Contents

List of Figures xix

List of Tables xxiii

Introduction 1

0.1 What Is This Book? 1
0.2 Special Features in This Book 1
0.3 Who Is This Book for and What Do You Learn? 2
0.4 Structure of This Book 2
0.5 C# Source Code 3

1 Global Overview of the Book 5

1.1 Introduction and Objectives 5
1.2 Comparing C# and C++ 5
1.3 Using This Book 6

2 C# Fundamentals 9

2.1 Introduction and Objectives 9
2.2 Background to C# 9
2.3 Value Types, Reference Types and Memory Management 10
2.4 Built-in Data Types in C# 10
2.5 Character and String Types 12
2.6 Operators 13
2.7 Console Input and Output 14
2.8 User-defined Structures 15
2.9 Mini Application: Option Pricing 16
2.10 Summary and Conclusions 21
2.11 Exercises and Projects 22

3 Classes in C# 25

3.1 Introduction and Objectives 25
3.2 The Structure of a Class: Methods and Data 25
3.3 The Keyword ‘this’ 28
| 3.4 Properties | 28 |
| 3.5 Class Variables and Class Methods | 30 |
| 3.6 Creating and Using Objects in C# | 33 |
| 3.7 Example: European Option Price and Sensitivities | 33 |
| 3.7.1 Supporting Mathematical Functions | 34 |
| 3.7.2 Black-Scholes Formula | 35 |
| 3.7.3 C# Implementation | 36 |
| 3.7.4 Examples and Applications | 39 |
| 3.8 Enumeration Types | 40 |
| 3.9 Extension Methods | 42 |
| 3.10 An Introduction to Inheritance in C# | 44 |
| 3.11 Example: Two-factor Payoff Hierarchies and Interfaces | 46 |
| 3.12 Exception Handling | 50 |
| 3.13 Summary and Conclusions | 50 |
| 3.14 Exercises and Projects | 51 |
| 4 Classes and C# Advanced Features | 53 |
| 4.1 Introduction and Objectives | 53 |
| 4.2 Interfaces | 53 |
| 4.3 Using Interfaces: Vasicek and Cox-Ingersoll-Ross (CIR) Bond and Option Pricing | 54 |
| 4.3.1 Defining Standard Interfaces | 55 |
| 4.3.2 Bond Models and Stochastic Differential Equations | 55 |
| 4.3.3 Option Pricing and the Visitor Pattern | 58 |
| 4.4 Interfaces in .NET and Some Advanced Features | 61 |
| 4.4.1 Copying Objects | 62 |
| 4.4.2 Interfaces and Properties | 63 |
| 4.4.3 Comparing Abstract Classes and Interfaces | 64 |
| 4.4.4 Explicit Interfaces | 65 |
| 4.4.5 Casting an Object to an Interface | 65 |
| 4.5 Combining Interfaces, Inheritance and Composition | 67 |
| 4.5.1 Design Philosophy: Modular Programming | 67 |
| 4.5.2 A Model Problem and Interfacing | 68 |
| 4.5.3 Implementing the Interfaces | 69 |
| 4.5.4 Examples and Testing | 72 |
| 4.6 Introduction to Delegates and Lambda Functions | 72 |
| 4.6.1 Comparing Delegates and Interfaces | 74 |
| 4.7 Lambda Functions and Anonymous Methods | 76 |
| 4.8 Other Features in C# | 77 |
| 4.8.1 Static Constructors | 77 |
| 4.8.2 Finalisers | 78 |
| 4.8.3 Casting | 79 |
| 4.8.4 The var Keyword | 80 |
| 4.9 Advanced .NET Delegates | 80 |
| 4.9.1 Provides and Requires Interfaces: Creating Plug-in Methods with Delegates | 82 |
| 4.9.2 Multicast Delegates | 85 |
5 Data Structures and Collections

