A Workout in Computational Finance
For other titles in the Wiley Finance series please see www.wiley.com/go/finance
A Workout in Computational Finance

Michael Aichinger
Andreas Binder
To Elke, Michael, Lisa and Florian

To Julian
4.2 Black76 for Caps and Swaptions 43
4.3 One-Factor Short Rate Models 45
    4.3.1 Prominent Short Rate Models 45
4.4 The Hull-White Model in More Detail 46
4.5 Trinomial Trees 47

5 Upwinding Techniques for Short Rate Models 55
    5.1 Derivation of a PDE for Short Rate Models 55
5.2 Upwind Schemes 56
    5.2.1 Model Equation 57
5.3 A Puttable Fixed Rate Bond under the Hull-White One Factor Model 63
    5.3.1 Bond Details 64
    5.3.2 Model Details 64
    5.3.3 Numerical Method 65
    5.3.4 An Algorithm in Pseudocode 68
    5.3.5 Results 69

6 Boundary, Terminal and Interface Conditions and their Influence 71
    6.1 Terminal Conditions for Equity Options 71
    6.2 Terminal Conditions for Fixed Income Instruments 72
    6.3 Callability and Bermudan Options 74
    6.4 Dividends 74
    6.5 Snowballs and TARNs 75
    6.6 Boundary Conditions 77
        6.6.1 Double Barrier Options and Dirichlet Boundary Conditions 77
        6.6.2 Artificial Boundary Conditions and the Neumann Case 78

7 Finite Element Methods 81
    7.1 Introduction 81
        7.1.1 Weighted Residual Methods 81
        7.1.2 Basic Steps 82
    7.2 Grid Generation 83
    7.3 Elements 85
        7.3.1 1D Elements 86
        7.3.2 2D Elements 88
    7.4 The Assembling Process 90
        7.4.1 Element Matrices 93
        7.4.2 Time Discretization 97
        7.4.3 Global Matrices 98
        7.4.4 Boundary Conditions 101
        7.4.5 Application of the Finite Element Method to
            Convection-Diffusion-Reaction Problems 103
    7.5 A Zero Coupon Bond Under the Two Factor Hull-White Model 105
    7.6 Appendix: Higher Order Elements 107
        7.6.1 3D Elements 109
        7.6.2 Local and Natural Coordinates 111
8 Solving Systems of Linear Equations

8.1 Direct Methods
8.1.1 Gaussian Elimination
8.1.2 Thomas Algorithm
8.1.3 LU Decomposition
8.1.4 Cholesky Decomposition
8.2 Iterative Solvers
8.2.1 Matrix Decomposition
8.2.2 Krylov Methods
8.2.3 Multigrid Solvers
8.2.4 Preconditioning

9 Monte Carlo Simulation

9.1 The Principles of Monte Carlo Integration
9.2 Pricing Derivatives with Monte Carlo Methods
9.2.1 Discretizing the Stochastic Differential Equation
9.2.2 Pricing Formalism
9.2.3 Valuation of a Steepeener under a Two Factor Hull-White Model
9.3 An Introduction to the Libor Market Model
9.4 Random Number Generation
9.4.1 Properties of a Random Number Generator
9.4.2 Uniform Variates
9.4.3 Random Vectors
9.4.4 Recent Developments in Random Number Generation
9.4.5 Transforming Variables
9.4.6 Random Number Generation for Commonly Used Distributions

10 Advanced Monte Carlo Techniques

10.1 Variance Reduction Techniques
10.1.1 Antithetic Variates
10.1.2 Control Variates
10.1.3 Conditioning
10.1.4 Additional Techniques for Variance Reduction
10.2 Quasi Monte Carlo Method
10.2.1 Low-Discrepancy Sequences
10.2.2 Randomizing QMC
10.3 Brownian Bridge Technique
10.3.1 A Steepeener under a Libor Market Model

11 Valuation of Financial Instruments with Embedded American/Bermudan Options within Monte Carlo Frameworks

11.1 Pricing American options using the Longstaff and Schwartz algorithm
11.2 A Modified Least Squares Monte Carlo Algorithm for Bermudan Callable Interest Rate Instruments
11.2.1 Algorithm: Extended LSMC Method for Bermudan Options
11.2.2 Notes on Basis Functions and Regression
11.3 Examples 186
   11.3.1 A Bermudan Callable Floater under Different Short-rate Models 186
   11.3.2 A Bermudan Callable Steepener Swap under a Two Factor Hull-White Model 188
   11.3.3 A Bermudan Callable Steepener Cross Currency Swap in a 3D IR/FX Model Framework 189

