

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 Structs	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 <code>'this'</code>	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 <i>Visitor</i> 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	<i>Provides</i> and <i>Requires</i> Interfaces: Creating Plug-in Methods with Delegates	82
4.9.2	Multicast Delegates	85

4.9.3	Generic Delegate Types	86
4.9.4	Delegates versus Interfaces, Again	87
4.10	The Standard Event Pattern in .NET and the Observer Pattern	87
4.11	Summary and Conclusions	91
4.12	Exercises and Projects	92
5	Data Structures and Collections	97
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

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 <code>IXmlSerializable</code> 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 <i>Strategy</i> 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	271
11.2.1	Applications of Namespaces	272
11.3	An Introduction to Assemblies	273
11.3.1	Assembly Types	274
11.3.2	Specifying Assembly Attributes in AssemblyInfo.cs	275
11.3.3	The Relationship between Namespaces and Assemblies	276
11.4	Reflection and Metadata	276
11.4.1	Other Classes in the Reflection Namespace	281
11.4.2	Dynamic Method Invocation	283
11.4.3	Dynamic Object Creation	283
11.4.4	Dynamic Assembly Loading	284
11.4.5	Attributes and Reflection	284
11.4.6	Custom Attributes	286
11.5	C# and Native C++ Interoperability: How Is That Possible?	289
11.5.1	Using Native C++ from C#	289
11.6	Using C# from C++	293
11.7	Code Generation Using the Reflection API	298
11.7.1	The DynamicMethod Class	299
11.7.2	The Evaluation Stack and Argument Passing to Dynamic Methods	300
11.7.3	The Case in Hand: Operator Overloading for Generic Vectors and Matrices	301
11.8	Application Domains	304
11.8.1	Creating and Destroying Application Domains	304
11.8.2	Multiple Application Domains	305
11.8.3	Sharing Data between Domains	307
11.8.4	When to Use Application Domains	308
11.9	Summary and Conclusions	309
11.10	Exercises and Projects	309
12	Bond Pricing: Design, Implementation and Excel Interfacing	311
12.1	Introduction and Objectives	311
12.2	High-level Design of Bond Pricing Problem	311
12.3	Bond Scheduling	312
12.4	Bond Functionality and Class Hierarchies	313
12.5	Calculating Price, Yield and Discount Factors: MathTools	317
12.6	Data Presentation and Excel Interop	319
12.7	Bond Data Management	321
12.7.1	Data into Memory	321
12.7.2	Serialisation and Deserialisation	322
12.8	Using the Excel Files	324
12.9	Summary and Conclusions	328
12.10	Exercises and Projects	328
1	Code Integration: Handling Bond Details	328
2	Spread on Benchmark	330
3	Floating Rate Bond and Other Structured Notes	331
4	Class Hierarchy Integration	333

