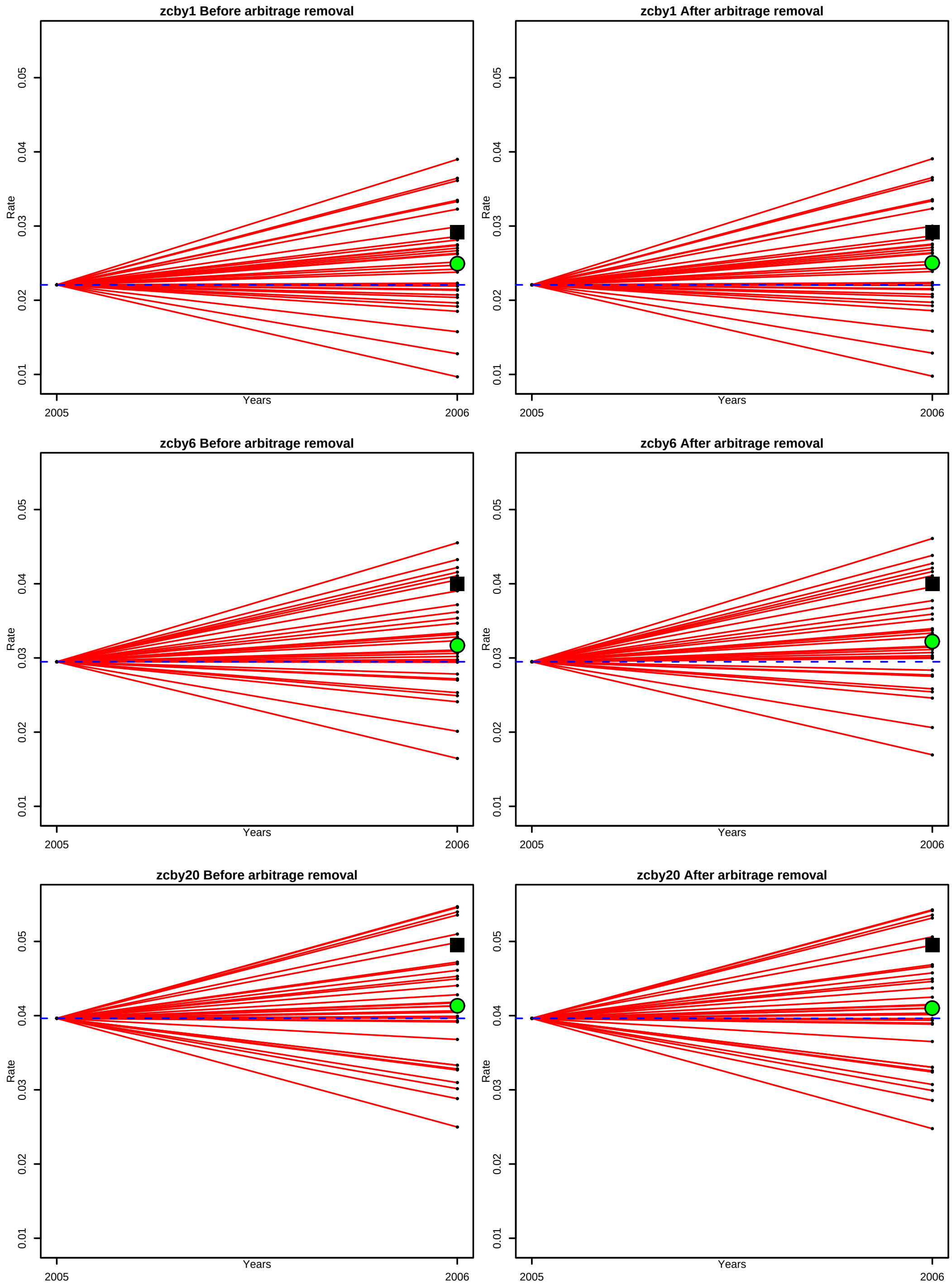


Figure 7.15: 32 Scenarios to Represent the Yields Curve of the 1, 6 and 20 Year Rates Before and After Arbitrage Removal. (Moment Matching is Based on 4.2 and Nelson-Siegel Smoothing is Used on Data Up To August 2005.)