5.1 Introduction and Objectives 97
5.2 Arrays 97
5.2.1 Rectangular and Jagged Arrays 98
5.2.2 Bounds Checking 101
5.3 Dates, Times and Time Zones 101
5.3.1 Creating and Modifying Dates 101
5.3.2 Formatting and Parsing Dates 103
5.3.3 Working with Dates 104
5.4 Enumeration and Iterators 105
5.5 Object-based Collections and Standard Collection Interfaces 107
5.6 The List<T> Class 109
5.7 The Hashtable<T> Class 110
5.8 The Dictionary<Key, Value> Class 111
5.9 The HashSet<T> Classes 112
5.10 BitArray: Dynamically Sized Boolean Lists 114
5.11 Other Data Structures 114
5.11.1 Stack<T> 114
5.11.2 Queue<T> 115
5.11.3 Sorted Dictionaries 116
5.12 Strings and StringBuilder 117
5.12.1 Methods in string 118
5.12.2 Manipulating Strings 119
5.13 Some new Features in .NET 4.0 120
5.13.1 Optional Parameters 120
5.13.2 Named Parameters 121
5.13.3 COM Interoperability in .NET 4.0 121
5.13.4 Dynamic Binding 122
5.14 Summary and Conclusions 123
5.15 Exercises and Projects 123

6 Creating User-defined Data Structures 125
6.1 Introduction and Objectives 125
6.2 Design Rationale and General Guidelines 125
6.2.1 An Introduction to C# Generics 125
6.2.2 Generic Methods and Generic Delegates 128
6.2.3 Generic Constraints 129
6.2.4 Generics, Interfaces and Inheritance 130
6.2.5 Other Remarks 130
6.3 Arrays and Matrices 131
6.4 Vectors and Numeric Matrices 135
Contents

6.5 Higher-dimensional Structures 139
6.6 Sets 140
6.7 Associative Arrays and Matrices 142
  6.7.1 Associative Arrays 142
  6.7.2 Associative Matrices 144
6.8 Standardisation: Interfaces and Constraints 145
6.9 Using Associative Arrays and Matrices to Model Lookup Tables 152
6.10 Tuples 155
6.11 Summary and Conclusions 156
6.12 Exercises and Projects 156

7 An Introduction to Bonds and Bond Pricing 159
  7.1 Introduction and Objectives 159
  7.2 Embedded Optionality 160
  7.3 The Time Value of Money: Fundamentals 160
    7.3.1 A Simple Bond Class 164
    7.3.2 Testing the Bond Functionality 165
  7.4 Measuring Yield 166
  7.5 Macauley Duration and Convexity 167
  7.6 Dates and Date Schedulers for Fixed Income Applications 168
    7.6.1 Accrued Interest Calculations and Day Count Conventions 169
    7.6.2 C# Classes for Dates 170
    7.6.3 DateSchedule Class 174
  7.7 Exporting Schedulers to Excel 176
  7.8 Other Examples 177
  7.9 Pricing Bonds: An Extended Design 178
  7.10 Summary and Conclusions 181
  7.10.1 Appendix: Risks Associated with Bonds 181
  7.11 Exercises and Projects 181

8 Data Management and Data Lifecycle 185
  8.1 Introduction and Objectives 185
  8.2 Data Lifecycle in Trading Applications 185
    8.2.1 Configuration Data and Calculated Data 186
    8.2.2 Which Kinds of Data Storage Devices Can We Use? 186
  8.3 An Introduction to Streams and I/O 186
    8.3.1 Stream Architecture 186
    8.3.2 Backing Store Streams Functionality 187
    8.3.3 Stream Decorators 189
    8.3.4 Stream Adapters 191
  8.4 File and Directory Classes 195
    8.4.1 The Class Hierarchy 196
    8.4.2 FileInfo and DirectoryInfo Classes 198
  8.5 Serialisation Engines in .NET 199
    8.5.1 DataContractSerializer 199
    8.5.2 NetDataContractSerializer 201
8.5.3 Formatters 201
8.5.4 Implicit and Explicit Serialisation 203
8.6 The Binary Serialiser 203
8.7 XML Serialisation 204
  8.7.1 Subclasses and Child Objects 205
  8.7.2 Serialisation of Collections 206
  8.7.3 The IXmlSerializable Interface 207
8.8 Data Lifetime Management in Financial and Trading Applications 209
8.9 Summary and Conclusions 213
8.10 Exercises and Projects 213

