ELSEVIER
爱思唯尔

Manfred Gilli,
Dietmar Maringer,
Enrico Schumann

Numerical Methods and Optimization in Finance

金融中的数值方法和优化

世界图书出版公司
www.wpcbj.com.cn

Manfred Gilli,
Dietmar Maringer,
Enrico Schumann

Numerical Methods and Optimization in Finance

[illegible]

[illegible]
[illegible]

Numerical Methods and Optimization in Finance

Manfred Gilli
University of Geneva and Swiss Finance Institute

Dietmar Maringer
University of Basel and University of Geneva

Enrico Schumann
VIP Value Investment Professionals AG, Switzerland

ELSEVIER

AMSTERDAM • BOSTON • HEIDELBERG • LONDON
NEW YORK • OXFORD • PARIS • SAN DIEGO
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO
Academic Press is an imprint of Elsevier

图书在版编目（CIP）数据

金融中的数值方法和优化 = Numerical Methods and Optimization in Finance：英文/（瑞士）吉利（Gilli，M.）著．—影印本．—北京：世界图书出版公司北京公司，2012.9

ISBN 978－7－5100－5265－1

Ⅰ.①金…　Ⅱ.①吉…　Ⅲ.①金融—数值方法—研究—英文②金融—最佳化—研究—英文　Ⅳ.①F830

中国版本图书馆 CIP 数据核字（2012）第 217183 号

书　　名：	Numerical Methods and Optimization in Finance
作　　者：	Manfred Gilli，Dietmar Maringer，Enrico Schumann
中 译 名：	金融中的数值方法和优化
责任编辑：	高蓉　刘慧
出 版 者：	世界图书出版公司北京公司
印 刷 者：	三河市国英印务有限公司
发　　行：	世界图书出版公司北京公司（北京朝内大街 137 号 100010）
联系电话：	010－64021602，010－64015659
电子信箱：	kjb@ wpcbj. com. cn
开　　本：	24 开
印　　张：	25
版　　次：	2013 年 1 月
版权登记：	图字：01－2012－6311
书　　号：	978－7－5100－5265－1　　　**定　　价：**　89.00 元

Acknowledgements

This book would not have been possible without the help of many people. First, we thank Peter Winker who was a teacher to all of us (in one way or another). Peter coordinated the COMISEF (Computational Optimization in Statistics, Econometrics and Finance) project; much of the research described in this book has grown out of this project. We are grateful to the EU for having financed COMISEF (Project MRTN-CT-2006-034270). Also, we would like to thank all the members of COMISEF for providing a fruitful network.

Next, there is Karen Maloney from Elsevier who initiated this endeavor. Without her, the book would never have been written. Many thanks also go to Scott Bentley for managing our project. Then, there is Evis Këllezi, who has been contributing for many years to the research that went into this book, and, not least, she read and commented on many drafts.

There are many other people whom we want to thank. They contributed in many ways: critical suggestions and comments, moral support, patient technical help, inspiration, and proofreading. Thanks go to Gerda Cabej, Giacomo di Tollo, Alain Dubois, Stefan Große, Benoît Guilleminot, Marc Hofmann, Hilda Hysi, Susan Kriete-Dodds, Yvan Lengwiler, Jonela Lula, Alfio Marazzi, Christian Oesch, Ilir Roko, Giorgio Pauletto, Evdoxia Pliota, Tikesh Ramtohul, Gregor Reich, Jin Zhang, and many more.

And finally, one of the authors would like to thank Caroline and Walter, who allowed him to work in the inspiring environment of Chalet Dagobert in Verbier from which the book greatly benefitted.