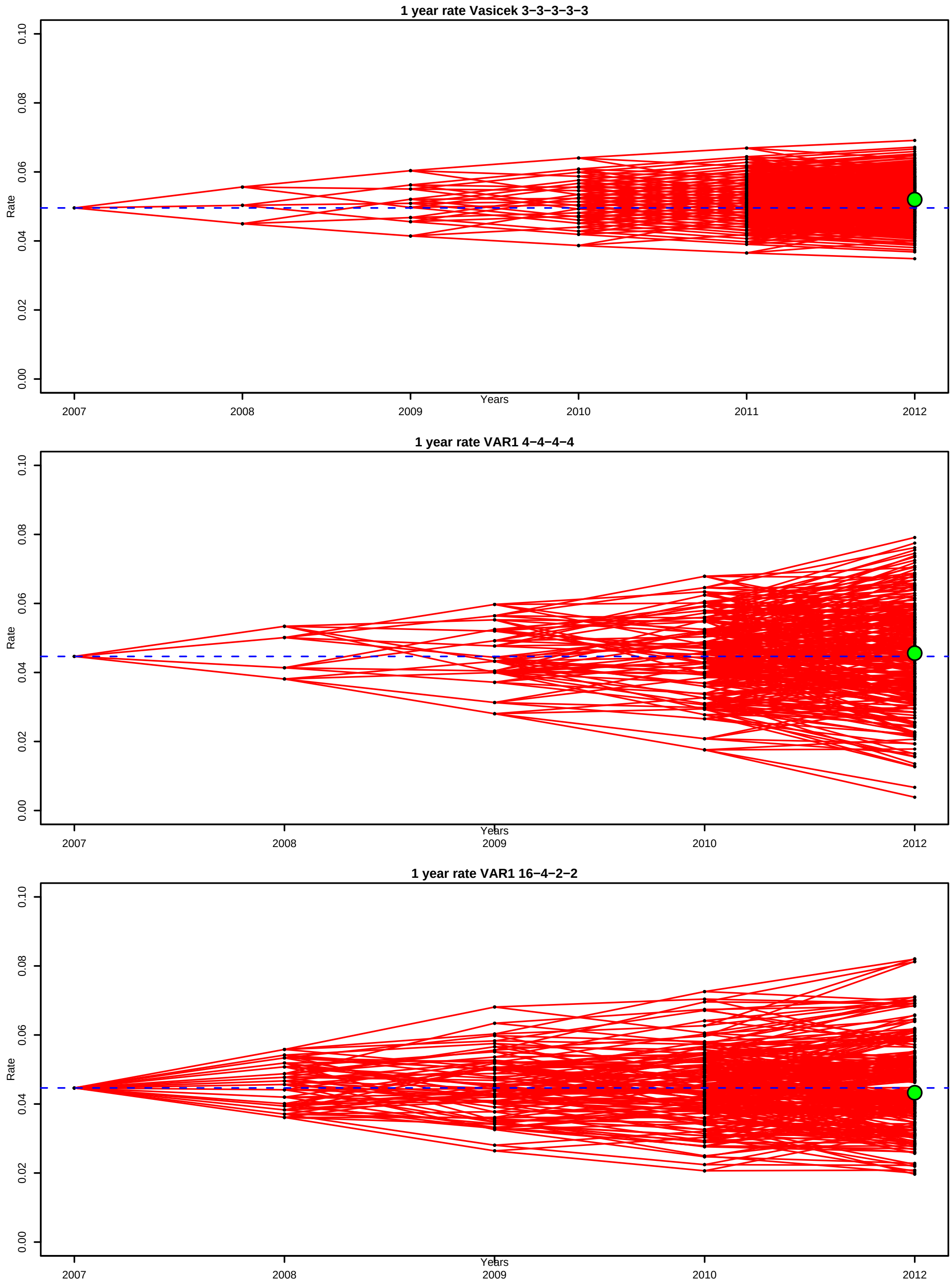


Figure 7.16: Comparing Scenarios for the 1–Year Rate as Achieved by the Vasicek and VAR1 Models for Different Tree Structures)

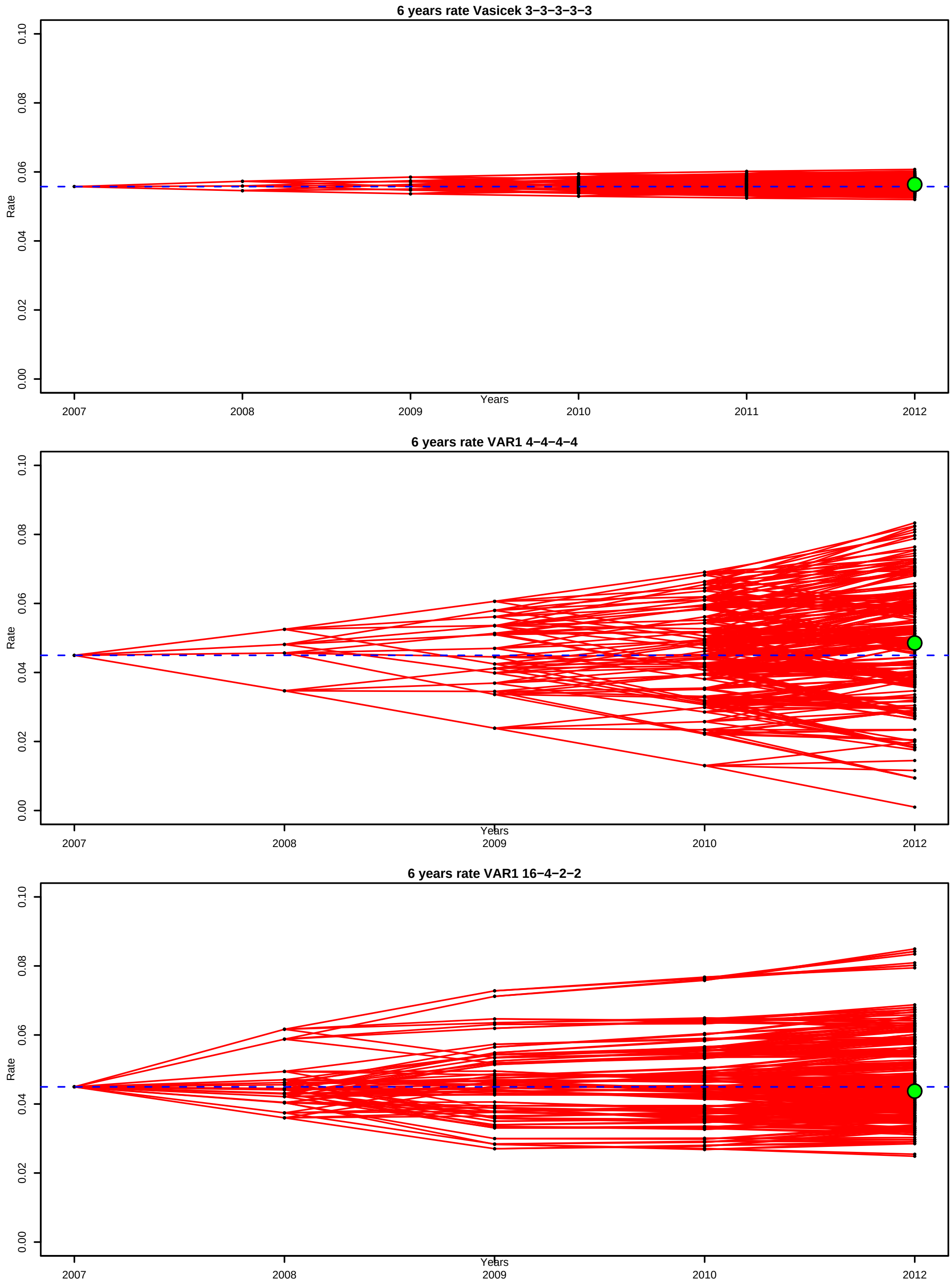


Figure 7.17: Comparing Scenarios for the 6–Year Rate as Achieved by the Vasicek and VAR1 Models for Different Tree Structures)

