

Contents

Notation	xix
Executive Overview	xxiii
0 My First Monte Carlo Application One-Factor Problems	1
0.1 Introduction and objectives	1
0.2 Description of the problem	1
0.3 Ordinary differential equations (ODE)	2
0.4 Stochastic differential equations (SDE) and their solution	3
0.5 Generating uniform and normal random numbers	4
0.5.1 Uniform random number generation	4
0.5.2 Polar Marsaglia method	4
0.5.3 Box-Muller method	5
0.5.4 C++ code for uniform and normal random variate generation	5
0.5.5 Other methods	8
0.6 The Monte Carlo method	8
0.7 Calculating sensitivities	9
0.8 The initial C++ Monte Carlo framework: hierarchy and paths	10
0.9 The initial C++ Monte Carlo framework: calculating option price	19
0.10 The predictor-corrector method: a scheme for all seasons?	23
0.11 The Monte Carlo approach: caveats and nasty surprises	24
0.12 Summary and conclusions	25
0.13 Exercises and projects	25
 PART I FUNDAMENTALS	
 1 Mathematical Preparations for the Monte Carlo Method	 31
1.1 Introduction and objectives	31
1.2 Random variables	31
1.2.1 Fundamental concepts	31
1.2.2 Relative frequency and the beginnings of probability	32
1.2.3 Random variables	32

1.3	Discrete and continuous random variables	34
1.4	Multiple random variables	37
1.5	A short history of integration	38
1.6	σ -algebras, measurable spaces and measurable functions	39
1.7	Probability spaces and stochastic processes	40
1.8	The Ito stochastic integral	41
1.9	Applications of the Lebesgue theory	43
1.9.1	Convergence theorems	43
1.9.2	Fubini's theorem	45
1.10	Some useful inequalities	45
1.11	Some special functions	46
1.12	Convergence of function sequences	48
1.13	Applications to stochastic analysis	49
1.14	Summary and conclusions	50
1.15	Exercises and projects	50
2	The Mathematics of Stochastic Differential Equations (SDE)	53
2.1	Introduction and objectives	53
2.2	A survey of the literature	53
2.3	Mathematical foundations for SDEs	55
2.3.1	Metric spaces	55
2.3.2	Cauchy sequences, complete metric spaces and convergence	56
2.3.3	Continuous and Lipschitz continuous functions	57
2.4	Motivating random (stochastic) processes	59
2.5	An introduction to one-dimensional random processes	59
2.6	Stochastic differential equations in Banach spaces: prologue	62
2.7	Classes of SDEs and properties of their solutions	62
2.8	Existence and uniqueness results	63
2.9	A special SDE: the Ito equation	64
2.10	Numerical approximation of SDEs	66
2.11	Transforming an SDE: the Ito formula	68
2.12	Summary and conclusions	69
2.13	Appendix: proof of the Banach fixed-point theorem and some applications	69
2.14	Exercises and projects	71
3	Alternative SDEs and Toolkit Functionality	73
3.1	Introduction and objectives	73
3.2	Bessel processes	73
3.3	Random variate generation	74
3.4	The exponential distribution	74
3.5	The beta and gamma distributions	75
3.6	The chi-squared, Student and other distributions	79
3.7	Discrete variate generation	79
3.8	The Fokker-Planck equation	80
3.9	The relationship with PDEs	81
3.9.1	The Fichera function	81
3.10	Alternative stochastic processes	84