9 Binomial Method, Design Patterns and Excel Output 215
  9.1 Introduction and Objectives 215
  9.2 Design of Binomial Method 216
  9.3 Design Patterns and Classes 217
  9.3.1 Creating Input Data: Factory Method Pattern 217
  9.3.2 Binomial Parameters and the Strategy Pattern 219
  9.3.3 The Complete Application Object and the Mediator Pattern 228
  9.3.4 Lattice Presentation in Excel 230
  9.4 Early Exercise Features 232
  9.5 Computing Hedge Sensitivities 233
  9.6 Multi-dimensional Binomial Method 233
  9.7 Improving Performance Using Padé Rational Approximants 236
  9.8 Summary and Conclusions 238
  9.9 Projects and Exercises 238

10 Advanced Lattices and Finite Difference Methods 241
  10.1 Introduction and Objectives 241
  10.2 Trinomial Model of the Asset Price and Its C# Implementation 241
  10.3 Stability and Convergence of the Trinomial Method 246
  10.4 The Black-Scholes Partial Differential Equation and Explicit Schemes 246
  10.5 Implementing Explicit Schemes in C# 247
    10.5.1 Using the Explicit Finite Difference Method 251
  10.6 Stability of the Explicit Finite Difference Scheme 252
  10.7 An Introduction to the Alternating Direction Explicit Method (ADE) 255
    10.7.1 ADE in a Nutshell: The One-factor Diffusion Equation 255
    10.7.2 ADE for Equity Pricing Problems 256
  10.8 Implementing ADE for the Black-Scholes PDE 258
  10.9 Testing the ADE Method 262
  10.10 Advantages of the ADE Method 263
  10.11 Summary and Conclusions 263
  10.12 Appendix: ADE Numerical Experiments 263
  10.13 Exercises and Projects 268

11 Interoperability: Namespaces, Assemblies and C++/CLI 271
  11.1 Introduction and Objectives 271
11.2 Namespaces
  11.2.1 Applications of Namespaces

11.3 An Introduction to Assemblies
  11.3.1 Assembly Types
  11.3.2 Specifying Assembly Attributes in AssemblyInfo.cs
  11.3.3 The Relationship between Namespaces and Assemblies

11.4 Reflection and Metadata
  11.4.1 Other Classes in the Reflection Namespace
  11.4.2 Dynamic Method Invocation
  11.4.3 Dynamic Object Creation
  11.4.4 Dynamic Assembly Loading
  11.4.5 Attributes and Reflection
  11.4.6 Custom Attributes

11.5 C# and Native C++ Interoperability: How Is That Possible?
  11.5.1 Using Native C++ from C#

11.6 Using C# from C++

11.7 Code Generation Using the Reflection API
  11.7.1 The DynamicMethod Class
  11.7.2 The Evaluation Stack and Argument Passing to
dynamic Methods
  11.7.3 The Case in Hand: Operator Overloading for Generic
Vectors and Matrices

11.8 Application Domains
  11.8.1 Creating and Destroying Application Domains
  11.8.2 Multiple Application Domains
  11.8.3 Sharing Data between Domains
  11.8.4 When to Use Application Domains