Manfred Gilli
Dietmar Maringer
Enrico Schumann

Geneva, Basel, Lucerne
January 2011

试读结束，需要全本PDF请购买 www.ertongbook.com

Contents

List of Algorithms xiii

Acknowledgements xv

1. Introduction 1

- **1.1 About this book** 1
- **1.2 Principles** 3
- **1.3 On software** 5
- **1.4 On approximations and accuracy** 9
- **1.5 Summary: the theme of the book** 14

Part One Fundamentals 15

2. Numerical analysis in a nutshell 17

- **2.1 Computer arithmetic** 17
  - Representation of real numbers 17
  - Machine precision 20
  - Example of limitations of floating point arithmetic 20
- **2.2 Measuring errors** 21
- **2.3 Approximating derivatives with finite differences** 22
  - Approximating first-order derivatives 22
  - Approximating second-order derivatives 23
  - Partial derivatives 23
  - How to choose *h* 24
  - Truncation error for forward difference 24
- **2.4 Numerical instability and ill-conditioning** 26
  - Example of a numerically unstable algorithm 26
  - Example of an ill-conditioned problem 27
- **2.5 Condition number of a matrix** 28
  - Comments and examples 29
- **2.6 A primer on algorithmic and computational complexity** 31
  - 2.6.1 Criteria for comparison 31
  - Order of complexity and classification 32
- **2.A Operation count for basic linear algebra operations** 33

3. Linear equations and Least Squares problems 35

- Choice of method 36
- **3.1 Direct methods** 36
  - 3.1.1 Triangular systems 36
  - 3.1.2 LU factorization 37

3.1.3 Cholesky factorization 40
3.1.4 QR decomposition 43
3.1.5 Singular value decomposition 44
3.2 Iterative methods 44
3.2.1 Jacobi, Gauss–Seidel, and SOR 45
Successive overrelaxation 46
3.2.2 Convergence of iterative methods 48
3.2.3 General structure of algorithms for iterative methods 49
3.2.4 Block iterative methods 52
3.3 Sparse linear systems 53
3.3.1 Tridiagonal systems 53
3.3.2 Irregular sparse matrices 55
3.3.3 Structural properties of sparse matrices 57
3.4 The Least Squares problem 60
3.4.1 Method of normal equations 61
3.4.2 Least Squares via QR factorization 64
3.4.3 Least Squares via SVD decomposition 65
3.4.4 Final remarks 67
The backslash operator in Matlab 67

4. Finite difference methods 69
4.1 An example of a numerical solution 69
A first numerical approximation 70
A second numerical approximation 71
4.2 Classification of differential equations 73
4.3 The Black–Scholes equation 74
4.3.1 Explicit, implicit, and θ-methods 76
4.3.2 Initial and boundary conditions and definition of the grid 76
4.3.3 Implementation of the θ-method with Matlab 82
4.3.4 Stability 85
4.3.5 Coordinate transformation of space variables 88
4.4 American options 90
4.A A note on Matlab's function `spdiags` 101

5. Binomial trees 103
5.1 Motivation 103
Matching moments 104
5.2 Growing the tree 105
5.2.1 Implementing a tree 106
5.2.2 Vectorization 107
5.2.3 Binomial expansion 108
5.3 Early exercise 110
5.4 Dividends 111
5.5 The Greeks 113
Greeks from the tree 113

Part Two Simulation 117

6. Generating random numbers 119
- **6.1 Monte Carlo methods and sampling** 119
  - 6.1.1 How it all began 119
  - 6.1.2 Financial applications 120
- **6.2 Uniform random number generators** 121
  - 6.2.1 Congruential generators 121
  - 6.2.2 Mersenne Twister 124
- **6.3 Nonuniform distributions** 125
  - 6.3.1 The inversion method 125
  - 6.3.2 Acceptance–rejection method 127
- **6.4 Specialized methods for selected distributions** 129
  - 6.4.1 Normal distribution 129
  - 6.4.2 Higher order moments and the Cornish–Fisher expansion 132
  - 6.4.3 Further distributions 134
- **6.5 Sampling from a discrete set** 136
  - 6.5.1 Discrete uniform selection 136
  - 6.5.2 Roulette wheel selection 137
  - 6.5.3 Random permutations and shuffling 138
- **6.6 Sampling errors—and how to reduce them** 139
  - 6.6.1 The basic problem 139
  - 6.6.2 Quasi-Monte Carlo 140
  - 6.6.3 Stratified sampling 142
  - 6.6.4 Variance reduction 143
- **6.7 Drawing from empirical distributions** 145
  - 6.7.1 Data randomization 145
  - 6.7.2 Bootstrap 146
- **6.8 Controlled experiments and experimental design** 152
  - 6.8.1 Replicability and *ceteris paribus* analysis 152
  - 6.8.2 Available random number generators in Matlab 153
  - 6.8.3 Uniform random numbers from Matlab's `rand` function 154
  - 6.8.4 Gaussian random numbers from Matlab's `randn` function 155
  - 6.8.5 Remedies 156