3.10.1	The constant elasticity of variance (CEV) model	85
3.10.2	Exact, quasi-exact and numerical solutions of CEV equation	86
3.10.3	Monte Carlo simulation of CEV models	87
3.10.4	American option pricing and hedging with the CEV model	91
3.10.5	Test cases and numerical results	92
3.11	Using associative arrays and matrices to model lookup tables and volatility surfaces	93
3.12	Summary and conclusion	96
3.13	Appendix: statistical distributions and special functions in the Boost library	97
3.13.1	The gamma functions	97
3.13.2	The error function	98
3.13.3	An overview of statistical distributions in Boost	99
3.14	Exercises and projects	102
4	An Introduction to the Finite Difference Method for SDE	107
4.1	Introduction and objectives	107
4.2	An introduction to discrete time simulation, motivation and notation	107
4.3	Foundations of discrete time approximation: ordinary differential equations	109
4.4	Foundations of discrete time approximation: stochastic differential equations	113
4.4.1	Absolute error criterion	113
4.4.2	Confidence intervals for the absolute error	114
4.4.3	Dependence of the absolute error on the step size	115
4.4.4	Strong and weak convergence	116
4.5	Some common schemes for one-factor SDEs	117
4.6	The Milstein schemes	117
4.7	Predictor-corrector methods	118
4.8	Stiff ordinary and stochastic differential equations	119
4.8.1	Robust schemes for stiff systems	121
4.8.2	Stiff stochastic differential equations	123
4.9	Software design and C++ implementation issues	125
4.10	Computational results	126
4.11	Aside: the characteristic equation of a difference scheme	127
4.12	Summary and conclusions	128
4.13	Exercises and projects	128
5	Design and Implementation of Finite Difference Schemes in Computational Finance	137
5.1	Introduction and objectives	137
5.2	Modelling SDEs and FDM in C++	137
5.3	Mathematical and numerical tools	138
5.3.1	Analytical solution and numerical solution of SDEs	139
5.3.2	Brownian Bridge	140
5.3.3	Finite difference methods for Brownian Bridges	142

5.4	The Karhunen-Loeve expansion	143
5.5	Cholesky decomposition	144
5.6	Spread options with stochastic volatility	146
5.7	The Heston stochastic volatility model	153
5.8	Path-dependent options and the Monte Carlo method	160
5.9	A small software framework for pricing options	161
5.10	Summary and conclusions	162
5.11	Exercises and projects	162
6	Advanced Finance Models and Numerical Methods	167
6.1	Introduction and objectives	167
6.2	Processes with jumps	168
6.2.1	Specific choices for jump processes	169
6.2.2	Simulating jumps, fixed dates	169
6.2.3	Simulating jump times	169
6.2.4	Finite difference approximations	170
6.3	Lévy processes	171
6.4	Measuring the order of convergence	172
6.4.1	A general convergence principle	172
6.4.2	Applications to ordinary differential equations	174
6.4.3	Applications to stochastic differential equations	175
6.5	Mollifiers, bump functions and function regularisation	176
6.6	When Monte Carlo does not work: counterexamples	177
6.6.1	How good is the random number generator <code>rand()</code> ?	177
6.6.2	Special stochastic differential equations	179
6.7	Approximating SDEs using strong Taylor, explicit and implicit schemes	179
6.7.1	Stability analysis of SDEs with multiplicative and additive noise	180
6.7.2	Strong Taylor approximants	181
6.7.3	Explicit strong schemes	182
6.7.4	Implicit strong schemes	183
6.8	Summary and conclusions	183
6.9	Exercises and projects	184
7	Foundations of the Monte Carlo Method	189
7.1	Introduction and objectives	189
7.2	Basic probability	189
7.3	The Law of Large Numbers	190
7.4	The Central Limit Theorem	191
7.4.1	Lindeberg-Feller Central Limit Theorem	192
7.4.2	Explicit calculations of the confidence intervals	193
7.5	Quasi Monte Carlo methods	194
7.5.1	Uniform distribution and its geometric interpretation	196
7.6	Summary and conclusions	198
7.7	Exercises and projects	198