12 Characteristic Function Methods for Option Pricing 193
   12.1 Equity Models 194
      12.1.1 Heston Model 196
      12.1.2 Jump Diffusion Models 198
      12.1.3 Infinite Activity Models 199
      12.1.4 Bates Model 200
   12.2 Fourier Techniques 201
      12.2.1 Fast Fourier Transform Methods 201
      12.2.2 Fourier-Cosine Expansion Methods 203

13 Numerical Methods for the Solution of PIDEs 209
   13.1 A PIDE for Jump Models 209
   13.2 Numerical Solution of the PIDE 210
      13.2.1 Discretization of the Spatial Domain 211
      13.2.2 Discretization of the Time Domain 211
      13.2.3 A European Option under the Kou Jump Diffusion Model 212
   13.3 Appendix: Numerical Integration via Newton-Cotes Formulae 214

14 Copulas and the Pitfalls of Correlation 217
   14.1 Correlation 218
      14.1.1 Pearson’s $\rho$ 218
      14.1.2 Spearman’s $\rho$ 218
      14.1.3 Kendall’s $\tau$ 220
      14.1.4 Other Measures 221
   14.2 Copulas 221
      14.2.1 Basic Concepts 222
      14.2.2 Important Copula Functions 222
      14.2.3 Parameter estimation and sampling 229
      14.2.4 Default Probabilities for Credit Derivatives 234

15 Parameter Calibration and Inverse Problems 239
   15.1 Implied Black-Scholes Volatilities 239
   15.2 Calibration Problems for Yield Curves 240
   15.3 Reversion Speed and Volatility 245
   15.4 Local Volatility 245
      15.4.1 Dupire’s Inversion Formula 246
      15.4.2 Identifying Local Volatility 246
      15.4.3 Results 247
   15.5 Identifying Parameters in Volatility Models 248
      15.5.1 Model Calibration for the FTSE-100 249
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>Optimization Techniques</td>
<td>253</td>
</tr>
<tr>
<td>16.1</td>
<td>Model Calibration and Optimization</td>
<td>255</td>
</tr>
<tr>
<td>16.1.1</td>
<td>Gradient-Based Algorithms for Nonlinear Least Squares Problems</td>
<td>256</td>
</tr>
<tr>
<td>16.2</td>
<td>Heuristically Inspired Algorithms</td>
<td>258</td>
</tr>
<tr>
<td>16.2.1</td>
<td>Simulated Annealing</td>
<td>259</td>
</tr>
<tr>
<td>16.2.2</td>
<td>Differential Evolution</td>
<td>260</td>
</tr>
<tr>
<td>16.3</td>
<td>A Hybrid Algorithm for Heston Model Calibration</td>
<td>261</td>
</tr>
<tr>
<td>16.4</td>
<td>Portfolio Optimization</td>
<td>265</td>
</tr>
<tr>
<td>17</td>
<td>Risk Management</td>
<td>269</td>
</tr>
<tr>
<td>17.1</td>
<td>Value at Risk and Expected Shortfall</td>
<td>269</td>
</tr>
<tr>
<td>17.1.1</td>
<td>Parametric VaR</td>
<td>270</td>
</tr>
<tr>
<td>17.1.2</td>
<td>Historical VaR</td>
<td>272</td>
</tr>
<tr>
<td>17.1.3</td>
<td>Monte Carlo VaR</td>
<td>273</td>
</tr>
<tr>
<td>17.1.4</td>
<td>Individual and Contribution VaR</td>
<td>274</td>
</tr>
<tr>
<td>17.2</td>
<td>Principal Component Analysis</td>
<td>276</td>
</tr>
<tr>
<td>17.2.1</td>
<td>Principal Component Analysis for Non-scalar Risk Factors</td>
<td>276</td>
</tr>
<tr>
<td>17.2.2</td>
<td>Principal Components for Fast Valuation</td>
<td>277</td>
</tr>
<tr>
<td>17.3</td>
<td>Extreme Value Theory</td>
<td>278</td>
</tr>
<tr>
<td>18</td>
<td>Quantitative Finance on Parallel Architectures</td>
<td>285</td>
</tr>
<tr>
<td>18.1</td>
<td>A Short Introduction to Parallel Computing</td>
<td>285</td>
</tr>
<tr>
<td>18.2</td>
<td>Different Levels of Parallelization</td>
<td>288</td>
</tr>
<tr>
<td>18.3</td>
<td>GPU Programming</td>
<td>288</td>
</tr>
<tr>
<td>18.3.1</td>
<td>CUDA and OpenCL</td>
<td>289</td>
</tr>
<tr>
<td>18.3.2</td>
<td>Memory</td>
<td>289</td>
</tr>
<tr>
<td>18.4</td>
<td>Parallelization of Single Instrument Valuations using (Q)MC</td>
<td>290</td>
</tr>
<tr>
<td>18.5</td>
<td>Parallelization of Hybrid Calibration Algorithms</td>
<td>291</td>
</tr>
<tr>
<td>18.5.1</td>
<td>Implementation Details</td>
<td>292</td>
</tr>
<tr>
<td>18.5.2</td>
<td>Results</td>
<td>295</td>
</tr>
<tr>
<td>19</td>
<td>Building Large Software Systems for the Financial Industry</td>
<td>297</td>
</tr>
<tr>
<td></td>
<td>Bibliography</td>
<td>301</td>
</tr>
<tr>
<td></td>
<td>Index</td>
<td>307</td>
</tr>
</tbody>
</table>
Acknowledgements