11.9 Summary and Conclusions
11.10 Exercises and Projects

12 Bond Pricing: Design, Implementation and Excel Interfacing
  12.1 Introduction and Objectives
  12.2 High-level Design of Bond Pricing Problem
  12.3 Bond Scheduling
  12.4 Bond Functionality and Class Hierarchies
  12.5 Calculating Price, Yield and Discount Factors: MathTools
  12.6 Data Presentation and Excel Interop
  12.7 Bond Data Management
    12.7.1 Data into Memory
    12.7.2 Serialisation and Deserialisation
  12.8 Using the Excel Files
  12.9 Summary and Conclusions
12.10 Exercises and Projects
  1 Code Integration: Handling Bond Details
  2 Spread on Benchmark
  3 Floating Rate Bond and Other Structured Notes
  4 Class Hierarchy Integration
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>Interpolation Methods in Interest Rate Applications</td>
<td>335</td>
</tr>
<tr>
<td>13.1</td>
<td>Introduction and Objectives</td>
<td>335</td>
</tr>
<tr>
<td>13.2</td>
<td>Interpolation and Curve Building: Basic Formula for Interpolator Tests</td>
<td>335</td>
</tr>
<tr>
<td>13.3</td>
<td>Types of Curve Shape</td>
<td>337</td>
</tr>
<tr>
<td>13.4</td>
<td>An Overview of Interpolators</td>
<td>338</td>
</tr>
<tr>
<td>13.5</td>
<td>Background to Interpolation</td>
<td>339</td>
</tr>
<tr>
<td>13.6</td>
<td>Approximation of Function Derivatives</td>
<td>341</td>
</tr>
<tr>
<td>13.7</td>
<td>Linear and Cubic Spline Interpolation</td>
<td>342</td>
</tr>
<tr>
<td>13.8</td>
<td>Positivity-preserving Cubic Interpolations: Dougherty/Hyman and Hussein</td>
<td>344</td>
</tr>
<tr>
<td>13.9</td>
<td>The Akima Method</td>
<td>348</td>
</tr>
<tr>
<td>13.10</td>
<td>Hagan-West Approach</td>
<td>349</td>
</tr>
<tr>
<td>13.11</td>
<td>Global Interpolation</td>
<td>350</td>
</tr>
<tr>
<td>13.11.1</td>
<td>Polynomial Interpolation</td>
<td>351</td>
</tr>
<tr>
<td>13.11.2</td>
<td>Rational Interpolation</td>
<td>352</td>
</tr>
<tr>
<td>13.12</td>
<td>Bilinear Interpolation</td>
<td>352</td>
</tr>
<tr>
<td>13.13</td>
<td>Some General Guidelines, Hints and Tips</td>
<td>355</td>
</tr>
<tr>
<td>13.14</td>
<td>Using the Interpolators and Test Examples</td>
<td>357</td>
</tr>
<tr>
<td>13.14.1</td>
<td>The 101 Example, from A to Z</td>
<td>357</td>
</tr>
<tr>
<td>13.14.2</td>
<td>Some Financial Formulae</td>
<td>360</td>
</tr>
<tr>
<td>13.14.3</td>
<td>Cubic Spline Interpolation: an Application Example</td>
<td>361</td>
</tr>
<tr>
<td>13.14.4</td>
<td>A Bilinear Interpolation Simple Example</td>
<td>364</td>
</tr>
<tr>
<td>13.15</td>
<td>Summary and Conclusions</td>
<td>367</td>
</tr>
<tr>
<td>13.16</td>
<td>Exercises and Projects</td>
<td>367</td>
</tr>
<tr>
<td>14</td>
<td>Short Term Interest Rate (STIR) Futures and Options</td>
<td>369</td>
</tr>
<tr>
<td>14.1</td>
<td>Introduction and Objectives</td>
<td>369</td>
</tr>
<tr>
<td>14.2</td>
<td>An Overview of Cash Money Markets</td>
<td>370</td>
</tr>
<tr>
<td>14.3</td>
<td>Sources of Risk in Money Market Transactions</td>
<td>370</td>
</tr>
<tr>
<td>14.4</td>
<td>Reference Rate and Fixings</td>
<td>371</td>
</tr>
<tr>
<td>14.5</td>
<td>STIR Futures</td>
<td>371</td>
</tr>
<tr>
<td>14.6</td>
<td>Pricing STIR Options</td>
<td>374</td>
</tr>
<tr>
<td>14.7</td>
<td>Generating International Monetary Market (IMM) Dates</td>
<td>378</td>
</tr>
<tr>
<td>14.7.1</td>
<td>Modelling Option Delta and Sensitivity Analysis</td>
<td>380</td>
</tr>
<tr>
<td>14.7.2</td>
<td>Listed Instruments and Contracts</td>
<td>383</td>
</tr>
<tr>
<td>14.8</td>
<td>List STIR Futures and STIR Futures Options</td>
<td>384</td>
</tr>
<tr>
<td>14.9</td>
<td>Putting It All Together: STIR versus OTC from a Trader’s Perspective</td>
<td>387</td>
</tr>
<tr>
<td>14.10</td>
<td>Summary and Conclusions</td>
<td>389</td>
</tr>
<tr>
<td>14.11</td>
<td>Exercises and Projects</td>
<td>389</td>
</tr>
<tr>
<td>15</td>
<td>Single-curve Building</td>
<td>393</td>
</tr>
<tr>
<td>15.1</td>
<td>Introduction and Objectives</td>
<td>393</td>
</tr>
<tr>
<td>15.2</td>
<td>Starting Definitions and Overview of Curve Building Process</td>
<td>393</td>
</tr>
<tr>
<td>15.3</td>
<td>Building Blocks</td>
<td>395</td>
</tr>
<tr>
<td>15.3.1</td>
<td>Unsecured Deposit</td>
<td>395</td>
</tr>
<tr>
<td>15.3.2</td>
<td>Forward Rate Agreements (FRA)</td>
<td>396</td>
</tr>
<tr>
<td>15.3.3</td>
<td>Future Implied Rate</td>
<td>397</td>
</tr>
<tr>
<td>15.3.4</td>
<td>Interest Rate Swap (IRS)</td>
<td>397</td>
</tr>
</tbody>
</table>
15.4 Introduction to Interest Rate Swap