13	Interpolation Methods in Interest Rate Applications	335
13.1	Introduction and Objectives	335
13.2	Interpolation and Curve Building: Basic Formula for Interpolator Tests	335
13.3	Types of Curve Shape	337
13.4	An Overview of Interpolators	338
13.5	Background to Interpolation	339
13.6	Approximation of Function Derivatives	341
13.7	Linear and Cubic Spline Interpolation	342
13.8	Positivity-preserving Cubic Interpolations: Dougherty/Hyman and Hussein	344
13.9	The Akima Method	348
13.10	Hagan-West Approach	349
13.11	Global Interpolation	350
13.11.1	Polynomial Interpolation	351
13.11.2	Rational Interpolation	352
13.12	Bilinear Interpolation	352
13.13	Some General Guidelines, Hints and Tips	355
13.14	Using the Interpolators and Test Examples	357
13.14.1	The 101 Example, from A to Z	357
13.14.2	Some Financial Formulae	360
13.14.3	Cubic Spline Interpolation: an Application Example	361
13.14.4	A Bilinear Interpolation: Simple Example	364
13.15	Summary and Conclusions	367
13.16	Exercises and Projects	367
14	Short Term Interest Rate (STIR) Futures and Options	369
14.1	Introduction and Objectives	369
14.2	An Overview of Cash Money Markets	370
14.3	Sources of Risk in Money Market Transactions	370
14.4	Reference Rate and Fixings	371
14.5	STIR Futures	371
14.6	Pricing STIR Options	374
14.7	Generating International Monetary Market (IMM) Dates	378
14.7.1	Modelling Option Delta and Sensitivity Analysis	380
14.7.2	Listed Instruments and Contracts	383
14.8	List STIR Futures and STIR Futures Options	384
14.9	Putting It All Together: STIR versus OTC from a Trader's Perspective	387
14.10	Summary and Conclusions	389
14.11	Exercises and Projects	389
15	Single-curve Building	393
15.1	Introduction and Objectives	393
15.2	Starting Definitions and Overview of Curve Building Process	393
15.3	Building Blocks	395
15.3.1	Unsecured Deposit	395
15.3.2	Forward Rate Agreements (FRA)	396
15.3.3	Future Implied Rate	397
15.3.4	Interest Rate Swap (IRS)	397

15.4	Introduction to Interest Rate Swap	397
15.4.1	IRS Cash Flow	398
15.4.2	The Use of Interest Rate Swaps	399
15.4.3	Contract Specification and Practical Aspects	399
15.4.4	Traditional Swap Valuation	402
15.4.5	Overnight Index Swap (OIS)	403
15.5	The Curve Construction Mechanism	403
15.5.1	Traditional Bootstrapping Method	404
15.5.2	Best Fit Method	405
15.5.3	The Key Role of Interpolation	405
15.6	Code Design and Implementation	406
15.6.1	Process Design	406
15.6.2	ISingleRateCurve Interface	406
15.6.3	RateSet Class and BuildingBlock Class	407
15.6.4	Interpolator and Adapters	409
15.6.5	The Generic Base Class SingleCurveBuilder	410
15.6.6	Derived Class for Traditional Bootstrapping Method	412
15.6.7	Derived Class for Global Method with Interpolation	413
15.6.8	Derived Class for Global Method with Smoothness Condition	415
15.7	Console Examples	418
15.7.1	Calculating Present Value (PV) of the Floating Leg of a Swap	418
15.7.2	Checking If the Curve is Calibrated	418
15.7.3	Calculate the Time Taken to Instantiate a SingleCurveBuilder	418
15.7.4	Visualise Forward Rates in Excel	419
15.7.5	Computing Forward Start Swap	421
15.7.6	Computing Sensitivities: An Initial Example	421
15.7.7	More on Sensitivities	422
15.8	Summary and Conclusions	426
15.9	Exercises and Projects	427
15.10	Appendix: Types of Swaps	429
16	Multi-curve Building	431
16.1	Introduction and Objectives	431
16.2	The Consequences of the Crisis on Interest Rate Derivatives Valuation	431
16.2.1	The Growing Importance of Overnight Indexed Swap	432
16.2.2	Collateralisation under a CSA	432
16.2.3	The Role of OIS Discounting: One Curve Is Not Enough	433
16.2.4	Basis	433
16.2.5	The Par Swap Rate Formulae	434
16.3	Impact of Using OIS Discounting	436
16.3.1	Effect on Forward Rates	436
16.3.2	Effect on Mark-to-Market	436
16.3.3	Risk Effect	436
16.4	The Bootstrapping Process Using Two Curves: Description of the Mechanism	437