The authors would not have been able to write this book without the support of their colleagues: Johannes Fürst and Christian Kletzmayr provided a lot of detailed calculations. Sascha Kratky has been an invaluable help for technical support. Andreas Obereder tried (but did not always succeed) to organize tight deadlines. Michael Schwaiger, as the UnRisk product manager, was a great help in providing software parts. Stefan Janecek is the one who improved the English language of the first version.

AB wants to thank Heinz W. Engl for being his academic teacher and mentor for the last 30 years.

MA wants to thank Herbert Exner for many fruitful discussions that led to valuable insight. MA and AB are grateful to Wiley for their understanding when deadlines did not hold.
Michael Aichinger

Michael Aichinger, FRM, obtained his Ph.D. in Theoretical Physics from the Johannes Kepler Universität Linz with a thesis on numerical methods in density functional theory and their application to 2D finite electron systems. A mobility grant led him to the Texas A&M University (2003) and to the Helsinki University of Technology (2004). In 2007, Michael Aichinger joined the Industrial Mathematics Competence Center, where he has been working as a senior researcher and consultant in the field of quantitative finance for the last five years. He also works for the Austrian Academy of Sciences at the Radon Institute for Computational and Applied Mathematics, where he is involved in several industrial mathematics and computational physics projects.

Andreas Binder

Introduction and Reading Guide

1

PROLOGUE

We wrote this book with the aim of giving practitioners in computational finance a sound overview of relevant numerical methods. Some of the methods presented in this book are widely used today, while others should, in our opinion, gain more importance in the future. By, “computational finance” we loosely refer to all tasks related to the valuation of financial instruments, risk analysis and some aspects of risk management. Together with our colleagues at MathConsult GmbH, we have been working on a wide range of computational finance projects since 1997. During that time, we have observed that the numerical quality of software used in financial institutions widely varies.

Particular attention is thus given to working out the strengths and weaknesses of the different methods, and to reveal possible traps in their application. We have used real-world examples of valuation, risk analysis and calibration of specific financial instruments and models to introduce each method. A strong emphasis is laid on stable and robust schemes for the numerical treatment.

We have named the book “A Workout in Computational Finance” because due to our experience in training finance professionals, it is our strong belief that computational methods are best studied in a practical, hands-on approach, requiring the student to write at least part of the program code herself. To facilitate this style of learning, the book comes with accompanying software distilled from the UnRisk software package.¹

The reader is assumed to have a basic knowledge of mathematical finance and financial derivatives, and a strong interest in quantitative methods. The typical reader of the book is a “junior quant” at a financial institution who wants to gain deeper insight into numerical methods, or, if she has a background in economy, wants to take first steps towards a more quantitative approach. Alternatively, university students at the graduate level may find the topics in this book useful when deciding on a possible future career in finance.

WHAT YOU CAN EXPECT FROM THE DIFFERENT CHAPTERS

In the following, we give a short overview of the contents of the different chapters. Together with the reading guide this should allow the reader to select her topics of interest.