15.4.1 IRS Cash Flow
15.4.2 The Use of Interest Rate Swaps
15.4.3 Contract Specification and Practical Aspects
15.4.4 Traditional Swap Valuation
15.4.5 Overnight Index Swap (OIS)

15.5 The Curve Construction Mechanism

15.5.1 Traditional Bootstrapping Method
15.5.2 Best Fit Method
15.5.3 The Key Role of Interpolation

15.6 Code Design and Implementation

15.6.1 Process Design
15.6.2 ISingleRateCurve Interface
15.6.3 RateSet Class and BuildingBlock Class
15.6.4 Interpolator and Adapters
15.6.5 The Generic Base Class SingleCurveBuilder
15.6.6 Derived Class for Traditional Bootstrapping Method
15.6.7 Derived Class for Global Method with Interpolation
15.6.8 Derived Class for Global Method with Smoothness Condition

15.7 Console Examples

15.7.1 Calculating Present Value (PV) of the Floating Leg of a Swap
15.7.2 Checking If the Curve is Calibrated
15.7.3 Calculate the Time Taken to Instantiate a SingleCurveBuilder
15.7.4 Visualise Forward Rates in Excel
15.7.5 Computing Forward Start Swap
15.7.6 Computing Sensitivities: An Initial Example
15.7.7 More on Sensitivities

15.8 Summary and Conclusions

15.9 Exercises and Projects

15.10 Appendix: Types of Swaps

16 Multi-curve Building

16.1 Introduction and Objectives

16.2 The Consequences of the Crisis on Interest Rate Derivatives Valuation

16.2.1 The Growing Importance of Overnight Indexed Swap
16.2.2 Collateralisation under a CSA
16.2.3 The Role of OIS Discounting: One Curve Is Not Enough
16.2.4 Basis
16.2.5 The Par Swap Rate Formulae

16.3 Impact of Using OIS Discounting

16.3.1 Effect on Forward Rates
16.3.2 Effect on Mark-to-Market
16.3.3 Risk Effect

16.4 The Bootstrapping Process Using Two Curves: Description of the Mechanism
16.5 Sensitivities
16.6 How to Organise the Code: A Possible Solution
   16.6.1 IRateCurve Base Interface and Derived Interfaces
   16.6.2 The class MultiCurveBuilder
16.7 Putting it Together, Working Examples
   16.7.1 Calibration Consistency
   16.7.2 Print Forward Rates and Discount Factors on Excel
   16.7.3 Sensitivities on Console
   16.7.4 Forward Swap Matrix
   16.7.5 Mark-to-Market Differences
   16.7.6 Comparing Two Versions of the MultiCurveBuilder
   16.7.7 Input Data, Interpolation and Forward Rates
   16.7.8 Comparing Discount Factor
16.8 Summary and Conclusions
16.9 Exercises and Projects
16.10 Appendix: Par Asset Swap Spread and Zero Volatility Spread