16.5	Sensitivities	438
16.6	How to Organise the Code: A Possible Solution	439
16.6.1	IRateCurve Base Interface and Derived Interfaces	439
16.6.2	The class MultiCurveBuilder	441
16.7	Putting it Together, Working Examples	445
16.7.1	Calibration Consistency	445
16.7.2	Print Forward Rates and Discount Factors on Excel	446
16.7.3	Sensitivities on Console	446
16.7.4	Forward Swap Matrix	447
16.7.5	Mark-to-Market Differences	448
16.7.6	Comparing Two Versions of the MultiCurveBuilder	450
16.7.7	Input Data, Interpolation and Forward Rates	452
16.7.8	Comparing Discount Factor	453
16.8	Summary and Conclusions	453
16.9	Exercises and Projects	453
16.10	Appendix: Par Asset Swap Spread and Zero Volatility Spread	455
17	Swaption, Cap and Floor	459
17.1	Introduction and Objectives: A Closed Formula World	459
17.2	Description of Instruments and Formulae	459
17.2.1	Cap and Floor: Description and Formulae	459
17.2.2	Cap and Floor <i>at the money</i> Strike	461
17.2.3	Cap Volatility and Caplet Volatility	462
17.2.4	Implied Volatility	463
17.2.5	Multi-strike and Amortising Cap and Floor	463
17.2.6	Swaption: Mechanism and Closed Pricing Formulae	464
17.2.7	Call Put Parity for Cap, Floor and Swaption	466
17.3	Multi-curve Framework on Cap, Floor and Swaption	467
17.4	Bootstrapping Volatility for Cap and Floor	469
17.4.1	Cap Stripping	469
17.4.2	Missing Data, Volatility Models and Interpolation	473
17.5	How to Organise the Code in C#: A Possible Solution	474
17.5.1	Ready to Use Formula	474
17.5.2	Cap Stripping Code	476
17.5.3	Calculating Mono-strike Caplet Volatilities	476
17.5.4	Managing More Mono-strike Caplet Volatilities	479
17.6	Console and Excel Working Examples	481
17.6.1	Simple Caplet Price	481
17.6.2	Cap As a Sum of Caplets	482
17.6.3	Simple Cap Volatility Bootstrapping: First Unknown Volatility	483
17.6.4	ATM Strike and Recursive Bootstrapping	485
17.6.5	Sparse Data from the Market: Volatility Optimisation and Input Interpolation	487
17.7	Summary and Conclusions	490
17.8	Exercise and Discussion	491

18	Software Architectures and Patterns for Pricing Applications	493
18.1	Introduction and Objectives	493
18.2	An Overview of the GOF Pattern	494
18.3	Creational Patterns	496
18.4	Builder Pattern	496
18.5	Structural Patterns	499
18.5.1	Facade Pattern	499
18.5.2	Layers Pattern	499
18.6	Behavioural Patterns	500
18.6.1	<i>Visitor</i> Pattern	501
18.6.2	Strategy and Template Method Patterns	501
18.7	Builder Application Example: Calibration Algorithms for Cap and Floor	502
18.7.1	Example Caplet Volatility Matrix	502
18.7.2	Volatility Matrix with Multiple Strikes	503
18.8	A PDE/FDM Patterns-based Framework for Equity Options	504
18.8.1	High-level Design	506
18.8.2	Generalisations and Extensions	508
18.9	Using Delegates to Implement Behavioural Design Patterns	509
18.10	A System Design for Monte Carlo Applications	510
18.10.1	A Universal System Design Methodology	511
18.11	Dynamic Programming in .NET	513
18.11.1	Numeric Type Unification	514
18.11.2	Implementing Dynamic Objects	516
18.12	Summary and Conclusions	516
18.13	Exercises and Projects	517
19	LINQ (Language Integrated Query) and Fixed Income Applications	523
19.1	Introduction and Objectives	523
19.2	Scope of Chapter and Prerequisites	523
19.3	LINQ Query Operators	524
19.3.1	Collection as Input, Collection as Output	524
19.3.2	Collection as Input, Noncollection as Output	525
19.3.3	Noncollection to Collection	526
19.4	LINQ Queries and Initial Examples	526
19.4.1	Lambda Queries and Composition	527
19.4.2	Comprehension Queries	528
19.4.3	Deferred Execution	529
19.5	Advanced Queries	531
19.5.1	Subqueries	531
19.5.2	Composition Strategies	532
19.5.3	Interpreted Queries	533
19.6	A Numerical Example	533
19.6.1	Basic Functionality	533
19.6.2	User-defined Aggregation Methods	534
19.6.3	Set Operations	535
19.7	Join and GroupJoin	535