PART II DESIGN PATTERNS

8 Architectures and Frameworks for Monte Carlo Methods: Overview	203
8.1 Goals of Part II of this book	203
8.2 Introduction and objectives of this chapter	203
8.3 The advantages of domain architectures	204
8.4 Software Architectures for the Monte Carlo method	207
8.4.1 Market model systems	207
8.4.2 Numerical simulator systems	209
8.4.3 Pricing, hedge and risk systems	211
8.5 Summary and conclusions	212
8.6 Exercises and projects	213
9 System Decomposition and System Patterns	217
9.1 Introduction and objectives	217
9.2 Software development process; a critique	217
9.3 System decomposition, from concept to code	217
9.3.1 Phases	218
9.4 Decomposition techniques, the process	220
9.4.1 A special case: geometric decomposition	221
9.4.2 Examples of task and data decomposition	222
9.5 Whole-part	222
9.6 Whole-part decomposition; the process	223
9.6.1 Task-based decomposition	223
9.6.2 Data decomposition	224
9.7 Presentation-Abstraction Control (PAC)	226
9.8 Building complex objects and configuring applications	229
9.8.1 The Factory Method pattern	229
9.8.2 Abstract Factory pattern	234
9.8.3 Builder pattern	237
9.9 Summary and conclusions	239
9.10 Exercises and projects	239
10 Detailed Design using the GOF Patterns	243
10.1 Introduction and objectives	243
10.2 Discovering which patterns to use	244
10.2.1 <i>Whole-Part</i> structure for a finite difference scheme	245
10.2.2 The Preprocessor agent	246
10.2.3 The Conversion agent	251
10.2.4 The Display agent and Top-Level agent	254
10.3 An overview of the GOF patterns	255
10.3.1 Strengths and limitations of GOF patterns	257
10.4 The essential structural patterns	257
10.4.1 Facade pattern	258
10.4.2 Bridge pattern, template version	258
10.4.3 Bridge pattern, run-time version	262
10.4.4 Adapter pattern	266
10.4.5 Layers pattern	266

10.5	The essential creational patterns	266
10.5.1	Factory Method pattern	267
10.5.2	Builder pattern	270
10.6	The essential behavioural patterns	270
10.6.1	<i>Visitor</i> pattern	271
10.6.2	<i>Strategy</i> and <i>Template</i> Method patterns	271
10.7	Summary and conclusions	276
10.8	Exercises and projects	276
11	Combining Object-Oriented and Generic Programming Models	281
11.1	Introduction and objectives	281
11.2	Using templates to implement components: overview	281
11.3	Templates versus inheritance, run-time versus compile-time	283
11.4	Advanced C++ templates	286
11.4.1	Default values for template parameters	286
11.4.2	Template nesting	287
11.4.3	Template member functions	289
11.4.4	Partial template specialisation	290
11.4.5	Template template parameters	291
11.5	Traits and policy-based design	294
11.5.1	Traits	294
11.5.2	Policy-based design (PBD)	301
11.5.3	When to use inheritance and when to use generics	304
11.6	Creating templated design patterns	306
11.7	A generic <i>Singleton</i> pattern	307
11.8	Generic composite structures	310
11.8.1	Applications of generic composites	314
11.9	Summary and conclusions	314
11.10	Exercises and projects	314
12	Data Structures and their Application to the Monte Carlo Method	319
12.1	Introduction and objectives	319
12.2	Arrays, vectors and matrices	319
12.3	Compile-time vectors and matrices	324
12.3.1	One-dimensional data structures: vectors	324
12.3.2	Two-dimensional data structures: matrices	327
12.3.3	Application: modelling N-factor SDEs	328
12.4	Creating adapters for STL containers	331
12.5	Date and time classes	334
12.5.1	Basic functionality in date class	335
12.5.2	Creating sets of dates and using set operations	335
12.5.3	Mapping dates to mesh points	338
12.5.4	Calendars, business days and business conventions	339
12.6	The class string	339
12.6.1	String constructors and string properties	340
12.6.2	Extracting characters and substrings	342

12.7	Modifying strings	343
12.7.1	Searching and finding in strings	343
12.7.2	Conversions between strings and other data types	344
12.8	A final look at the generic composite	345
12.9	Summary and conclusions	348
12.10	Exercises and projects	348