17 Swaption, Cap and Floor
   17.1 Introduction and Objectives: A Closed Formula World
   17.2 Description of Instruments and Formulae
      17.2.1 Cap and Floor: Description and Formulae
      17.2.2 Cap and Floor at the money Strike
      17.2.3 Cap Volatility and Caplet Volatility
      17.2.4 Implied Volatility
      17.2.5 Multi-strike and Amortising Cap and Floor
      17.2.6 Swaption: Mechanism and Closed Pricing Formulae
      17.2.7 Call Put Parity for Cap, Floor and Swaption
17.3 Multi-curve Framework on Cap, Floor and Swaption
17.4 Bootstrapping Volatility for Cap and Floor
   17.4.1 Cap Stripping
   17.4.2 Missing Data, Volatility Models and Interpolation
17.5 How to Organise the Code in C#: A Possible Solution
   17.5.1 Ready to Use Formula
   17.5.2 Cap Stripping Code
   17.5.3 Calculating Mono-strike Caplet Volatilities
   17.5.4 Managing More Mono-strike Caplet Volatilities
17.6 Console and Excel Working Examples
   17.6.1 Simple Caplet Price
   17.6.2 Cap As a Sum of Caplets
   17.6.3 Simple Cap Volatility Bootstrapping: First Unknown Volatility
   17.6.4 ATM Strike and Recursive Bootstrapping
   17.6.5 Sparse Data from the Market: Volatility Optimisation and Input Interpolation
17.7 Summary and Conclusions
17.8 Exercise and Discussion
Contents

19.8 Examples in Fixed Income Applications 540
  19.8.1 Using Conversion Operators 540
  19.8.2 Discount Factors 540
  19.8.3 Bonds 542
  19.8.4 Scenarios 543
  19.8.5 Cash Flow Aggregation 545
  19.8.6 Ordering Collections 546
  19.8.7 Eonia Rates Replication 547
19.9 LINQ and Excel Interoperability 549
  19.9.1 Applications in Computational Finance 557
19.10 Summary and Conclusions 557
19.11 Exercises and Projects 557

20 Introduction to C# and Excel Integration 561
  20.1 Introduction and Objectives 561
  20.2 Excel Object Model 561
  20.3 Using COM Technology in .NET 561
  20.4 Primary Interop Assemblies (PIA) 563
  20.5 Standalone Applications 564
    20.5.1 Standalone Application: Workbook and Worksheets 564
    20.5.2 Charts 565
    20.5.3 Using Excel with C++/CLI 565
  20.6 Types of Excel Add-ins 566
    20.6.1 XLL 567
    20.6.2 XLA 567
    20.6.3 COM 567
    20.6.4 Automation 567
    20.6.5 VSTO 568
  20.7 The IDTExtensibility2 Interface and COM/.NET Interoperability 569
  20.8 Data Visualisation in Excel 570
    20.8.1 Excel Driver 570
    20.8.2 Data Structures 572
    20.8.3 ExcelMechanisms and Exception Handling 572
    20.8.4 Examples and Applications 575
  20.9 Conclusion and Summary 578
  20.10 Exercises and Projects 579