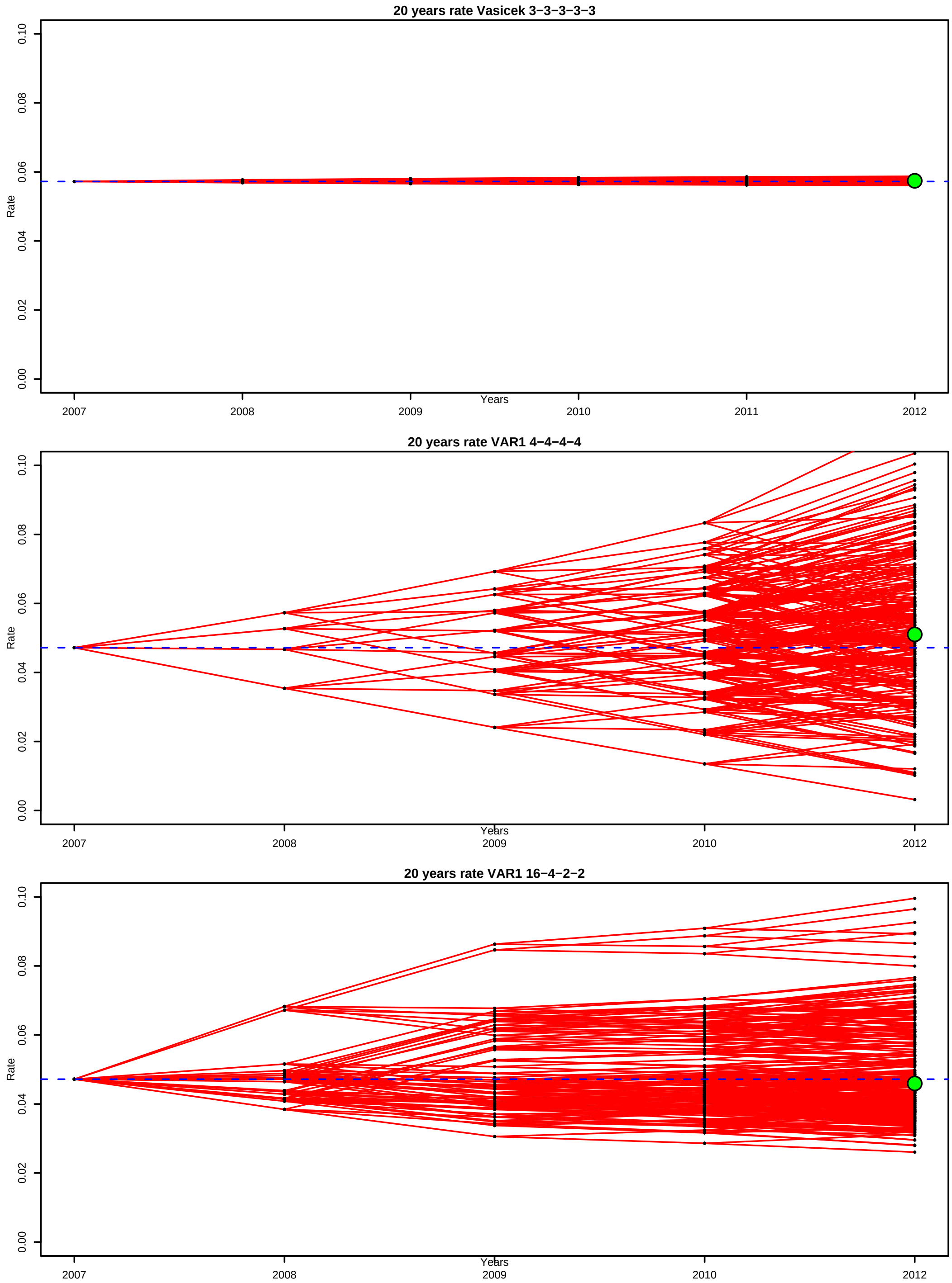


Figure 7.18: Comparing Scenarios for the 20–Year Rate as Achieved by the Vasicek and VAR1 Models for Different Tree Structures)

Chapter 8

Conclusions

8.1 Summary and Research Contribution

As in general stochastic programming, the principle of garbage in garbage out (GIGO) holds. Therefore, there is a steadily growing interest in the evaluation of good scenario generation methods. That is a result of the increasing computing power that allows stochastic programming to solve immense multi–stage optimization problems following the swell of industrial projects that are based on stochastic programming (such as the work done by Nykredit on the mortgagor problem, the work done by Pioneer Investment for pension funds, etc.)

It is not believed, however, that a general scenario generation for all multi–stage stochastic programming can be found. One of the vital contributions presented in this thesis is exploring specific scenario generation methods that are valid for the term structure of interest rate. This report therefore suggests an iterative mathematical program in order to receive high-quality scenario trees for interest rates by following these steps:

1. Choose appropriate and trusted (econometric) model.

2. Estimate model parameters.
3. Identify an appropriate theoretical scenario generation method for the problem domain.
4. Assemble the scenario generation process with the econometric factors.
5. Add domain specific constraints for the scenario generation process.
6. Generate scenario tree (for stochastic optimization).

The first two steps are accomplished by researching the term structure of interest rate, and identifying the interest rate factors and later simulating the corresponding parameters. These steps can help the scenario generation achieve consistent results according to the known literature about the term structure of interest rates. This will ensure that a scenario generation criterion for correctness is achieved.

The third step is to follow an understanding of the appropriate scenario generation methodology for the specific problem that is solved[1]. This will ensure that the scenario created satisfies the accuracy and consistency criteria for scenario generation.

Steps 4 and 5 ensure that the accomplished scenario generation methodology is domain specific and ensure that the scenarios received are correct for all of the mentioned quality criteria for scenario generation.

The process suggested above is useful for most scenario generation methods that are industry specific. Correctness parameters might be different when developing a scenario generation approach for Supply Chain Management (SCM). However, the same procedure can be used for a SCM scenario generation as well, with different model parameters and trusted modeling of the state of the art and the industry fundamentals in that specific field.

[1]For example, use the heuristic for scenario generation suggested in section 4.3, if the matched distribution resembles normal distribution because this approach is started by identifying the desired distribution based on a normal distribution.

The research carried out in this thesis has a number of other values – such as identifying different scenario generation approaches and studying the possibility to extend it for interest rate scenario generation. The moment matching scenario generation approach was extended to a model that identified the econometric parameters needed for interest rate scenario generation and an implementation of a VAR1 model that accomplishes these standards was presented. This study was further tested by looking into different variations of the VAR1 model, testing various time periods, diverse smoothing methods, an assortment of scenario generation approaches and the number of scenarios created. The results appear very promising and this approach might be further used in connection with the mortgagor optimization problem at Nykredit.

As mentioned, this project identifies the lack of standards for scenario generation qualities and presents an iterative conceptual method for domain specific scenario generation approaches.

8.2 Future Work

At present there is no standardization of scenario generation approaches. My desire is develop a standardization that could benefit the stochastic programming society. In addition, I encourage others to do further research on top of the suggested models as presented in this thesis.

- **Creating industry standards for scenario generation**

  This research has identified many instruments that are essential for different scenario generation approaches. It is also known that there is no dominant scenario generation strategy. Furthermore, it is almost impossible to develop one, since it is very difficult to compare different scenario generation approaches.

Today with the augmentation of computation power along with a growth in the use of stochastic programming techniques in the industry, scenario generation is identified to be an important pillar in the creation of products. Much research has been done in regards to scenario generation (Professor Wallace and Dr. Kaut at Norway, Professor Pflug and Professor Hochreiter in Austria, Professor Dempster and Dr Medova at Cambridge University, Professor Gautam Mitra of Brunnel University, etc). However, a useful in depth comparison of their results is hard to establish since there are no clear standards for scenario generation. I would suggest to establish a list of classical stochastic programming problems as standard problems that can be solved using scenario generation. These problems should include data to be used in order to estimate the scenario generator parameters and predict future scenarios. The result of this process could then be compared to other results achieved by the different scenario generators. This could lead to a more coherent discussion around the quality achieved by different scenario generation approaches and assure that future research will comply with the identified and accepted quality standards.