Chapter 2: Binomial Trees

Binomial trees a conceptionally elegant method for valuating derivatives: They are explicit (i.e., no system of equations needs to be solved), and they intrinsically include no-arbitrage

¹ The UnRisk ENGINE and the UnRisk FACTORY are software packages for valuation and risk management of financial instruments and portfolios thereof. UnRisk has been developed by MathConsult since 1999 and now contains more than 1 million lines of multi language code. UnRisk is a registered trademark of MathConsult. Details: www.unrisk.com
conditions. From a numerical point of view, their lack of adaptivity as well as stability problems limit their range of applicability.

Chapter 3: Finite Differences and the Black-Scholes PDE

In this chapter the derivation of the Black-Scholes partial differential equation is explained in detail. The differential operators are discretized using finite difference formulae, and various methods for the time discretization are introduced. Stability issues resulting from the chosen spatial and time discretizations are discussed. The application of the finite difference method to the prototype model of the heat equation concludes the chapter.

Chapter 4: Mean Reversion and Trinomial Trees

When models exhibit mean reverting behavior, such as the Vasicek model for interest rates, binomial trees do not recombine anymore. Trinomial trees have been introduced to cure this problem. To retain their stability, up- and down-branching is used, cutting off the calculation domain and therefore implicitly changing the boundary conditions.

Chapter 5: Upwinding Techniques

Particular finite difference formulae need to be applied to cure the instabilities occurring if partial differential equations arising from mean reverting models are discretized. These formulae are derived in Chapter 5 and their ability to cope with the instabilities is examined. The chapter concludes with a detailed example of the application of upwinding techniques to a putable fixed rate bond under a one factor short rate model.

Chapter 6: Boundary, Terminal and Interface Conditions

To valuate a specific financial instrument, its term sheet must be translated into boundary-, terminal- and possibly also interface conditions (for coupons or callabilities) for the differential equation to be solved. These conditions are formulated for a range of instruments. It turns out that for heavily path-dependent instruments, Monte Carlo techniques may be the more appropriate choice. For the case of mean reverting interest rate models, the influence of artificial boundary conditions is studied.

Chapter 7: Finite Element Methods

The basic concepts of the finite element method are described, and for a number of different elements, the element matrices are derived. Particular emphasis is laid on the assembling process of the global matrices and the incorporation of boundary conditions. Similarly to the finite difference technique, stabilization terms need to be added if the finite element method is applied to convection-diffusion-reaction problems. An example comparing the numerical results obtained with the finite element method to results obtained with tree techniques concludes the chapter.

Chapter 8: Solving Systems of Linear Equations

In Chapters 3, 5, 7 and 13 different discretization techniques for partial (integro) differential equations are discussed, all of them leading to systems of linear equations. This chapter provides an overview of different techniques to solve them. Dependent on the type of problem
and the problem size, direct methods or iterative solvers methods may be preferable. A number of basic algorithms for both types of methods are discussed and explained based on small examples.

Chapter 9: Monte Carlo Simulation

The principles of Monte Carlo integration techniques and their application for the pricing of derivatives are explained. Furthermore, different discretization techniques for the stochastic differential equations used to model the propagation of the underlying risk factors are discussed. The Libor market model is examined as an example of a high-dimensional model where Monte Carlo methods are typically applied to valuate derivatives. The final part of the chapter emphasizes random number generation, which is one of the major building blocks of every Monte Carlo-based pricing routine.

Chapter 10: Advanced Monte Carlo Techniques

Different techniques exist to reduce the variance of Monte Carlo estimators. Some of the more general algorithms are explained based on examples such as the pricing of exotic options. In contrast to the Monte Carlo method, Quasi Monte Carlo methods do not use sequences of pseudo-random numbers, but use sequences of low-discrepancy numbers instead. The most important sequences are introduced and applied to the pricing of a structured interest rate product under the Libor market model. As a technique to speed up Monte Carlo as well as Quasi Monte Carlo simulations the Brownian bridge method is outlined.

Chapter 11: Least Squares Monte Carlo

A well-known problem when using Monte Carlo methods for pricing is the inclusion of American or Bermudan call- or putability. We present a detailed outline of a Least Squares Monte Carlo algorithm in this chapter and apply this algorithm to a number of structured interest and foreign exchange rate instruments.

Chapter 12: Characteristic Function Methods for Option Pricing

For many distributions the density functions are not known or are not analytically tractable, but their corresponding Fourier transforms, the characteristic functions, are. This circumstance provides the basis for the methods presented in two fast and reliable methods for the valuation of European vanilla options based on the Fast Fourier Transformation and cosine series expansion are discussed. An overview of equity models beyond Black-Scholes is given in this chapter, as the calibration of these models is one of the major areas where the characteristic function methods can be applied.