21 Excel Automation Add-ins 581
  21.1 Introduction and Objectives 581
  21.2 COM Overview 581
  21.3 Creating Automation Add-ins: The Steps 583
  21.4 Example: Creating a Calculator, Version 1 585
  21.5 Example: Creating a Calculator, Version 2 588
  21.6 Versioning 590
  21.7 Working with Ranges 590
  21.8 Volatile Methods 590
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>21.9</td>
<td>Optional Parameters</td>
</tr>
<tr>
<td>21.10</td>
<td>Using VBA with Automation Add-ins</td>
</tr>
<tr>
<td>21.11</td>
<td>Summary and Conclusions</td>
</tr>
<tr>
<td>21.12</td>
<td>Exercises and Projects</td>
</tr>
<tr>
<td>22</td>
<td>C# and Excel Integration COM Add-ins</td>
</tr>
<tr>
<td>22.1</td>
<td>Introduction and Objectives</td>
</tr>
<tr>
<td>22.2</td>
<td>Preparations for COM Add-ins</td>
</tr>
<tr>
<td>22.3</td>
<td>The Interface <code>IDTExtensibility2</code></td>
</tr>
<tr>
<td>22.4</td>
<td>Creating COM Add-ins: The Steps</td>
</tr>
<tr>
<td>22.5</td>
<td>Utility Code and Classes</td>
</tr>
<tr>
<td>22.6</td>
<td>Using Windows Forms</td>
</tr>
<tr>
<td>22.7</td>
<td>Example: Creating a COM Add-in</td>
</tr>
<tr>
<td>22.8</td>
<td>Debugging and Troubleshooting</td>
</tr>
<tr>
<td>22.9</td>
<td>An Introduction to Excel-DNA</td>
</tr>
<tr>
<td>22.9.1</td>
<td>Example 001: Hello World</td>
</tr>
<tr>
<td>22.9.2</td>
<td>Example 101: Simple Option Pricer</td>
</tr>
<tr>
<td>22.9.3</td>
<td>Excel-DNA and Rate Curves</td>
</tr>
<tr>
<td>22.9.4</td>
<td>Registration and Loading</td>
</tr>
<tr>
<td>22.9.5</td>
<td>What Is Inside <code>ExcelDNA.Integration.dll</code>?</td>
</tr>
<tr>
<td>22.10</td>
<td>Excel COM Interoperability and Rate Multi-curve</td>
</tr>
<tr>
<td>22.11</td>
<td>Conclusion and Summary</td>
</tr>
<tr>
<td>22.12</td>
<td>Exercises and Projects</td>
</tr>
<tr>
<td>23</td>
<td>Real-time Data (RTD) Server</td>
</tr>
<tr>
<td>23.1</td>
<td>Introduction and Objectives</td>
</tr>
<tr>
<td>23.2</td>
<td>Real-time Data in Excel: Overview</td>
</tr>
<tr>
<td>23.3</td>
<td>Real-time Data Function</td>
</tr>
<tr>
<td>23.4</td>
<td>Example</td>
</tr>
<tr>
<td>23.5</td>
<td>The Topic Class and Data</td>
</tr>
<tr>
<td>23.6</td>
<td>Creating an RTD Server</td>
</tr>
<tr>
<td>23.7</td>
<td>Using the RTD Server</td>
</tr>
<tr>
<td>23.8</td>
<td>Testing and Troubleshooting the RTD Server</td>
</tr>
<tr>
<td>23.9</td>
<td>Conclusion and Summary</td>
</tr>
<tr>
<td>23.10</td>
<td>Exercises and Projects</td>
</tr>
<tr>
<td>24</td>
<td>Introduction to Multi-threading in C#</td>
</tr>
<tr>
<td>24.1</td>
<td>Introduction and Objectives</td>
</tr>
<tr>
<td>24.2</td>
<td>Processes</td>
</tr>
<tr>
<td>24.3</td>
<td>Using <code>ProcessStartInfo</code> to Redirect Process I/O</td>
</tr>
<tr>
<td>24.4</td>
<td>An Introduction to Threads in C#</td>
</tr>
<tr>
<td>24.4.1</td>
<td>The Differences between Processes and Threads</td>
</tr>
<tr>
<td>24.5</td>
<td>Passing Data to a Thread and between Threads</td>
</tr>
<tr>
<td>24.6</td>
<td>Thread States and Thread Lifecycle</td>
</tr>
<tr>
<td>24.6.1</td>
<td>Sleep</td>
</tr>
<tr>
<td>24.6.2</td>
<td>Thread Joining</td>
</tr>
<tr>
<td>24.6.3</td>
<td>Thread Interrupt and Abort</td>
</tr>
<tr>
<td>24.7</td>
<td>Thread Priority</td>
</tr>
</tbody>
</table>
24.8 Thread Pooling 651
24.9 Atomic Operations and the Interlocked Class 652
24.10 Exception Handling 653
24.11 Multi-threaded Data Structures 654
24.11.1 Extended Producer–Consumer Pattern 657
24.12 A Simple Example of Traditional Multi-threading 659
24.13 Summary and Conclusions 661
24.14 Exercises and Projects 661

25 Advanced Multi-threading in C# 665
25.1 Introduction and Objectives 665
25.2 Thread Safety 666
25.3 Locking Mechanisms for Objects and Classes 667
25.3.1 Locking a Class 669
25.3.2 Nested Locking 669
25.4 Mutex and Semaphore 673
25.5 Notification and Signalling 676
25.5.1 Thread Notification and the Monitor Class 678
25.6 Asynchronous Delegates 679
25.7 Synchronising Collections 681
25.8 Timers 682
25.9 Foreground and Background Threads 684
25.10 Executing Operations on Separate Threads: the BackgroundWorker Class 685
25.11 Parallel Programming in .NET 687
25.11.1 The Parallel Class 687
25.12 Task Parallel Library (TPL) 691
25.12.1 Creating and Starting Tasks 692
25.12.2 Continuations 694
25.13 Concurrent Data Structures 694
25.13.1 An Example: Producer Consumer Pattern and Random Number Generation 695
25.13.2 The Barrier Class 698
25.13.3 PLINQ 699
25.14 Exception Handling 701
25.15 Shifting Curves 702
25.16 Summary and Conclusions 704
25.17 Exercises and Projects 704

