

Contents

List of figures	xi	Example of an ill-conditioned problem	25
List of tables	xvii	2.5 Condition number of a matrix	26
List of algorithms	xix	Comments and examples	27
Acknowledgments	xxi	2.6 A primer on algorithmic and computational complexity	28
Foreword to the second edition	xxiii	Criteria for comparison	28
		Order of complexity and classification	29
		Appendix 2.A Operation count for basic linear algebra operations	30
Part I			
Fundamentals			
1. Introduction		3. Linear equations and Least Squares problems	
1.1 About this book	3	3.1 Direct methods	32
The growth of computing power	3	3.1.1 Triangular systems	32
Computational finance	4	3.1.2 LU factorization	33
1.2 Principles	6	3.1.3 Cholesky factorization	35
1.3 On software	7	3.1.4 QR decomposition	37
1.4 On approximations and accuracy	10	3.1.5 Singular value decomposition	37
1.5 Summary: the theme of the book	15	3.2 Iterative methods	38
2. Numerical analysis in a nutshell		3.2.1 Jacobi, Gauss–Seidel, and SOR	39
2.1 Computer arithmetic	17	Successive overrelaxation	40
Representation of real numbers	17	3.2.2 Convergence of iterative methods	41
Machine precision	19	3.2.3 General structure of algorithms for iterative methods	42
Example of limitations of floating point arithmetic	19	3.2.4 Block iterative methods	44
2.2 Measuring errors	20	3.3 Sparse linear systems	45
2.3 Approximating derivatives with finite differences	21	3.3.1 Tridiagonal systems	45
Approximating first-order derivatives	21	3.3.2 Irregular sparse matrices	47
Approximating second-order derivatives	22	3.3.3 Structural properties of sparse matrices	48
Partial derivatives	22	3.4 The Least Squares problem	50
How to choose h	22	3.4.1 Method of normal equations	51
Truncation error for forward difference	23	3.4.2 Least Squares via QR factorization	54
2.4 Numerical instability and ill-conditioning	24	3.4.3 Least Squares via SVD decomposition	55
Example of a numerically unstable algorithm	24	3.4.4 Final remarks	56

The backslash operator in MATLAB	56	6.2.2 Mersenne Twister	107
Appendix 3.A Solving linear systems in R	56	6.3 Nonuniform distributions	107
solve	57	6.3.1 The inversion method	107
Least Squares	58	6.3.2 Acceptance–rejection method	109
4. Finite difference methods		6.4 Specialized methods for selected distributions	111
4.1 An example of a numerical solution	61	6.4.1 Normal distribution	111
A first numerical approximation	62	6.4.2 Higher order moments and the Cornish–Fisher expansion	113
A second numerical approximation	63	6.4.3 Further distributions	114
4.2 Classification of differential equations	64	6.5 Sampling from a discrete set	116
4.3 The Black–Scholes equation	65	6.5.1 Discrete uniform selection	116
4.3.1 Explicit, implicit, and θ -methods	67	6.5.2 Roulette wheel selection	117
4.3.2 Initial and boundary conditions and definition of the grid	67	6.5.3 Random permutations and shuffling	118
4.3.3 Implementation of the θ -method with MATLAB	71	6.6 Sampling errors—and how to reduce them	118
4.3.4 Stability	73	6.6.1 The basic problem	118
4.3.5 Coordinate transformation of space variables	76	6.6.2 Quasi-Monte Carlo	119
4.4 American options	79	6.6.3 Stratified sampling	120
Appendix 4.A A note on MATLAB’s function <code>spdiags</code>	86	6.6.4 Variance reduction	121
5. Binomial trees		6.7 Drawing from empirical distributions	122
5.1 Motivation	89	6.7.1 Data randomization	122
Matching moments	89	6.7.2 Bootstrap	122
5.2 Growing the tree	90	6.8 Controlled experiments and experimental design	127
5.2.1 Implementing a tree	91	6.8.1 Replicability and <i>ceteris paribus</i> analysis	127
5.2.2 Vectorization	92	6.8.2 Available random number generators in MATLAB	128
5.2.3 Binomial expansion	93	6.8.3 Uniform random numbers from MATLAB’s <code>rand</code> function	128
5.3 Early exercise	95	6.8.4 Gaussian random numbers from MATLAB’s <code>randn</code> function	129
5.4 Dividends	95	6.8.5 Remedies	131
5.5 The Greeks	97	7. Modeling dependencies	
Greeks from the tree	97	7.1 Transformation methods	133
		7.1.1 Linear correlation	133
		7.1.2 Rank correlation	138
		7.2 Markov chains	144
		7.2.1 Concepts	144
		7.2.2 The Metropolis algorithm	146
		7.3 Copula models	148
		7.3.1 Concepts	148
		7.3.2 Simulation using copulas	150
Part II		8. A gentle introduction to financial simulation	
Simulation		8.1 Setting the stage	153
6. Generating random numbers		8.2 Single-period simulations	154
6.1 Monte Carlo methods and sampling	103		
6.1.1 How it all began	103		
6.1.2 Financial applications	104		
6.2 Uniform random number generators	104		
6.2.1 Congruential generators	104		