Many thorough multi–stage optimization problems can be found in the literature. The GIGO principle states that the usefulness of these well thought of projects in the sense of solving practical problems can not be achieved without a solid scenario generation technique presented. The author argues that there is a necessity to establishing a set of problems for stability testing of different methods and performing thorough research by comparing these approaches across different industries which is vital for the growth of the stochastic programming community.

– **Further extension of scenario generation approaches suggested in this thesis**

  ∗ **Extending the VAR1 model**

    The complete VAR1 model presented in chapter 6 has not yet been implemented as a stand alone stochastic program. This sort of implementation will allow the model to achieve better control over the sensitivity of its different parts. A comparative study of the two models, both the iterative model used in this report as well as the complete one stage model, could develop into a very interesting mathematical and practical study. Finding appropriate decomposition methods in order to try and solve the one stage problem could be extremely challenging.

∗ **Sensitivity test of scenario generation approaches**

As demonstrated, the scenario generation method used as part of the VAR1 model can lead to different scenario trees based on the values of different starting points. A sensitivity analysis study should be performed on the values received by the optimization model (in the case presented here, the mortgagor problem) to examine the stability of the results using different starting points.

∗ **Try different scenario generators**

It would be necessary to compare the efficiency of different scenario generators as part of the iterative procedure to generate interest rate scenarios.

∗ **Trying different arbitrage removal processes**

Currently the arbitrage removal process can effect the quality of the solutions found using moment matching. An alternative solution can be reached by adding non arbitrage constraints directly into the scenario generation process, as shown by Klaassen at [38].

∗ **Try different variations of the factor analysis**

Examine different interpretations of the yield curve, especially regarding the level, slope and curvature that can lead to better econometric parameters and suggest better correctness for the term structure scenario generation process.

Appendix A

Appendix 1 - More Test Results

A.1 May 2007 - Nelson Siegel Smoothing

A.2 Results Including All The Term Structure May 2007

32 scenarios

Here I show a figure of the complete term structure of interest rates for 32 scenarios perdicted based on information until May 2007. I hold similiar resuslts for other periods and different number of scenarios as well as for Nelson-Siegel smoothing with and without arbitrage. Please contact me if you are interested in the results or in receiving the R code for drawing the trees.

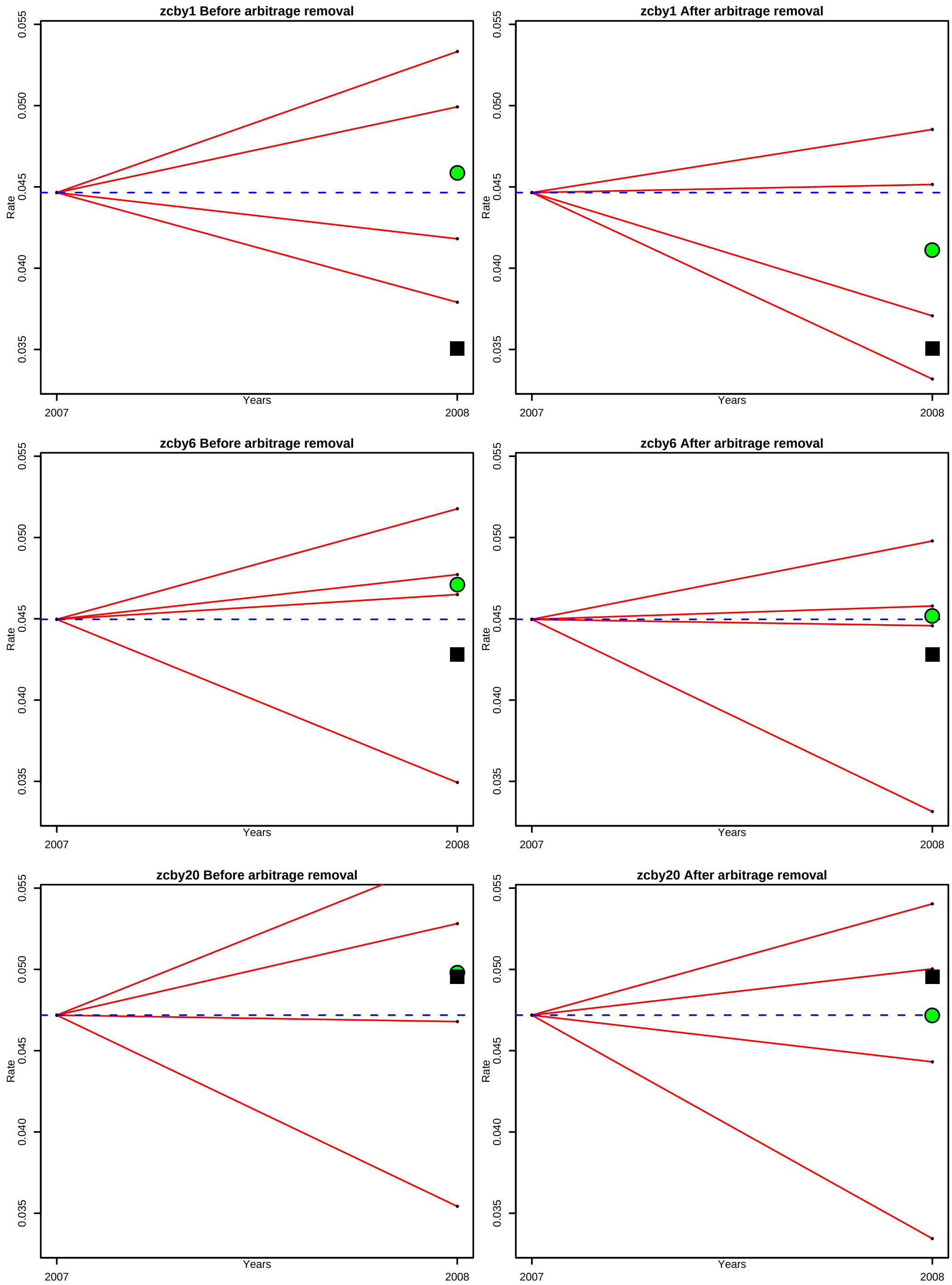


Figure A.1: 4 Scenarios to Represent the Yields Curve of the 1, 6 and 20 Year Rates Before and After Arbitrage Removal. (Moment Matching is Based on 4.2 and Nelson-Siegel Smoothing is Used on Data Up To May 2007.)