13 The Boost Library: An Introduction 353

13.1	Introduction and objectives	353
13.2	A taxonomy of C++ pointer types	353
13.2.1	The smart pointer types	354
13.2.2	Scoped pointers	354
13.2.3	Shared pointers	357
13.2.4	Using smart pointers in Monte Carlo applications	361
13.3	Modelling homogeneous and heterogeneous data in Boost	361
13.3.1	Tuples	361
13.3.2	Variants and discriminated unions	362
13.3.3	Any and undiscriminated types	363
13.3.4	A property class	365
13.3.5	Boost arrays	366
13.4	Boost signals: notification and data synchronisation	367
13.5	Input and output	368
13.5.1	Boost.Serialisation	368
13.5.2	Boost.Filesystem	370
13.6	Linear algebra and uBLAS	371
13.7	Date and time	372
13.8	Other libraries	372
13.8.1	String and text processing	373
13.8.2	Function objects and higher-order programming	374
13.8.3	Concurrent and multi-threading programming	374
13.8.4	Interval arithmetic	374
13.9	Summary and conclusions	374
13.10	Exercises and projects	374

PART III ADVANCED APPLICATIONS

14 Instruments and Payoffs 379

14.1	Introduction and objectives	379
14.2	Creating a C++ instrument hierarchy	379
14.2.1	Instrument hierarchy	380
14.2.2	Underlyings and derivatives	380
14.3	Modelling payoffs in C++	383
14.3.1	Simple property sets	383
14.3.2	Extending the payoff class	386
14.3.3	Heterogeneous property sets	386
14.3.4	The payoff base class	388

14.3.5	Abstract factories for payoff classes	389
14.3.6	Exception handling	391
14.4	Summary and conclusions	392
14.5	Exercises and projects	393
15	Path-Dependent Options	395
15.1	Introduction and objectives	395
15.2	Monte Carlo – a simple general-purpose version	396
15.2.1	The structure	396
15.2.2	Adding models	397
15.2.3	Adding schemes	398
15.2.4	The Monte Carlo evaluation	400
15.3	Asian options	401
15.3.1	Mapping Asian options into our framework	402
15.3.2	Approximation formulae for Asian options	405
15.3.3	Control variates	406
15.4	Options on the running Max/Min	411
15.5	Barrier options	412
15.5.1	Mapping barrier options into our framework	413
15.5.2	Approximation formulae for barrier options	414
15.5.3	Importance sampling	415
15.5.4	Antithetic variables	416
15.5.5	Numerical results	417
15.6	Lookback options	418
15.6.1	Approximation formulae for lookback options	420
15.6.2	Results	421
15.7	Cliquet Options	422
15.7.1	Remarks on cliquet options	422
15.7.2	C++ implementation	423
15.7.3	Version of the basic cliquet option	423
15.7.4	Results	424
15.8	Summary and conclusions	424
15.9	Exercises and projects	424
16	Affine Stochastic Volatility Models	427
16.1	Introduction and objectives	427
16.2	The volatility skew/smile	427
16.3	The Heston model	429
16.3.1	Implementing the Heston model	432
16.3.2	Euler scheme	434
16.3.3	Full truncation Euler scheme	434
16.3.4	Andersen’s QE scheme	435
16.3.5	A biased-free scheme	441
16.4	The Bates/SVJ model	441
16.5	Implementing the Bates model	443
16.6	Numerical results – European options	444