8.2.1 Terminal asset prices	154
8.2.2 1-over- N portfolios	155
8.2.3 European options	157
8.2.4 VaR of a covered put portfolio	159
8.3 Simple price processes	161
8.4 Processes with memory in the levels of returns	163
8.4.1 Efficient versus adaptive markets	163
8.4.2 Moving averages	163
8.4.3 Autoregressive models	164
8.4.4 Autoregressive moving average (ARMA) models	165
8.4.5 Simulating ARMA models	166
8.4.6 Models with long-term memory	167
8.5 Time-varying volatility	169
8.5.1 The concepts	169
8.5.2 Autocorrelated time-varying volatility	170
8.5.3 Simulating GARCH processes	173
8.5.4 Selected further autoregressive volatility models	175
8.6 Adaptive expectations and patterns in price processes	178
8.6.1 Price-earnings models	178
8.6.2 Models with learning	179
8.7 Historical simulation	180
8.7.1 Backtesting	180
8.7.2 Bootstrap	181
8.8 Agent-based models and complexity	185
9. Financial simulation at work: some case studies	
9.1 Constant proportion portfolio insurance (CPPI)	189
9.1.1 Basic concepts	189
9.1.2 Bootstrap	191
9.2 VaR estimation with Extreme Value Theory	192
9.2.1 Basic concepts	192
9.2.2 Scaling the data	193
9.2.3 Using Extreme Value Theory	193
9.3 Option pricing	195
9.3.1 Modeling prices	196
9.3.2 Pricing models	199
9.3.3 Greeks	208
9.3.4 Quasi-Monte Carlo	210

Part III Optimization

10. Optimization problems in finance

10.1 What to optimize?	219
10.2 Solving the model	220
10.2.1 Problems	220
10.2.2 Classical methods and heuristics	222
10.3 Evaluating solutions	222
10.4 Examples	224
Portfolio optimization with alternative risk measures	224
Model selection	225
Robust/resistant regression	225
Agent-based models	226
Calibration of option-pricing models	226
Calibration of yield structure models	227
10.5 Summary	228

11. Basic methods

11.1 Finding the roots of $f(x) = 0$	229
11.1.1 A naïve approach	229
Graphical solution	230
Random search	231
11.1.2 Bracketing	231
11.1.3 Bisection	232
11.1.4 Fixed point method	233
Convergence	235
11.1.5 Newton's method	238
Comments	240
11.2 Classical unconstrained optimization	241
Convergence	242
11.3 Unconstrained optimization in one dimension	243
11.3.1 Newton's method	243
11.3.2 Golden section search	244
11.4 Unconstrained optimization in multiple dimensions	245
11.4.1 Steepest descent method	245
11.4.2 Newton's method	247
11.4.3 Quasi-Newton method	248
11.4.4 Direct search methods	250
11.4.5 Practical issues with MATLAB	254
11.5 Nonlinear Least Squares	256
11.5.1 Problem statement and notation	256
11.5.2 Gauss-Newton method	257

11.5.3	Levenberg–Marquardt method	258	Appendix 12.B Parallel computations in MATLAB	309	
11.6	Solving systems of nonlinear equations $F(x) = 0$	260	12.B.1	Parallel execution of restart loops	311
11.6.1	General considerations	260	Appendix 12.C Heuristic methods in the NMOF package	314	
11.6.2	Fixed point methods	262	12.C.1	Local Search	314
11.6.3	Newton’s method	263	12.C.2	Simulated Annealing	315
11.6.4	Quasi-Newton methods	268	12.C.3	Threshold Accepting	315
11.6.5	Further approaches	269	12.C.4	Genetic Algorithm	315
11.7	Synoptic view of solution methods	270	12.C.5	Differential Evolution	316
			12.C.6	Particle Swarm Optimization	316
			12.C.7	Restarts	317
12.	Heuristic methods in a nutshell		13. Heuristics: a tutorial		
12.1	Heuristics	273	13.1 On Optimization	319	
	What is a heuristic?	274	13.1.1	Models	319
	Iterative search	275	13.1.2	Methods	320
12.2	Single-solution methods	276	13.2 The problem: choosing few from many	320	
12.2.1	Stochastic Local Search	276	13.2.1	The subset-sum problem	320
12.2.2	Simulated Annealing	277	13.2.2	Representing a solution	321
12.2.3	Threshold Accepting	278	13.2.3	Evaluating a solution	321
12.2.4	Tabu Search	279	13.2.4	Knowing the solution	322
12.3	Population-based methods	279	13.3 Solution strategies	323	
12.3.1	Genetic Algorithms	279	13.3.1	Being thorough	323
12.3.2	Differential Evolution	280	13.3.2	Being constructive	324
12.3.3	Particle Swarm Optimization	281	13.3.3	Being random	325
12.3.4	Ant Colony Optimization	282	13.3.4	Getting better	327
12.4	Hybrids	282	13.4 Heuristics	330	
12.5	Constraints	284	13.4.1	On heuristics	330
12.6	The stochastics of heuristic search	285	13.4.2	Local Search	331
12.6.1	Stochastic solutions and computational resources	285	13.4.3	Threshold Accepting	336
12.6.2	An illustrative experiment	287	13.4.4	Settings, or: how (long) to run an algorithm	340
12.7	General considerations	289	13.4.5	Stochastics of LS and TA	340
12.7.1	What technique to choose?	289	13.5 Application: selecting variables in a regression	342	
12.7.2	Efficient implementations	289	13.5.1	Linear models	342
12.7.3	Parameter settings	293	13.5.2	Fast least squares	343
12.8	Outlook	294	13.5.3	Selection criterion	344
			13.5.4	Putting it all together	345
Appendix 12.A Implementing heuristic methods with MATLAB		294	13.6 Application: portfolio selection	347	
12.A.1	The problems	296	13.6.1	Models	347
12.A.2	Threshold Accepting	298	13.6.2	Local-Search algorithms	348
12.A.3	Genetic Algorithm	303	14. Portfolio optimization		
12.A.4	Differential Evolution	306	14.1 The investment problem	355	
12.A.5	Particle Swarm Optimization	308	14.2 Mean–variance optimization	357	
			14.2.1	The model	357