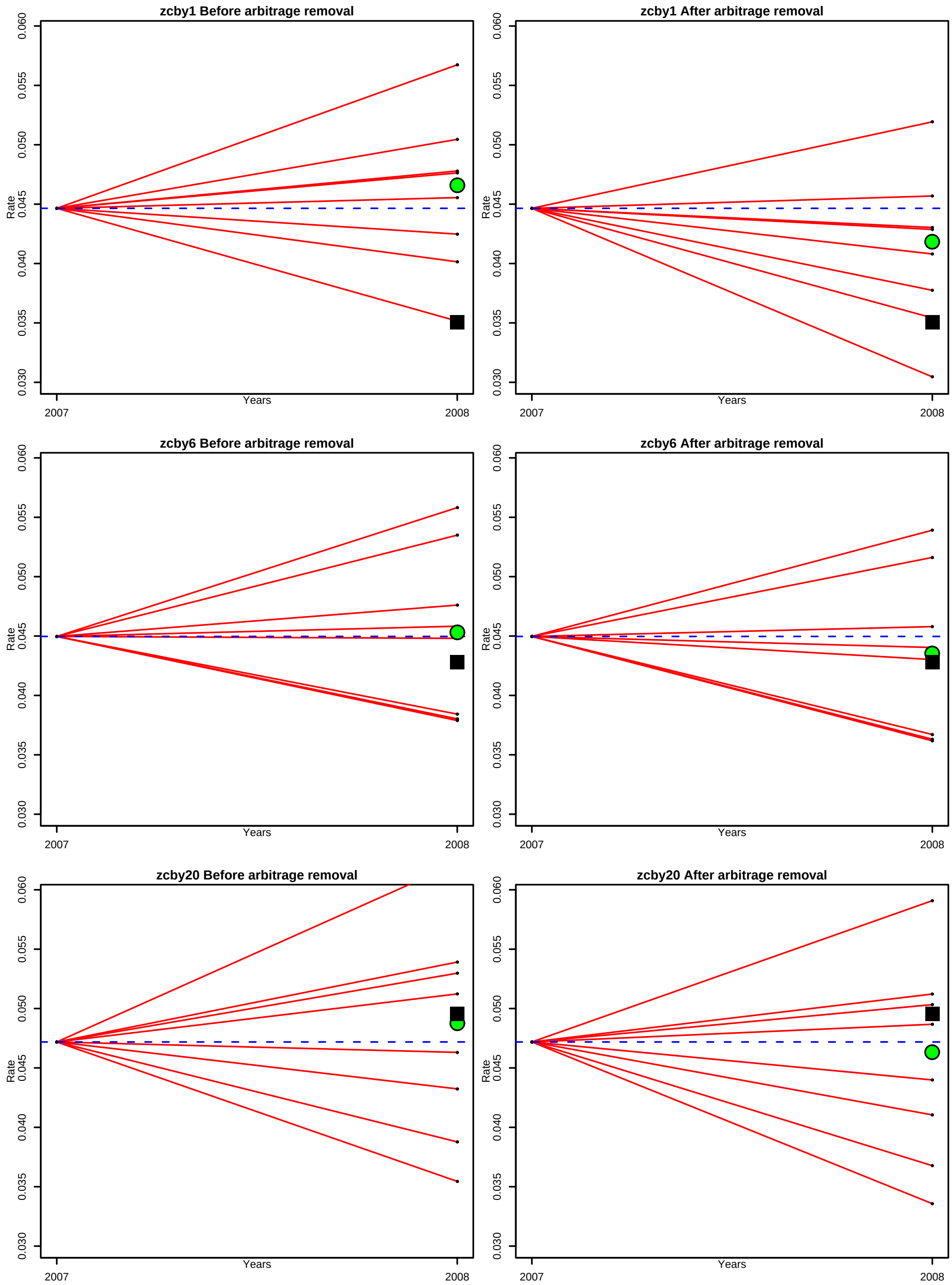


Figure A.2: 8 Scenarios to Represent the Yields Curve of the 1, 6 and 20 Year Rates Before and After Arbitrage Removal. (Moment Matching is Based on 4.2 and Nelson-Siegel Smoothing is Used on Data Up To May 2007.)

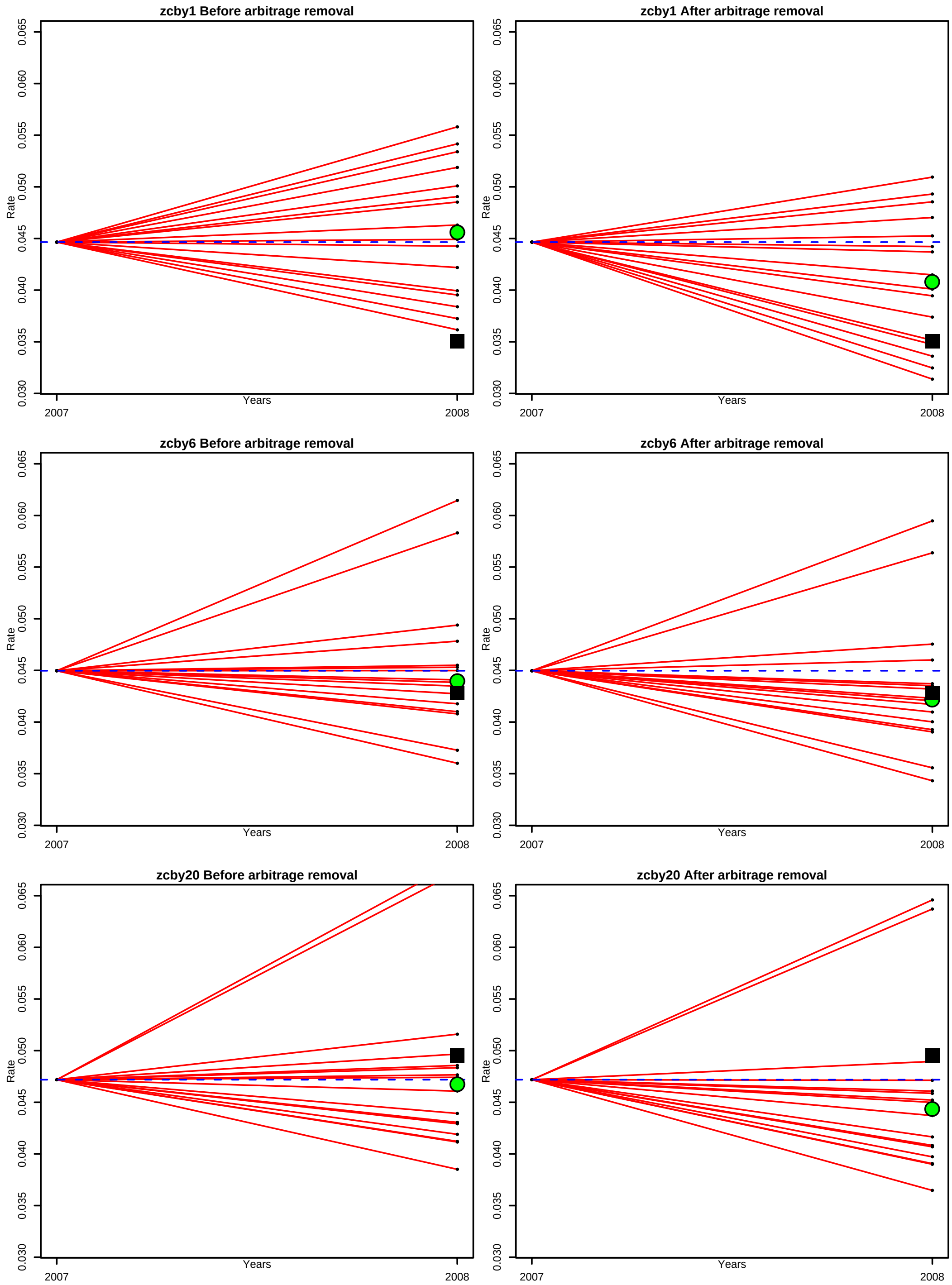


Figure A.3: 16 Scenarios to Represent the Yields Curve of the 1, 6 and 20 Year Rates Before and After Arbitrage Removal. (Moment Matching is Based on 4.2 and Nelson-Siegel Smoothing is Used on Data Up To May 2007.)