	Contents	xiii
16.7	Numerical results – skew-dependent options	446
16.7.1	European options – digital options	446
16.7.2	Path-dependent options – cliquet options	448
16.7.3	Results	449
16.8	XLL – using DLL within Microsoft Excel	449
16.8.1	What is an xll?	452
16.8.2	XLW – A framework for creating xll	452
16.8.3	Developing using XLW	452
16.9	Analytic solutions for affine stochastic volatility models	455
16.10	Summary and conclusions	457
16.11	Exercises and projects	458
17	Multi-Asset Options	461
17.1	Introduction and objectives	461
17.2	Modelling in multiple dimensions	461
17.3	Implementing payoff classes for multi-asset options	465
17.4	Some multi-asset options	466
17.4.1	Spread options	466
17.4.2	Approximation formulae	466
17.4.3	Quanto options	468
17.5	Basket options	469
17.6	Min/Max options	471
17.7	Mountain range options	475
17.8	The Heston model in multiple dimensions	480
17.8.1	Extending the QE scheme	481
17.8.2	Spread options	482
17.9	Equity interest rate hybrids	482
17.10	Summary and conclusions	486
17.11	Exercises and projects	486
18	Advanced Monte Carlo I – Computing Greeks	489
18.1	Introduction and objectives	489
18.2	The finite difference method	489
18.3	The pathwise method	492
18.4	The likelihood ratio method	497
18.4.1	Examples	499
18.5	Likelihood ratio for finite differences – proxy simulation	503
18.6	Summary and conclusions	504
18.7	Exercises and projects	506
19	Advanced Monte Carlo II – Early Exercise	511
19.1	Introduction and objectives	511
19.2	Description of the problem	511
19.3	Pricing American options by regression	512
19.4	C++ design	513
19.5	Linear least squares regression	516

19.5.1	The regression function	518
19.5.2	The exercise decision	519
19.6	Example – step by step	520
19.7	Analysis of the method and improvements	521
19.7.1	Selecting basis functions	522
19.8	Upper bounds	525
19.9	Examples	527
19.10	Summary and conclusions	528
19.11	Exercises and projects	528
20	Beyond Brownian Motion	531
20.1	Introduction and objectives	531
20.2	Normal mean variance mixture models	531
20.2.1	The normal inverse Gaussian model	532
20.2.2	C++ implementation	532
20.2.3	The variance gamma model	535
20.2.4	Matching underlying assets and martingale dynamics	535
20.3	The multi-dimensional case	536
20.4	Summary and conclusions	536
20.5	Exercises and projects	538
 PART IV SUPPLEMENTS		
21	C++ Application Optimisation and Performance Improvement	543
21.1	Introduction and objectives	543
21.2	Modelling functions in C++: choices and consequences	543
21.2.1	Function pointers	544
21.2.2	Generic classes that model functions	547
21.2.3	Function objects in STL	549
21.2.4	What is polymorphism?	550
21.3	Performance issues in C++: classifying potential bottlenecks	552
21.3.1	Inlining	553
21.3.2	Static and dynamic casting	554
21.3.3	Preventing unnecessary object construction	554
21.3.4	Exception handling	557
21.3.5	Templates versus inheritance	558
21.3.6	Performance and the STL	559
21.3.7	Some final tips	559
21.4	Temporary objects	560
21.5	Special features in the Boost library	562
21.6	Boost multiaarray library	563
21.7	Boost random number library	564
21.8	STL and Boost smart pointers: final remarks	566
21.9	Summary and conclusions	568
21.10	Exercises, projects and guidelines	569

22	Random Number Generation and Distributions	571
22.1	Introduction and objectives	571
22.2	Uniform number generation	571
22.2.1	Pseudo random number generators	572
22.2.2	Quasi random number generators	576
22.2.3	Sobol numbers	576
22.3	The Sobol class	578
22.4	Number generation due to given distributions	580
22.4.1	Methods for computing variates	581
22.4.2	The inverse method	581
22.4.3	Acceptance/Rejection and ratio of uniforms	581
22.4.4	C++ implementation	582
22.4.5	Generating normal variates	583
22.4.6	Generating gamma distributed variates	583
22.4.7	Generating χ^2 -distributed variates	587
22.4.8	Generating Poisson distributed variates	587
22.5	Jump processes	588
22.5.1	The Poisson process	588
22.5.2	Simple generalisations	590
22.5.3	Compensation and compounding	590
22.5.4	Poisson random measures	591
22.5.5	Jump measures	592
22.6	The random generator templates	593
22.7	Tests for randomness	596
22.8	Summary and conclusions	596
22.9	Exercises and projects	597
23	Some Mathematical Background	601
23.1	Introduction and objectives	601
23.2	A matrix class	601
23.3	Matrix functions	601
23.3.1	The Cholesky decomposition	601
23.3.2	The spectral decomposition	603
23.3.3	Singular value decomposition	607
23.4	Functional analysis	608
23.4.1	The basics	608
23.4.2	Fourier transform	609
23.4.3	Hilbert spaces and L_2 -basis	609
23.4.4	Measures and Fourier transform	609
23.4.5	The characteristic function of a random variable	610
23.5	Applications to option pricing	610
23.5.1	Semi-analytical prices and their C++ implementation	611
23.6	Summary and conclusions	614
23.7	Exercises and projects	614