Chapter 13: Numerical Methods for the Solution of PIDEs

The extension of the methods discussed in Chapters 3, 5 and 7 to cope with partial integro differential equations is discussed in this chapter.

Chapter 14: Copulas and the Pitfalls of Correlation

In the first two parts of this chapter, a number of common measures for dependence and basic concepts of copulas are presented. Subsequently, the most important copulas applied in
finance are discussed and some estimation and sampling methods for them are outlined. An example showing the impact of different copula functions on the probability of default closes the chapter.

Chapter 15: Parameter Calibration and Inverse Problems

Market data are changing more or less continuously. To reflect this fact in the valuation of financial instruments, the parameters of the mathematical models describing the movement of underlyings have to be (re)calibrated on a regular basis. This is a classical inverse problem, and therefore instabilities are to be expected. Several examples for equity and for interest rate models are discussed in detail.

Chapter 16: Optimization Techniques

To solve the calibration problems outlined in Chapter 15, optimization techniques need to be applied. We differentiate gradient-based and heuristically motivated methods, discuss algorithms for both types and show that hybrids of the two worlds can successfully be applied to estimate parameters, using the calibration of a Heston model as an example. For constrained optimization problems we introduce the interior point method and show the capability of this method in the field of portfolio optimization.

Chapter 17: Risk Management

Many of the methods discussed up to this point in the book can be used to value single instruments or portfolios. Frequently, such algorithms are used as building blocks in a risk management system where these valuations must be performed over thousands of different scenarios. In this chapter we will discuss the different possibilities to generate such scenarios and how to assess the risk measures from the simulation results. A short outline of extreme value theory and its application to the calculation of Value at Risk and Expected Shortfall concludes the chapter.

Chapter 18: Quantitative Finance on Parallel Architectures

In many fields of quantitative finance it is necessary to perform thousands or even millions of valuations of often highly structured instruments. Parallel computation techniques are used to significantly speed up these valuations. Multicore CPUs and recent GPUs, have fueled this trend by providing affordable and readily available parallel hardware. Based on examples we examine how different parallelization techniques can successfully be applied.

Chapter 19: Building Large Software Systems for the Financial Industry

This is a short chapter on the authors’ experiences in building a large software system for risk management of financial instruments.

ACCOMPANYING SOFTWARE

The buyer of the book is entitled to download the accompanying software from www.unrisk.com/Workout after registration. A tutorial on how to use the software together with the book is available on the webpage.
READING GUIDE

The diagram below shows the interrelations between different chapters and the suggested order of reading. Vertical arrows indicate suggested order of reading within a group of associated chapters, all other arrows indicate relations between different groups. For example, the Chapters on P(I)DE methods should be read in the order 3, 5–7, 13, and Chapter 8 (Solving Systems of Linear Equations) depends on material covered the P(I)DE chapters.
Binomial trees are widely used in option pricing, but are, as will be discussed later, not our preferred method. In this chapter, we will present the basic binomial tree algorithm and some of its refinements. As we want to apply binomial trees to the valuation of options, we will discuss options beforehand.

### 2.1 EQUITIES AND BASIC OPTIONS

An equity (stock, share) is a specified portion of ownership of a company, giving the holder of the share several rights, e.g., to participate in the annual general meeting and in the elections there and to obtain a dividend as a part of the profit of the company. Throughout this book, we assume that the stock is exchange-traded, so that the actual trading price of the share can be obtained (maybe with some time delay) by consulting the exchange’s homepage or a Reuters or Bloomberg screen. With the exception of an initial public offering or an increase in capital, a change in the share price does not change the firm’s capital or liquidity, but is – at least in an ideal world – the investors’ consensus on the value of the share. If the universe of investors thinks that the iPhone will be a cash cow, then the stock price of Apple will increase.

As experience during the last years showed, equity prices cannot only rise, but also can drop significantly. In order to limit the impact of downward moves of an investment at a certain point in time, a European call option may be an appropriate instrument:

- **European call and put options**: A European call option on an underlying $S$ gives the holder of the option the right, but not the obligation, to buy one share of the equity $S$ at a future time $T$ the expiry time, for a fixed price $K$, the strike price.

- A European put option on the underlying $S$ gives the holder of the option the right, but not the obligation, to sell one share of the equity $S$ at a future time $T$ the expiry time, for a fixed price $K$, the strike price.

