

Contents

1	Executive Overview of this Book	1
1.1	What is this book?	1
1.2	What's special about this book?	1
1.3	Who is this book for?	2
1.4	Software requirements	3
1.5	The structure of this book	4
1.6	Pedagogical approach	5
1.7	What this book is not	6
1.8	Source code on the CD	6
PART I TEMPLATE PROGRAMMING IN C++		
2	A Gentle Introduction to Templates in C++	9
2.1	Introduction and objectives	9
2.2	Motivation and background	10
2.3	Defining a template	11
2.3.1	An example	13
2.4	Template instantiation	15
2.5	Function templates	16
2.5.1	An example	17
2.6	Default values and typedefs	18
2.7	Guidelines when implementing templates	18
2.8	Conclusions and summary	19
3	An Introduction to the Standard Template Library	20
3.1	Introduction and objectives	20
3.1.1	Why use STL?	20
3.2	A Bird's-eye view of STL	20
3.3	Sequence containers	23
3.3.1	Programming lists	24
3.3.2	Vectors and arrays in STL	25
3.4	Associative containers	27
3.4.1	Sets in STL	27
3.4.2	Maps in STL	29

3.5	Iterators in STL	30
3.5.1	What kinds of iterators?	30
3.6	Algorithms	33
3.7	Using STL for financial instruments	35
3.8	Conclusions and summary	35
4	STL for Financial Engineering Applications	36
4.1	Introduction and objectives	36
4.2	Clever data structures	36
4.2.1	A simple output mechanism	38
4.3	Set theory and STL	40
4.4	Useful algorithms	43
4.5	STL adaptor containers	45
4.6	Conclusions and summary	46
5	The Property Pattern in Financial Engineering	47
5.1	Introduction and objectives	47
5.2	The Property pattern	47
5.2.1	Requirements for a Property pattern	48
5.3	An example	51
5.4	Extending the Property pattern: property sets and property lists	52
5.4.1	An example	55
5.5	Properties and exotic options	57
5.5.1	Example: Executive options	57
5.6	Conclusions and summary	59
 PART II BUILDING BLOCK CLASSES		
6	Arrays, Vectors and Matrices	63
6.1	Introduction and objectives	63
6.2	Motivation and background	64
6.3	A layered approach	66
6.4	The Array and Matrix classes in detail	66
6.4.1	Simple print functions	69
6.4.2	Array example	70
6.4.3	Matrix example	71
6.5	The Vector and NumericMatrix classes in detail	72
6.5.1	Vector example	74
6.5.2	NumericMatrix example	74
6.6	Associative arrays and matrices	74
6.7	Conclusions and summary	77
7	Arrays and Matrix Properties	78
7.1	Introduction and objectives	78
7.2	An overview of the functionality	78
7.3	Software requirements	79
7.3.1	Accuracy	79

7.3.2	Efficiency	79
7.3.3	Reliability	79
7.3.4	Understandability	80
7.4	The core processes	80
7.4.1	Interactions between matrices and vectors	83
7.4.2	Some examples	84
7.5	Other function categories	85
7.5.1	Measures of central tendency	85
7.5.2	Measures of dispersion	86
7.5.3	Moments, skewness, kurtosis	86
7.5.4	Inequalities	87
7.6	Using the functions	87
7.6.1	Calculating historical volatility	88
7.6.2	Variance of return of a portfolio	88
7.7	An introduction to exception handling	88
7.7.1	Try, throw and catch: A bit like tennis	89
7.8	Conclusions and summary	90
8	Numerical Linear Algebra	91
8.1	Introduction and objectives	91
8.2	An introduction to numerical linear algebra	91
8.2.1	Direct methods	93
8.2.2	Iterative methods	93
8.3	Tridiagonal systems	94
8.3.1	<i>LU</i> decomposition	94
8.3.2	Godunov's Double Sweep method	97
8.3.3	Designing and implementing tridiagonal schemes	99
8.4	Block tridiagonal systems	99
8.5	What requirements should our matrix satisfy?	101
8.5.1	Positive-definite matrices and diagonal dominance	101
8.5.2	<i>M</i> -Matrices	102
8.6	Conclusions and summary	102
9	Modelling Functions in C++	103
9.1	Introduction and objectives	103
9.2	Function pointers in C++	103
9.3	Function objects in STL	106
9.3.1	Comparison functions	108
9.3.2	STL and financial engineering	108
9.4	Some function types	109
9.4.1	Applications in numerical analysis and financial engineering	111
9.4.2	An example: Functions in option pricing	111
9.5	Creating your own function classes	111
9.6	Arrays of functions	114
9.7	Vector functions	115
9.8	Real-valued functions	115