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

21.9	Optional Parameters	591
21.10	Using VBA with Automation Add-ins	592
21.11	Summary and Conclusions	593
21.12	Exercises and Projects	594
22	C# and Excel Integration COM Add-ins	595
22.1	Introduction and Objectives	595
22.2	Preparations for COM Add-ins	595
22.3	The Interface IDTExtensibility2	596
22.4	Creating COM Add-ins: The Steps	596
22.5	Utility Code and Classes	597
22.6	Using Windows Forms	600
22.7	Example: Creating a COM Add-in	601
22.8	Debugging and Troubleshooting	603
22.9	An Introduction to Excel-DNA	603
22.9.1	Example 001: Hello World	604
22.9.2	Example 101: Simple Option Pricer	605
22.9.3	Excel-DNA and Rate Curves	608
22.9.4	Registration and Loading	613
22.9.5	What Is Inside ExcelDna.Integration.dll ?	614
22.10	Excel COM Interoperability and Rate Multi-curve	615
22.11	Conclusion and Summary	622
22.12	Exercises and Projects	622
23	Real-time Data (RTD) Server	625
23.1	Introduction and Objectives	625
23.2	Real-time Data in Excel: Overview	625
23.3	Real-time Data Function	626
23.4	Example	627
23.5	The Topic Class and Data	629
23.6	Creating an RTD Server	631
23.7	Using the RTD Server	631
23.8	Testing and Troubleshooting the RTD Server	632
23.9	Conclusion and Summary	632
23.10	Exercises and Projects	632
24	Introduction to Multi-threading in C#	635
24.1	Introduction and Objectives	635
24.2	Processes	636
24.3	Using ProcessStartInfo to Redirect Process I/O	637
24.4	An Introduction to Threads in C#	638
24.4.1	The Differences between Processes and Threads	641
24.5	Passing Data to a Thread and between Threads	641
24.6	Thread States and Thread Lifecycle	644
24.6.1	Sleep	645
24.6.2	Thread Joining	646
24.6.3	Thread Interrupt and Abort	648
24.7	Thread Priority	650

	Contents	xv
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

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

	Contents	xvii
A3.5	Boundary Conditions	769
A3.6	Example: Boundary Conditions for the One-factor Black-Scholes PDE	772
A3.7	Motivating the ADE Method	772
A3.8	The ADE Method Exposed	773
A3.9	The Convection Term	773
A3.10	Other Kinds of Boundary Conditions	774
A3.11	Nonlinear Problems	775
A3.12	ADE for PDEs in Conservative Form	775
A3.13	Numerical Results and Guidelines	776
	A3.13.1 The Consequences of Conditional Consistency	776
	A3.13.2 Call Payoff Behaviour at the Far Field	777
	A3.13.3 General Formulation of the ADE Method	777
A3.14	The Steps to Use when Implementing ADE	778
A3.15	Summary and Conclusions	778
A3.16	Exercises and Projects	779
A4	Cap, Floor and Swaption Using Excel-DNA	789
	A4.1 Introduction	789
	A4.2 Different Ways of Stripping Cap Volatility	789
	A4.3 Comparing Caplet Volatility Surface	792
	A4.4 Call Put Parity	794
	A4.5 Cap Price Matrix	795
	A4.6 Multi-strike and Amortising	797
	A4.7 Simple Swaption Formula	798
	A4.8 Swaption Straddle	800
	A4.9 Exercises	804
	Bibliography	805
	Web References	812
	Index	815