26 Creating Multi-threaded and Parallel Applications for Computational Finance 707
26.1 Introduction and Objectives 707
26.2 Multi-threaded and Parallel Applications for Computational Finance 707
26.3 Fork and Join Pattern 709
26.4 Geometric Decomposition 711
26.5 Shared Data and Reader/Writer Locks: Multiple Readers and Multiple Writers 715
26.5.1 Upgradeable Locks and Recursion 718
## Contents

26.6 Monte Carlo Option Pricing and the Producer–Consumer Pattern 719  
26.7 The **StopWatch** Class 726  
26.8 Garbage Collection and Disposal 727  
   26.8.1 Disposal and the **IDisposable** Interface 727  
   26.8.2 Automatic Garbage Collection 728  
   26.8.3 Managed Memory Leaks 730  
26.9 Summary and Conclusions 730  
26.10 Exercises and Projects 730  

### A1 Object-oriented Fundamentals 735  
A1.1 Introduction and Objectives 735  
A1.2 Object-oriented Paradigm 735  
A1.3 Generic Programming 737  
A1.4 Procedural Programming 738  
A1.5 Structural Relationships 738  
   A1.5.1 Aggregation 739  
   A1.5.2 Association 740  
   A1.5.3 Generalisation/Specialisation (Gen/Spec Relationship) 742  
A1.6 An Introduction to Concept Modelling 743  
   A1.6.1 The Defining Attribute View 743  
   A1.6.2 The Prototype View 744  
   A1.6.3 The Exemplar-based View 744  
   A1.6.4 The Explanation-based View 744  
A1.7 Categorisation and Concept Levels 745  
A1.8 Whole–Part Pattern 745  
   A1.8.1 Data Decomposition 746  
A1.9 Message-passing Concept versus Procedural Programming 748  

### A2 Nonlinear Least-squares Minimisation 751  
A2.1 Introduction and Objectives 751  
A2.2 Nonlinear Programming and Multi-variable Optimisation 751  
A2.3 Nonlinear Least Squares 753  
   A2.3.1 Nonlinear Regression 753  
   A2.3.2 Simultaneous Nonlinear Equations 754  
   A2.3.3 Derivatives of Sum-of-Squares Functions 754  
A2.4 Some Specific Methods 755  
A2.5 The ALGLIB Library 756  
A2.6 An Application to Curve Building 758  
A2.7 Rate Calibration Example 759  
A2.8 Exercises and Projects 764  

### A3 The Mathematical Background to the Alternating Direction Explicit (ADE) Method 765  
A3.1 Introduction and Objectives 765  
A3.2 Background to ADE 765  
A3.3 Scoping the Problem: One-factor Problems 766  
A3.4 An Example: One-factor Black-Scholes PDE 768
A3.5 Boundary Conditions
A3.6 Example: Boundary Conditions for the One-factor Black-Scholes PDE
A3.7 Motivating the ADE Method
A3.8 The ADE Method Exposed
A3.9 The Convection Term
A3.10 Other Kinds of Boundary Conditions
A3.11 Nonlinear Problems
A3.12 ADE for PDEs in Conservative Form
A3.13 Numerical Results and Guidelines
  A3.13.1 The Consequences of Conditional Consistency
  A3.13.2 Call Payoff Behaviour at the Far Field
  A3.13.3 General Formulation of the ADE Method
A3.14 The Steps to Use when Implementing ADE
A3.15 Summary and Conclusions
A3.16 Exercises and Projects

A4 Cap, Floor and Swaption Using Excel-DNA
A4.1 Introduction
A4.2 Different Ways of Stripping Cap Volatility
A4.3 Comparing Caplet Volatility Surface
A4.4 Call Put Parity
A4.5 Cap Price Matrix
A4.6 Multi-strike and Amortising
A4.7 Simple Swaption Formula
A4.8 Swaption Straddle
A4.9 Exercises

Bibliography

Web References

Index