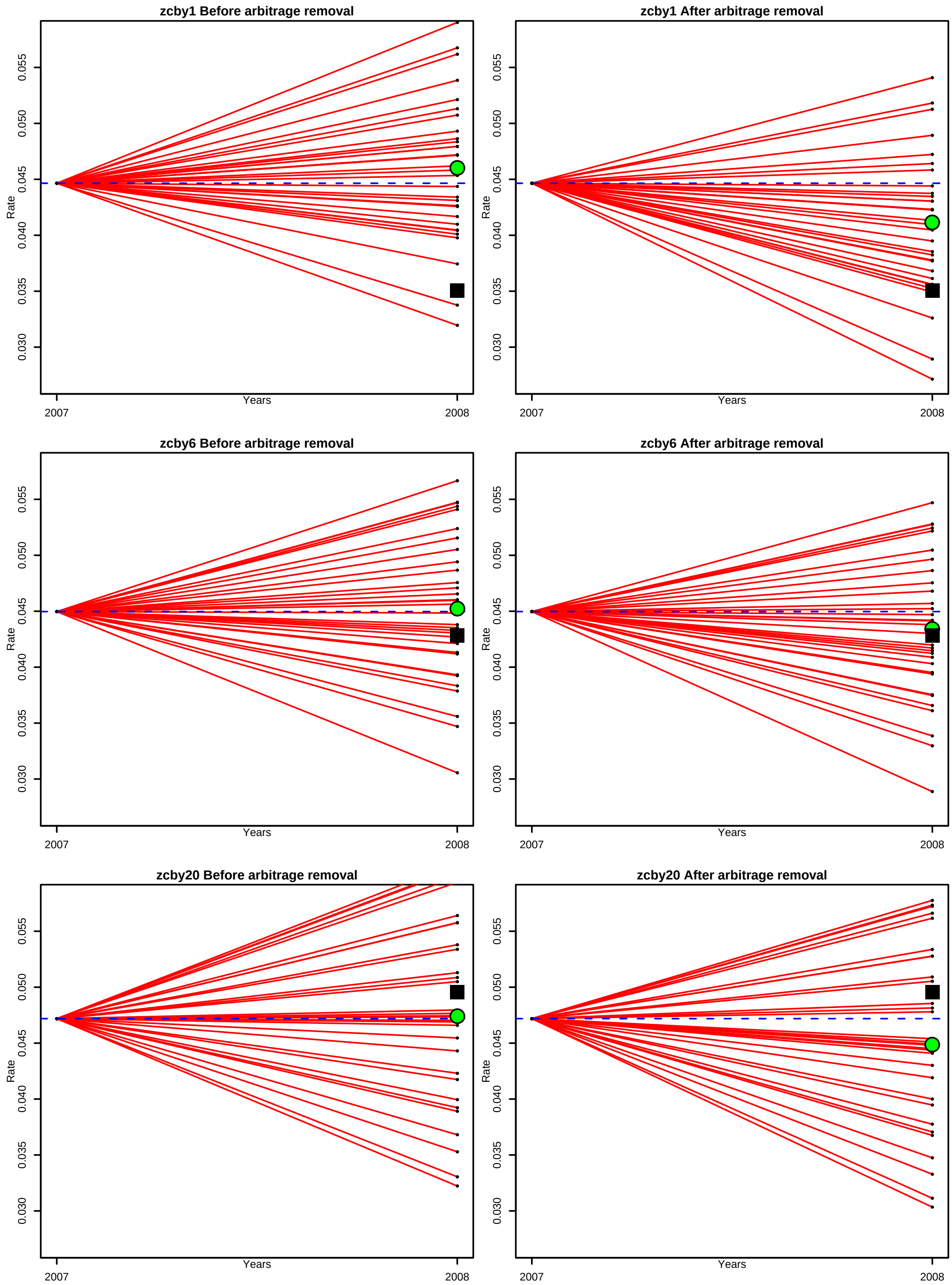


Figure A.4: 32 Scenarios to Represent the Yields Curve of the 1, 6 and 20 Year Rates Before and After Arbitrage Removal. (Moment Matching is Based on 4.2 and Nelson-Siegel Smoothing is Used on Data Up To May 2007.)