Thus, at expiry, the payoffs of European options are as shown in Figure 2.1.

While European options may only be exercised at the date of expiry, American options may be exercised at any time during their lifetimes; Bermudan options may be exercised only at certain but known dates. Bermudan style exercise is quite popular in the fixed income world, where bonds are sometimes equipped with an early redemption right for the issuer. This early redemption typically takes place on coupon days.

Call and put options on liquid underlyings are actively traded for different strike prices and for different expiries. Of course, a (call or put) option always has a non-negative value. Holding, e.g., a call option provides an insurance against movements of the underlying below the strike price. The value of the option can therefore be interpreted as the premium to be paid for this insurance. Calculating fair values of a wide variety of options and other derivative or structured

---

1 If you google for the combination of “binomial tree” and “option pricing”, you obtain 34 200 results (date of search: November 24, 2010).
instruments, analyzing their values under different scenarios and ultimately managing their risks is one of the main purposes of computational finance as the authors understand it.

2.2 THE ONE PERIOD MODEL

Consider (at time 0) an option written on an equity that can, at the expiry $T$, assume only two possible states $s_1$ and $s_2$ and assume that the random variable $S_T$ has the distribution

$$S_T = \begin{cases} s_1 & \text{with probability } p, \\ s_2 & \text{with probability } 1 - p. \end{cases}$$ (2.1)

The payoff of the option should then, depending on the price of the underlying, have the value $V_T(s_1) = v_1$ and $V_T(s_2) = v_2$. Is it possible to construct a portfolio $P$ consisting of cash (which is assumed to pay interest at a continuous rate $r$) and shares of the equity in such a way that the portfolio replicates the option value independent of the outcome of the random process for the equity?

At time 0, let our portfolio consist of $a_1$ units of cash and of $a_2$ shares. Its value $P_0$ at time 0 is then

$$P_0 = a_1 + a_2 s_0.$$ (2.2)

At time $T$ the cash amount has increased to $a_1 e^{rT}$ whereas the equity portion’s value has changed to $a_2 s_1$ or $a_2 s_2$, respectively. If the portfolio is to replicate the option value, the unknowns $a_1, a_2$ must satisfy

$$v_1 = a_1 e^{rT} + a_2 s_1,$$
$$v_2 = a_1 e^{rT} + a_2 s_2,$$

with the solutions

$$a_2 = \frac{v_1 - v_2}{s_1 - s_2},$$
$$a_1 = e^{-rT} \left( v_1 - \frac{(v_1 - v_2)s_1}{s_1 - s_2} \right).$$

---

2 See section 4.1.1 for compounding.
Hence, if we choose the weights $a_1$ and $a_2$ accordingly, $P$ replicates the option and therefore the option value equals the portfolio value also at time 0. Hence, we obtain

$$v_0 = P_0 = s_0 \left( \frac{v_1 - v_2}{s_1 - s_2} \right) + e^{-rT} \left( v_1 - \frac{(v_1 - v_2)s_1}{s_1 - s_2} \right).$$  \hspace{1cm} (2.3)

Note that the formula for the option value does not depend on the probability $p$ of the outcomes $s_1, s_2$. We would have obtained the same result by discounting the expected outcome (at time $T$) with the expectation value taken under a probability $q$ for $s_1$ and $1 - q$ for $s_2$ with

$$q = \frac{S_0e^{rT} - s_2}{s_1 - s_2}. \hspace{1cm} (2.4)$$

The measure implied from this change of probability is called the risk neutral measure of the binomial model, whereas the physical measure is the one with the probability $p$. For a more detailed discussion concerning the measure theoretic foundation of risk neutral valuation, see, e.g., Delbaen and Schachermayer (2006). In the risk neutral measure, the expected value of the share price grows with the risk free rate $r$ independent of the physical growth rate implied by $s_1$ and $s_2$ and their physical probabilities. In order to obtain a probability $q$ in the interval $(0, 1)$, it is required that the risk-free forward value $S_0e^{rT}$ lies between $s_1$ and $s_2$. If $S_0e^{rT}$ was outside this interval, arbitrage (a guaranteed profit at a higher rate than the risk-free rate) would be possible.