14.2.2	Solving the model	358	15.4 Backtesting portfolio strategies	459	
14.2.3	Examples of mean–variance models	358	15.4.1	Kenneth French’s data library	459
14.2.4	True, estimated, and realized frontiers	366	15.4.2	Momentum	464
14.2.5	Repairing matrices	368	15.4.3	Portfolio optimization	470
14.3 Optimization with heuristics		377	Appendix 15.A Prices in <code>btest</code>	479	
14.3.1	Asset selection with Local Search	377	Appendix 15.B Notes on <code>zoo</code>	479	
14.3.2	Scenario Optimization with Threshold Accepting	383	Appendix 15.C Parallel computations in R	480	
14.3.3	Portfolio optimization with TA: examples	392	15.C.1	Distributed computing	480
14.3.4	Diagnostics for techniques based on Local Search	411	15.C.2	Loops and <code>apply</code> functions	481
14.4 Portfolios under Value-at-Risk		413	15.C.3	Distributing data	482
14.4.1	Why Value-at-Risk matters	413	15.C.4	Distributing data, continued	484
14.4.2	Setting up experiments	414	15.C.5	Other functions in the <code>parallel</code> package	486
14.4.3	Numerical results	415	15.C.6	Parallel computations in the <code>NMOF</code> package	486
Appendix 14.A Computing returns		419	16. Econometric models		
Appendix 14.B More implementation issues in R		420	16.1 Term structure models	487	
14.B.1	Scoping rules in R and objective functions	420	16.1.1	Yield curves	487
14.B.2	Vectorized objective functions	422	16.1.2	The Nelson–Siegel model	492
Appendix 14.C A neighborhood for switching elements		424	16.1.3	Calibration strategies	496
15. Backtesting			16.1.4	Experiments	518
15.1 What is (the problem with) backtesting?		427	16.2 Robust and resistant regression	522	
15.1.1	The ugly: intentional overfitting	428	16.2.1	The regression model	525
15.1.2	The bad: unintentional overfitting and other difficulties	434	16.2.2	Estimation	527
15.1.3	The good: getting insights (and confidence) in strategies	436	16.2.3	An example	532
15.2 Data and software		437	16.2.4	Numerical experiments	535
15.2.1	What data to use?	437	16.2.5	Final remarks	540
15.2.2	Designing backtesting software	439	16.3 Estimating Time Series Models	542	
15.2.3	The <code>btest</code> function	439	16.3.1	Adventures with time series estimation	542
15.3 Simple backtests		440	16.3.2	The case of GARCH models	543
15.3.1	<code>btest</code> : a tutorial	440	16.3.3	Numerical experiments with Differential Evolution	545
15.3.2	Robert Shiller’s Irrational-Exuberance data	450	Appendix 16.A Maximizing the Sharpe ratio	549	
			17. Calibrating option pricing models		
			17.1 Implied volatility with Black–Scholes	552	
			The smile	554	
			17.2 Pricing with the characteristic function	555	
			17.2.1	A pricing equation	555

17.2.2 Numerical integration	560	A. The NMOF package	
17.3 Calibration	580	A.1 Installing the package	597
17.3.1 Techniques	580	A.2 News, feedback and discussion	597
17.3.2 Organizing the problem and implementation	582	A.3 Using the package	597
17.3.3 Two experiments	589		
17.4 Final remarks	593		
Appendix 17.A Quadrature rules for infinity	594	Bibliography	599
		Index	609