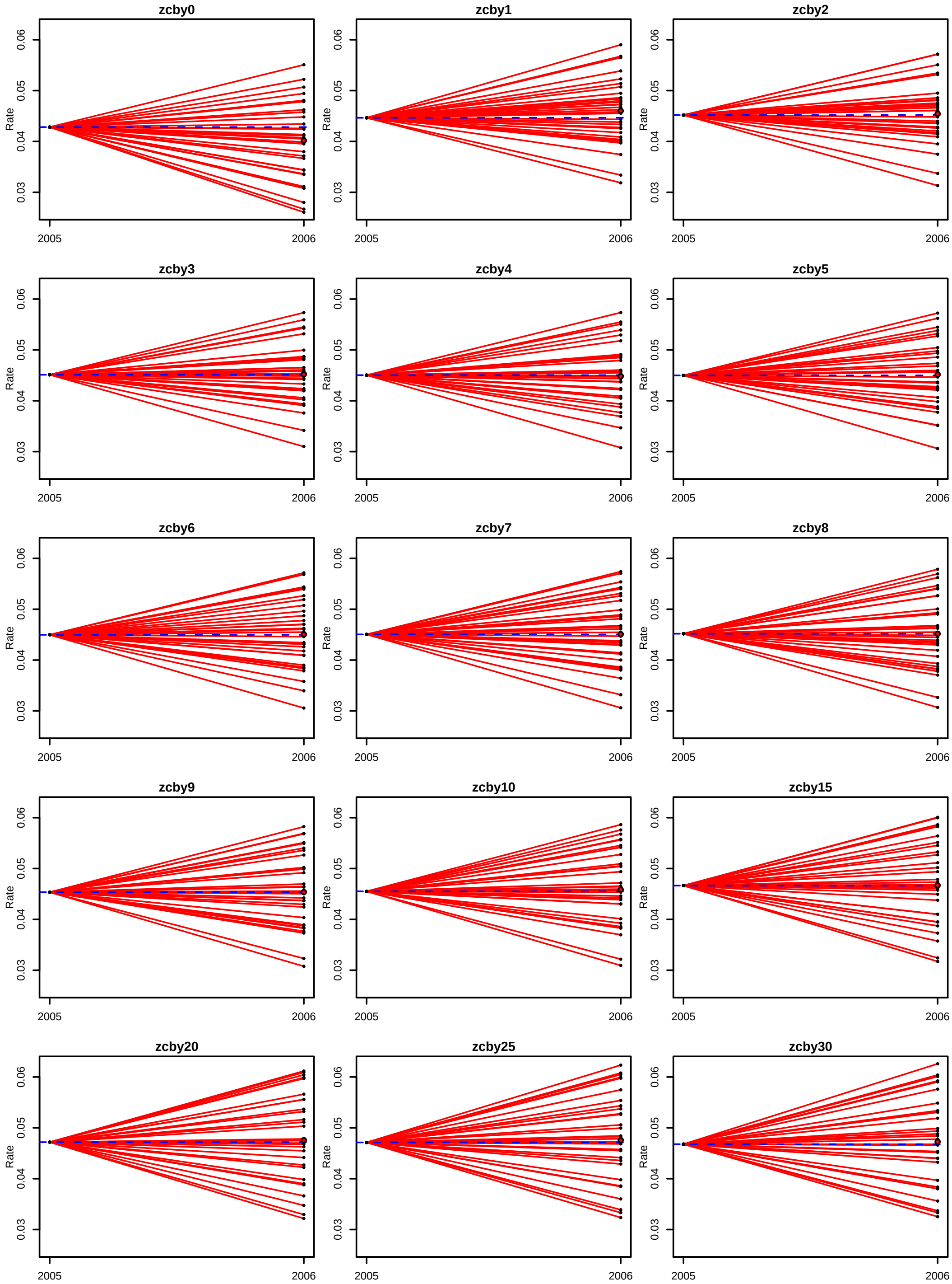


Figure A.5: Term Structure Generated for 32 Scenarios, Affine Smoothing and No Arbitrage Removal

Bibliography

[1] Beale, On minimizing a convex function. subject to linear inequalities J. Royal Statistical Society, Series B 17 (1955) pp. 173-184

[2] J.R. Birge and F. Louveaux, Introduction to Stochastic Programming. Springer Series in Operations Research and Financial Engineering Heidelberg, 1997.

[3] G.B. Dantzig, Linear programming under uncertainty. Management Science 1 (1955) pp. 197-206.

[4] K. Høiland and S. W. Wallace Generating Scenario Trees for Multistage Decision Problems. 2001, Management Science, Vol. 47, No. 2 (Feb., 2001), pp. 295-307

[5] K. Høiland, M. Kaut and S. W. Wallace A Heuristic for Moment-matching Scenario Generation. Computational Optimization and Applications, 24 (23), pp. 169185, 2003.

[6] K. Høiland, M. Kaut and S. W. Wallace Description of data used in numerical tests in the paper A Heuristic for Moment-matching Scenario Generation. 2002

[7] M. Kaut Updates to the published version of A Heuristic for Moment-matching Scenario Generation. 2003

[8] M. Kaut and S. W. Wallace Evaluation of scenario-generation methods for stochastic programming. Accepted in Pacific Journal of Optimization, to appear in 2007

[9] D. Lando, R. Poulsen, Lecture Notes for the Course Investerings- og Finansieringsteori. January 2006.

[10] S. Nielsen and R. Poulsen, A Two-Factor Stochastic Programming Model of Danish Mortgage-Backed Securities. Journal of Economic Dynamics and Control . Vol 28(7), pp 1267-1289 (2004).

[11] K. M. Rasmussen and J. Clausen. Mortgage Loan Portfolio Optimization Using Multi Stage Stochastic Programming. November, 2005.

[12] K.M. Ramsmussen and S.A. Zenios. A multistage model for optimizing diversification strategies of mortgage loans. Working paper, HERMES European Center of Excellence on Computational Finance and Economics, School of Economics and Management University of Cyprus, Nicosia, CY, 2006.

[13] K. M. Rasmussen and S. A. Zenios Well ARMed and FiRM: Diversification of mortgage loans for homeowners. Working Paper 06-02 HERMES Center on Computational Finance, University of Cyprus, Nicosia, May 2006.

[14] S. A. Zenios Practical Financial Optimization. June 2005.

[15] Wikipedia the free encyclopedia, http://www.wikipedia.org/

[16] Wolfram Mathworld, http://mathworld.wolfram.com/ [26]

[17] R. Hochreiter and G. Ch. Scenario tree generation as a multidimensional facility location problem. AURORA TR2002-33, December 2002. [25]

[18] A. Futschik , G.Ch. Pflug Optimal allocation of simulation experiments in discrete stochastic optimization and approximative algorithms. European Journal of Operational Research 101 245-260, 1997 [25]

[19] G.Ch. Pflug Scenario tree generation for multiperiod financial optimization by optimal discretization. Math. Program., Ser. B 89: 251271 (2001), Springer-Verlag 2000 [18, 30, 43, 45, 53, 57, 58, 68, 74, 107, 109]

[20] G.CH. PFLUG, A. SWIETANOWSKI, E. DOCKNER and H. MORITSCH The AURORA Financial Management System: Model and Parallel Implementation Design. Annals of Operations Research 99, 189206, 2000 [8, 18, 30, 43, 60, 62, 65, 67, 68, 112]

[21] Hans W. Moritsch and Georg Ch. Pflug Using a Distributed Active Tree in Java for the Parallel and Distributed Implementation of a Nested Optimization Algorithm. Proceedings of the 2003 International Conference on Parallel Processing Workshops, IEEE [18, 45, 107]

[22] Nalan Gulpinar, Berc Rustem and Reuben Settergren Multistage Stochastic Mean-Variance Portfolio Analysis with Transaction Costs. May, 2004 [-]

[23] P. Klaassen Comment on Generating Scenario Trees for Multistage Decision Problems. Management Science, Vol. 48, No. 11, November 2002 [26, 35, 43, 102]

[24] J. Dupacova and J. Polivka ASSET-LIABILITY MANAGEMENT FOR CZECH PENSION FUNDS USING STOCHASTIC PROGRAMMING. [82]

[25] COSP - Stochastic Programming Community, http://stoprog.org/ [17, 101]

[26] Ronald Hochreiter Scenario Optimization for Multi-Stage Stochastic Programming Problems. [17, 101]

[27] J. Dupacova, G. Consigli and S. Wallace Generating scenarios for multistage stochastic programs. Annals of Operations Research 100 (2000) 25-53 [17]

[28] H. Heitsch and W. *Römisch* Scenario reduction algorithms in stochastic programming. Computational Optimization and Applications 24 (2003) 187-206

[29] J. Dupacova , N. Groewe-Kuska and W. *Römisch* Scenario reduction in stochastic programming: An approach using probability metrics. Mathematical Programming, Series A 95 (2003) 493-511

[30] T. Pennanen Epi-convergent discretizations of multistage stochastic programs. Mathematics of Operations Research 30 (2005) 245-256

[31] T. Pennanen and M. Koivu Epi-convergent discretizations of stochastic programs via integration quadratures. Numerische Mathematik. to appear.