9.9	Vector-valued functions	116
9.10	Conclusions and summary	116
10	C++ Classes for Statistical Distributions	117
10.1	Introduction and objectives	117
10.2	Discrete and continuous probability distribution functions	117
10.3	Continuous distributions	119
10.3.1	Uniform (rectangular) distribution	119
10.3.2	Normal distribution	121
10.3.3	Lognormal distribution	122
10.3.4	Gamma distribution and its specialisations	122
10.4	Discrete distributions	124
10.4.1	Poisson distribution	124
10.4.2	Binomial and Bernoulli distributions	125
10.4.3	Pascal and geometric distributions	126
10.5	Tests	127
10.5.1	Continuous distributions	127
10.5.2	Discrete distributions	127
10.6	Conclusions and summary	128
 PART III ORDINARY AND STOCHASTIC DIFFERENTIAL EQUATIONS		
11	Numerical Solution of Initial Value Problems: Fundamentals	131
11.1	Introduction and objectives	131
11.2	A model problem	132
11.2.1	Qualitative properties of the solution	132
11.3	Discretisation	133
11.4	Common schemes	134
11.5	Some theoretical issues	136
11.6	Fitting: Special schemes for difficult problems	137
11.7	Non-linear scalar problems and predictor–corrector methods	138
11.8	Extrapolation techniques	139
11.9	C++ design and implementation	140
11.10	Generalisations	143
11.11	Conclusions and summary	144
12	Stochastic Processes and Stochastic Differential Equations	145
12.1	Introduction and objectives	145
12.2	Random variables and random processes	145
12.2.1	Random variables	145
12.2.2	Generating random variables	147
12.2.3	Random (stochastic) processes	150
12.3	An introduction to stochastic differential equations	151
12.4	Some finite difference schemes	152
12.4.1	Improving the accuracy: Richardson extrapolation	153
12.5	Which scheme to use?	153

	12.6 Systems of SDEs	154
	12.7 Conclusions and summary	154
13	Two-Point Boundary Value Problems	155
	13.1 Introduction and objectives	155
	13.2 Description of problem	155
	13.3 (Traditional) centred-difference schemes	157
	13.3.1 Does the discrete system have a solution?	158
	13.3.2 Extrapolation	158
	13.4 Approximation of the boundary conditions	158
	13.4.1 Linearity boundary condition	159
	13.5 Exponentially fitted schemes and convection–diffusion	160
	13.6 Approximating the derivatives	160
	13.7 Design issues	161
	13.8 Conclusions and summary	163
14	Matrix Iterative Methods	164
	14.1 Introduction and objectives	164
	14.2 Iterative methods	165
	14.3 The Jacobi method	165
	14.4 Gauss–Seidel method	166
	14.5 Successive overrelaxation (SOR)	166
	14.6 Other methods	166
	14.6.1 The conjugate gradient method	167
	14.6.2 Block SOR	167
	14.6.3 Solving sparse systems of equations	167
	14.7 The linear complementarity problem	168
	14.8 Implementation	169
	14.9 Conclusions and summary	171
PART IV PROGRAMMING THE BLACK–SCHOLES ENVIRONMENT		
15	An Overview of Computational Finance	175
	15.1 Introduction and objectives	175
	15.2 The development life cycle	175
	15.3 Partial differential equations	176
	15.4 Numerical approximation of PDEs	177
	15.5 The class of finite difference schemes	179
	15.6 Special schemes for special problems	179
	15.7 Implementation issues and the choice of programming language	180
	15.8 Origins and application areas	180
	15.9 Conclusions and summary	181
16	Finite Difference Schemes for Black–Scholes	182
	16.1 Introduction and objectives	182
	16.2 Model problem: The one-dimensional heat equation	182
	16.3 The Black–Scholes equation	186