### 2.3 THE MULTIPERIOD BINOMIAL MODEL

The assumption of only two possible states for the price of the equity at the expiry of the option is not a very realistic one. However at least in theory, it might be a reasonable assumption if the time interval under consideration is sufficiently small. Therefore, we recursively build an $N$-level tree as indicated in the following figure. Note that the random variables which choose the up or the down branch are assumed to be identical for all time steps. This is an essential assumption for the convergence analysis utilizing the central limit theorem.
Following this construction, each node $O$ has two successor nodes $A$ and $B$ in this multi-period binomial tree. The distribution (in the physical measure) of reaching $A$ or $B$ from $O$ is assumed to be

$$S_{nT/N} = \begin{cases} (1 + b(N))S_{(n-1)T/N} & \text{with probability } p \text{ (Point B)} \\ (1 + a(N))S_{(n-1)T/N} & \text{with probability } 1 - p \text{ (Point A)} \end{cases}$$

with the up and down factors $(1 + b(N))$ and $(1 + a(N))$, respectively, where $N$ is the number of time levels used.

In order to obtain a recombining tree in the figure, we implicitly assumed that the up and down factors are constant during the lifetime of the option. Otherwise, a bushy tree with up to $2^N$ branches would be the result. If the option value at $A$ and $B$ is known one can calculate the fair value of the option at node $O$ by a one-period binomial as shown in the previous section. The corresponding risk neutral probability is

$$q = \frac{e^{rT/N} - 1 - a(N)}{b(N) - a(N)}.$$  

(2.6)

As the option value is known for all nodes at expiry (the payoff function of the option), we can recursively calculate the values at all nodes.

As an example after some manipulation (and after taking into account the independence of branching also in the risk-free measure), we obtain for the fair value of a European call option (with strike price $K$ and an initial stock price of $S_0$)

$$V_0(N) = e^{-rT} \sum_{n=0}^{N} \binom{N}{n} q^n (1-q)^{N-n} \left( (1 + b(N))^n(1 + a(N))^{N-n} S_0 - K \right)^+,$$

(2.7)

with the interpretations $(DF)$ discount factor, $(PD)$ probability density of the binomial distribution in the risk-free measure and $(PO)$ payoff of the contingent claim, here the call option. For a real-valued argument $h$, we used the notation

$$(h)^+ := \max(h, 0).$$

(2.8)

Until now, we have not specified the dependence of $a(.)$ and $b(.)$ on the number of time levels $N$. Obviously, if the tree should not explode for $N \to \infty$, $a(N)$ and $b(N)$ must be chosen (and tend to zero) appropriately. This leads us to the Black-Scholes model.

## 2.4 BLACK-SCHOLES AND TREES

In the Black-Scholes model, the time-dependent evolution of the price of an equity $S$ starting in $S_0$ at time 0 is modeled by

$$dS_t = \mu Sdt + \sigma SdW,$$

(2.9)

with $S_t$ being the stock price at time $t$, $dS$ its incremental change in the infinitesimal time interval $(t, t + dt)$, $\sigma$ the annualized volatility, and $dW$ the increment of a standard Wiener process. The parameter $\mu$ is the expected growth rate in the physical measure.
The equation for valuating an option on an underlying equity following (2.9) will be derived in Chapter 3. There, the risk neutral measure induced by the replicating portfolios will replace \( \mu \) by the risk-free interest rate \( r \).

It can be shown that the random variable \( S_T / S_0 \) is (in the physical measure) normally distributed with mean \( (\mu - \sigma^2 / 2)T \) and variance \( \sigma^2 T \). In the risk neutral measure, \( \mu \) must be replaced by \( r \).

If the \( N \)-level binomial tree with time step \( \Delta T = (T / N) \) is to be a numerical approximation for the Black-Scholes model, the parameters \( a(N) \) and \( b(N) \) must be chosen in such a way that the corresponding distribution in the binomial model approaches the log-normal distribution of the Black-Scholes model in the risk neutral measure for \( N \to \infty \) (to be more specific: convergence in distribution will be obtained).

There are (infinitely) many ways to choose the up/down parameters \( a(.) \) and \( b(.) \). Following Korn and Müller (2010), the way to choose \( a, b \) and the physical probability \( p \) is such that mean and variance of the Black-Scholes return in the risk-free measure are matched. Note that, for the valuation process in the binomial tree, the physical probability \( p \) disappears, therefore this mingling of physical and risk-free measures is justified. Prominent examples of tree parameters are:

**Cox-Ross-Rubinstein**

\[
(1 + b) = e^{\sigma \sqrt{\Delta T}} \quad (1 + a) = e^{-\sigma \sqrt{\Delta T}} \quad (2.10)
\]

**Forward Tree**

\[
(1 + b) = e^{r \Delta T + \sigma \sqrt{\Delta T}} \quad (1 + a) = e^{r \Delta T - \sigma \sqrt{\Delta T}} \quad (2.11)
\]

Note that in the literature our “Forward Tree” is sometimes named Cox-Ross-Rubinstein.