[32] Mico Loretan. Generating market risk scenarios using principal components analysis: Methodological and practical considerations. In The Measurement of Aggregate Market Risk, CGFS Publications No. 7, pages 2360. Bank for International Settlements, November 1997. Available at http://www.bis.org/publ/ecsc07.htm.

[33] Roger Halldin. Scenario Trees for Inflow Modelling in Stochastic Optimisation for Energy Planning. PhD thesis, Lund Unversity, Sweden, 2002.

[34] N. Topaloglou, Vladimirou H., and S. A. Zenios. CVaR models with selective hedging for international asset allocation. Journal of Banking and Finance, 26(7):15351561, 2002.

[35] T. Pennanen and M. Koivu. Integration quadratures in discretization of stochastic programs. Stochastic Programming E-Print Series, http://www.speps.info, May 2002.

[36] R. R. P. Kouwenberg. Scenario generation and stochastic programming models for asset liability management. European Journal of Operational Research, 134(2):5164, 2001

[37] D. R. Carino, T. Kent, D. H. Myers, C. Stacy, M. Sylvanus, A. L. Turner, K.Watanabe, andW. T. Ziemba. The Russell-Yasuda Kasai model: an asset liability model for a Japanese insurance company using multistage stochastic programming. INTERFACES, 24(1):2949, 1994.

[38] P. Klaassen Comment on Generating Scenario Trees for Multistage Decision Problems. Management Science Vol. 48, No. 11, November 2002

[39] K. M. Rasmussen and R Poulsen Yield curve event tree construction for multi stage stochastic programming models. Techcnical Report, July 2007

[40] S. Sukhodolska Integrated Mortgage and Pension Portfolio Management for Households. M.Sc. Thesis, DTU 2006

[41] P. J. Knez, R. Litterman and J. Scheinkman Explorations Into Factors Explaining Money Market Returns. Journal of Finance, 49(5):1861-1882 , 1994.

[42] R. Litterman and J. Scheinkman, Common Factors Affecting Bond Returns. The journal of fixed income, June 1991, 54-61.

[43] H. Dahl A Flexible Approach to Interest–Rate Risk Management. Financial Optimization 1994, 189-209

[44] M. Bertocchi, R. Giacometti and S. A. Zenios, Risk Factor Analysis and Portfolio Immunization in the Corporate Bond Market. European Journal of Operational Research, Vol 161(2):348-363, 2005

[45] J. E. Jr., Ingersoll, Theory of Financial Decision Making. Rowman Littlefield, Totowa, NJ. 1987.

[46] P. Klaassen, Discretized reality and spurious profits in stochastic programming models for asset/liability management. Eur. J. Oper. Res. 101(2) 374392, 1997.

[47] G. Pflug and R. Hochreiter Scenario generation for stochsatic multi-stage decision processes as facility location problems. AURORA TR2002-33, December 2002.

[48] R. Kouwenberg. Scenario generation and stochastic programming models for asset liability management. European Journal of Operation Research, 2001.

[49] Johan Lyhagen. A method to generate multivariate data with moments arbitrary close to the desired moments. Working paper 481, Stockholm School of Economics, December 2001.

[50] Jessica James and Nick Webber Interest Rate Modelling: Financial Engineering. Wiley, 2000

[51] Damiano Brigo and Fabio Mercurio Interest Rate Models - Theory and Practice: With Smile, Inflation and Credit. Springer Finance, 2006.

[52] N. Gulpinar, B. Rustem, and R. Settergren. Optimisation and simulation approaches to scenario tree generation. Journal of Economics Dynamics and Control, 2002.

[53] J. L. Sigle and S. Sen Stochastic decomposition. An algorithm for two stage stochastic programs with recourse. Math Oper Res 16, 650-669, 1991.

[54] G. Infager Planning Under Uncertainty - Solving Large Scale Stochastic Linear Programs. Boyd and Fraser ,1994

[55] A. C. Miller and T.R. Rice Discrete approximations of probability distributions. Management Science 22, 340-358. Boyd and Fraser ,1994

[56] C.D. Vale and V.A. Maurelli, Simulating multivariate nonnormal distributions. Psychometrika, vol. 48, no. 3, pp. 465471, 1983.

[57] P.M. Lurie and M.S. Goldberg, An approximate method for sampling correlated random variables from partially-specified distributions. Management Science, vol. 44, no. 2, pp. 203218, 1998.

[58] A.I. Fleishman, A method for simulating nonnormal distributions. Psychometrika, vol. 43, pp. 521532, 1978.

[59] A.C. Bester Random Field and Affine Models for Interest Rates: An Empirical Comparison. 2004

[60] O. Vasicek, An Equilibrium Characterization of the Term Structure. Journal of Financial, 1977. Economics, 5, 177188.

[61] J. C. Cox, J. E. Ingersoll, and S. A. Ross, A Theory of the Term Structure of Interest Rates. Econometrica, 53, 385408, 1985.

[62] D. Duffie and R. Kan, A Yield-Factor Model of Interest Rates. Mathematical Finance, 6, 1996. 379406.

[63] Q. Dai and K.J. Singleton, Specification Analysis of Affine Term Structure Models. Journal of Finance, 55, 19431978, 2000.

[64] D. Duffie, J. Pan, and K.J. Singleton, Transform Analysis and Asset Pricing for Affine Jump Diffusions. Econometrica, 68, 13431367, 2000.

[65] K. C. Chan, G.A. Karolyi, F.A. Longstaff, and A.B. Sanders An Empirical Comparison of Alternative Models of the Short-Term Interest Rate. Journal of Finance, 47, 12091227, 1992.

[66] C.R. Nelson and A.F. Siegel, Parsimonious modeling of yield curves. Journal of Business, 60, 473-489, 1987.

[67] A.S. Hurn, K.A. Lindsay and V. Pavlov Smooth estimation of yield curves by Laguerre functions. 2006.s

[68] M A H Dempster, Institutional ALM with Risk Management. Presentation given at the Spring School on Stochastic Programming Bergamo 16 April 2007.

[17]

[7, 8, 18, 35, 36, 38, 41, 69, 70, 88, 89, 90, 101] [-]