16.4	Initial conditions and exotic options payoffs	187
16.4.1	Payoff functions in options modelling	188
16.5	Implementation	190
16.6	Method of lines: A whirlwind introduction	190
16.7	Conclusions and summary	191
17	Implicit Finite Difference Schemes for Black–Scholes	192
17.1	Introduction and objectives	192
17.2	Fully implicit method	193
17.3	An introduction to the Crank–Nicolson method	194
17.4	A critique of Crank–Nicolson	195
17.4.1	How are derivatives approximated?	195
17.4.2	Boundary conditions	197
17.4.3	Initial conditions	198
17.4.4	Proving stability	198
17.5	Is there hope? the Keller scheme	199
17.5.1	The advantages of the Box scheme	201
17.6	Conclusions and summary	202
18	Special Schemes for Plain and Exotic Options	203
18.1	Introduction and objectives	203
18.2	Motivating exponentially fitted schemes	203
18.2.1	A new class of robust difference schemes	203
18.3	Exponentially fitted schemes for parabolic problems	205
18.3.1	The fitted scheme in more detail: Main results	205
18.4	What happens when the volatility goes to zero?	208
18.4.1	Graceful degradation	208
18.5	Exponential fitting with explicit time	209
18.5.1	An explicit time-marching scheme	209
18.6	Exponential fitting and exotic options	210
18.7	Some final remarks	211
19	My First Finite Difference Solver	212
19.1	Introduction and objectives	212
19.2	Modelling partial differential equations in C++	214
19.2.1	Function classes in C++	215
19.2.2	Function classes for partial differential equations	217
19.3	Finite difference schemes as C++ classes, Part I	218
19.4	Finite difference schemes as C++ classes, Part II	219
19.5	Initialisation issues	220
19.5.1	Functions and parameters	220
19.5.2	The main program	221
19.6	Interfacing with Excel	224
19.7	Conclusions and summary	224
20	An Introduction to ADI and Splitting Schemes	225
20.1	Introduction and objectives	225

20.2	A model problem	226
20.3	Motivation and history	227
20.4	Basic ADI scheme for the heat equation	228
20.4.1	Three-dimensional heat equation	229
20.5	Basic splitting scheme for the heat equation	230
20.5.1	Three-dimensional heat equation	231
20.6	Approximating cross-derivatives	231
20.7	Handling boundary conditions	232
20.8	Algorithms and design issues	234
20.9	Conclusions and summary	236
21	Numerical Approximation of Two-Factor Derivative Models	237
21.1	Introduction and objectives	237
21.2	Two-factor models in financial engineering	237
21.2.1	Asian options	237
21.2.2	Convertible bonds with random interest rates	239
21.2.3	Options with two underlying assets	239
21.2.4	Basket options	240
21.2.5	Fixed-income applications	241
21.3	Finite difference approximations	241
21.4	ADI schemes for Asian options	242
21.4.1	Upwinding	242
21.5	Splitting schemes	243
21.6	Conclusions and summary	243
PART V DESIGN PATTERNS		
22	A C++ Application for Displaying Numeric Data	247
22.1	Introduction and objectives	247
22.2	Input mechanisms	248
22.3	Conversion and processing mechanisms	249
22.4	Output and display mechanisms	250
22.4.1	Ensuring that Excel is started only once	251
22.5	Putting it all together	252
22.6	Output	252
22.7	Other functionality	252
22.7.1	Accessing cell data	254
22.7.2	Cell data for functions	255
22.7.3	Using Excel with finite difference schemes	255
22.8	Using Excel and property sets	258
22.9	Extensions and the road to design patterns	259
22.10	Conclusions and summary	260
23	Object Creational Patterns	261
23.1	Introduction and objectives	261
23.2	The Singleton pattern	263
23.2.1	The templated Singleton solution	263