24	An Introduction to Multi-threaded and Parallel Programming	617
24.1	Introduction and objectives	617
24.2	Shared memory models	617
24.3	Sequential, concurrent and parallel programming 101	619
24.4	How fast is fast? Performance analysis of parallel programs	623
24.5	An introduction to processes and threads	625
24.5.1	The life of a thread	625
24.5.2	How threads communicate	626
24.6	What kinds of applications are suitable for multi-threading?	626
24.6.1	Suitable tasks for multi-threading	626
24.7	The multi-threaded application lifecycle	627
24.7.1	Finding concurrency and decomposition patterns	627
24.7.2	Algorithm structure	627
24.7.3	Supporting structures, models and patterns	628
24.7.4	Implementation mechanisms	628
24.8	Some model architectures	629
24.8.1	The Monte Carlo engine	631
24.8.2	The market model system	632
24.8.3	What's next?	633
24.9	Analysing and designing large software systems: a summary of the steps	633
24.10	Conclusions and summary	634
24.11	Exercises and projects	634
25	An Introduction to OpenMP and its Applications to the Monte Carlo Method	637
25.1	Introduction and objectives	637
25.2	Loop optimisation	637
25.2.1	Loop interchange	638
25.2.2	Loop fission and fusion	639
25.2.3	Loop unrolling	640
25.2.4	Loop tiling	641
25.2.5	Serial loop optimisation in Monte Carlo applications	642
25.3	An overview of OpenMP	644
25.4	Threads in OpenMP	644
25.5	Loop-level parallelism	646
25.6	Data sharing	646
25.6.1	Ensuring private variable initialisation and finalisation	648
25.6.2	Reduction operations	649
25.6.3	The <i>copyin</i> clause and <i>threadprivate</i> directive	650
25.6.4	The <i>flush</i> directive	651
25.7	Work-sharing and parallel regions	651
25.7.1	Work-sharing constructs	652
25.7.2	The <i>nowait</i> clause	653
25.8	Nested loop optimisation	654
25.9	Scheduling in OpenMP	656

25.10	OpenMP for the Monte Carlo method	657
25.11	Conclusions and summary	661
25.12	Exercises and projects	661
26	A Case Study of Numerical Schemes for the Heston Model	665
26.1	Introduction and objectives	665
26.2	Test scenarios	666
26.3	Numerical approximations for the Heston model	667
26.4	Testing different schemes and scenarios	672
26.5	Results	675
26.6	Lessons learned	678
26.7	Extensions, calibration and more	679
26.8	Other numerical methods for Heston	680
26.9	Summary and conclusions	681
26.10	Exercises and projects	681
27	Excel, C++ and Monte Carlo Integration	685
27.1	Introduction and objectives	685
27.2	Integrating applications and Excel	686
27.3	ATL architecture	686
27.3.1	Interfaces and data structures	686
27.3.2	The structure of an ATL project and its classes	688
27.3.3	Implementing the methods in IDTExtensibility2	691
27.4	Creating my first ATL project: the steps	693
27.5	Creating automation add-ins in Excel	695
27.5.1	Automation interfaces	695
27.6	Useful utilities and interoperability projects	696
27.7	Test Case: a COM add-in and complete code	697
27.7.1	COM add-in	697
27.7.2	Automation add-in	701
27.7.3	The utilities class	704
27.8	Summary and conclusions	707
27.9	Exercises and projects	707
	References	711
	Index	719