**Rendleman-Bartter**

\[
(1 + b) = e^{(r - \sigma^2 / 2) \Delta T + \sigma \sqrt{\Delta T}} \quad (1 + a) = e^{(r - \sigma^2 / 2) \Delta T - \sigma \sqrt{\Delta T}} \quad (2.12)
\]

Depending on the preferred choice of the tree, there may be upper bounds for \( \Delta T \) in order to guarantee the no-arbitrage condition \( q \in (0, 1) \) for \( q \) as in (2.6). For the forward tree, this is always fulfilled. The Cox-Ross-Rubinstein tree must restrict \( \Delta t \) for \( r \) large and \( \sigma \) small, the Rendleman-Bartter tree for large \( \sigma \). Working out the conditions on \( \Delta T \) in detail is an easy exercise and left to the reader.

The convergence in distribution of the discrete \( N \)-level trees towards the Black-Scholes model can be shown for all the trees mentioned above. The central limit theorem is the key tool for proving convergence (Kallenberg, 2006). It can be shown that with \( N \to \infty \), the tree value \( V_0(N) \) as in (2.7) converges to

\[
V_0 = e^{-rT} \int_0^\infty \frac{1}{\sigma S \sqrt{2\pi T}} e^{-s^2/2} \left( S - K \right)^+ dS, \quad (2.13)
\]

with \( \alpha = \frac{1}{\sigma \sqrt{T}} \left( \log(S/S_0) + (r - \sigma^2 / 2)T \right) \).
2.5 STRENGTHS AND WEAKNESSES OF BINOMIAL TREES

2.5.1 Ease of Implementation

The following piece of code is an example of a forward tree realized in Mathematica. By changing the settings for the up and the down parameters, one can easily obtain a Cox-Ross-Rubinstein or a Rendleman-Bartter tree.

```mathematica
BinomialEuropeanCall2[S_, K_, r_, sigma_, T_, n_] :=
Module[{dt, forward, up, down, P, Q, Probup, Probdown, BinomTree, value, level},
dt = T/n; forward = Exp[r*dt];
up = Exp[sigma*Sqrt[dt]]*forward;
down = Exp[-sigma*Sqrt[dt]]*forward;
Probup = (forward - down)/(up - down);
Probdown = 1 - Probup;
P = Probup*Exp[-r*dt]; (* Discounting per timestep *)
Q = Probdown*Exp[-r*dt]; (* Discounting per timestep *)
BinomTree = Table[Max[S*down^node*up^(n - node) - K, 0], node, 0, n];
(* Terminal condition of a call option *)
Do[BinomTree = Table[
{P, Q}.{BinomTree[[node]], BinomTree[[node + 1]]},
{node, 1, level}],
(* In the case of an American option, compare here against the exercise value: BinomTree = Max[BinomTree, exercise value] or add knock-out conditions of barrier options here *)
{level, n, 1, -1}];
value = BinomTree[[1]];
Clear[BinomTree];
value]
```

The main advantage of the binomial tree method is its ease of implementation, also for options which may be exercised early like American or Bermudan ones. It is an explicit method so that no systems of equations have to be solved (see also Chapter 3 for a comparison of explicit/implicit methods in the finite difference framework).

2.5.2 Oscillations

How fast do binomial trees converge? Figure 2.2 shows the binomial tree values of a European call option (strike price 100, expiry 1 year, current equity price 100, interest rate 0.03, volatility 0.35) for the different binomial trees described in the previous section. Valuation has been performed for $N = 2$ to 100.

We notice the oscillating behavior (between “even values” and “odd values”) for all tree implementations; the amplitude of the oscillations is typically larger for the Cox-Ross-Rubinstein tree than for the Forward and Rendleman-Bartter trees. To obtain a guaranteed relative accuracy of 0.05% for this specific example (in the sense the result stays below this error level for all larger numbers of time levels), $N$ must be greater than 437 (Cox-Ross-Rubinstein), 277 (Forward tree), 255 (Rendleman-Bartter), resulting in trees with at least 30 000 nodes.