23.2.2	An extended example	266
23.2.3	Applications to financial engineering	269
23.3	The Prototype pattern	270
23.3.1	The Prototype pattern: Solution	271
23.3.2	Applications to financial engineering	271
23.4	Factory Method pattern (virtual constructor)	272
23.4.1	An extended example	274
23.5	Abstract Factory pattern	275
23.5.1	The abstract factory: solution	277
23.5.2	An extended example	277
23.6	Applications to financial engineering	279
23.7	Conclusions and summary	279
24	Object Structural Patterns	281
24.1	Introduction and objectives	281
24.2	Kinds of structural relationships between classes	281
24.2.1	Aggregation	282
24.2.2	Association	283
24.2.3	Generalisation/specialisation	286
24.3	Whole–Part pattern	286
24.4	The Composite pattern	288
24.5	The Façade pattern	289
24.6	The Bridge pattern	290
24.6.1	An example of the Bridge pattern	290
24.7	Conclusions and summary	295
25	Object Behavioural Patterns	296
25.1	Introduction and objectives	296
25.2	Kinds of behavioural patterns	297
25.3	Iterator pattern	298
25.3.1	Iterating in composites	299
25.3.2	Iterating in property sets	300
25.4	The Visitor pattern	301
25.4.1	Visitors and the Extensible Markup Language (XML)	302
25.5	Notification patterns	305
25.6	Conclusions and summary	307
 PART VI DESIGN AND DEPLOYMENT ISSUES		
26	An Introduction to the Extensible Markup Language	311
26.1	Introduction and objectives	311
26.1.1	What’s the big deal with XML?	311
26.2	A short history of XML	312
26.3	The XML structure	312
26.3.1	XML files	312
26.3.2	XML syntax	313
26.3.3	Attributes in XML	314

26.4	Document Type Definition	315
26.4.1	DTD syntax	315
26.4.2	Validation issues	319
26.4.3	Limitations of DTDs	320
26.5	Extensible Stylesheet Language Transformation (XSLT)	320
26.5.1	Namespaces in XML	321
26.5.2	Main concepts in XSL	322
26.6	An application of XML: Financial products Markup Language	324
26.6.1	Product architecture overview	324
26.6.2	Example: Equity derivative options product architecture	326
26.7	Conclusions and summary	327
27	Advanced XML and Programming Interface	328
27.1	Introduction and objectives	328
27.2	XML Schema	328
27.2.1	Element declaration	329
27.2.2	User-defined simple and complex types	330
27.2.3	Multiplicity issues	331
27.2.4	An example	332
27.2.5	Comparing DTDs and the XML Schema	334
27.2.6	XML Schemas and FpML	334
27.3	Accessing XML data: The Document Object Model	334
27.3.1	DOM in a programming environment	335
27.4	DOM and C++: The essentials	335
27.5	DOM, entities and property sets	338
27.5.1	XML readers and writers	340
27.5.2	Examples and applications	342
27.6	XML structures for plain and barrier options	342
27.7	Conclusions and summary	345
28	Interfacing C++ and Excel	346
28.1	Introduction and objectives	346
28.2	Object model in Excel: An overview	346
28.3	Under the bonnet: Technical details of C++ interfacing to Excel	348
28.3.1	Startup	348
28.3.2	Creating charts and cell values	349
28.3.3	Interoperability with the SimplePropertySet	350
28.4	Implementing the core process	351
28.4.1	Registration: Getting basic input	352
28.4.2	Calculations	352
28.4.3	Displaying the results of the calculations	353
28.4.4	The application (main program)	354
28.5	Extensions	354
28.6	Application areas	355
28.7	Conclusions and summary	355

29	Advanced Excel Interfacing	356
29.1	Introduction and objectives	356
29.2	Status report and new requirements	356
29.3	A gentle introduction to Excel add-ins	357
29.3.1	What kinds of add-ins are there?	357
29.4	Automation add-in in detail	359
29.4.1	Functions with two parameters	362
29.4.2	Functions that accept a range	364
29.4.3	Using the Vector template class	366
29.5	Creating a COM add-in	367
29.6	Future trends	373
29.7	Conclusions and summary	373
30	An Extended Application: Option Strategies and Portfolios	374
30.1	Introduction and objectives	374
30.2	Spreads	374
30.3	Combinations: Straddles and strangles	375
30.4	Designing and implementing spreads	376
30.5	Delta hedging	378
30.6	An example	379
30.7	Tips and guidelines	381
Appendices		
A1	My C++ refresher	383
A2	Dates and other temporal types	394
References		397
Index		401