7. Modeling dependencies 159
- **7.1 Transformation methods** 159
  - 7.1.1 Linear correlation 159
  - 7.1.2 Rank correlation 165
- **7.2 Markov chains** 173
  - 7.2.1 Concepts 173
  - 7.2.2 The Metropolis algorithm 175
- **7.3 Copula models** 178
  - 7.3.1 Concepts 178
  - 7.3.2 Simulation using copulas 181

8. A gentle introduction to financial simulation 185

8.1 Setting the stage 185

8.2 Single-period simulations 186

8.2.1 Terminal asset prices 186

8.2.2 1-over-*N* portfolios 188

8.2.3 European options 190

8.2.4 VaR of a covered put portfolio 193

8.3 Simple price processes 196

8.4 Processes with memory in the levels of returns 198

8.4.1 Efficient versus adaptive markets 198

8.4.2 Moving averages 199

8.4.3 Autoregressive models 200

8.4.4 Autoregressive moving average (ARMA) models 202

8.4.5 Simulating ARMA models 203

8.4.6 Models with long-term memory 205

8.5 Time-varying volatility 206

8.5.1 The concepts 206

8.5.2 Autocorrelated time-varying volatility 208

8.5.3 Simulating GARCH processes 211

8.5.4 Selected further autoregressive volatility models 214

8.6 Adaptive expectations and patterns in price processes 218

8.6.1 Price–earnings models 218

8.6.2 Models with learning 220

8.7 Historical simulation 222

8.7.1 Backtesting 222

8.7.2 Bootstrap 223

8.8 Agent-based models and complexity 228

9. Financial simulation at work: some case studies 233

9.1 Constant proportion portfolio insurance (CPPI) 233

9.1.1 Basic concepts 233

9.1.2 Bootstrap 235

9.2 VaR estimation with Extreme Value Theory 237

9.2.1 Basic concepts 237

9.2.2 Scaling the data 238

9.2.3 Using Extreme Value Theory 238

9.3 Option pricing 242

9.3.1 Modeling prices 243

9.3.2 Pricing models 246

9.3.3 Greeks 258

9.3.4 Quasi-Monte Carlo 261

Part Three Optimization **269**

10. Optimization problems in finance 271

10.1 What to optimize? 271

10.2 Solving the model 273

10.2.1 Problems 273

10.2.2 Classical methods and heuristics 275

10.3 Evaluating solutions 276

10.4 Examples 278

Portfolio optimization with alternative risk measures 278

Model selection 280

Robust/resistant regression 280

Agent-based models 281

Calibration of option-pricing models 282

Calibration of yield structure models 283

10.5 Summary 284

11. Basic methods 287

11.1 Finding the roots of $f(x) = 0$ 287

11.1.1 A naïve approach 287

Graphical solution 288

Random search 289

11.1.2 Bracketing 289

11.1.3 Bisection 290

11.1.4 Fixed point method 292

Convergence 294

11.1.5 Newton's method 298

Comments 300

11.2 Classical unconstrained optimization 301

Convergence 302

11.3 Unconstrained optimization in one dimension 303

11.3.1 Newton's method 303

11.3.2 Golden section search 305

11.4 Unconstrained optimization in multiple dimensions 306

11.4.1 Steepest descent method 306

11.4.2 Newton's method 308

11.4.3 Quasi-Newton method 310

11.4.4 Direct search methods 311

11.4.5 Practical issues with Matlab 317

11.5 Nonlinear Least Squares 318

11.5.1 Problem statement and notation 318

11.5.2 Gauss–Newton method 320

11.5.3 Levenberg–Marquardt method 320

11.6 Solving systems of nonlinear equations $F(x) = 0$ 323

11.6.1 General considerations 323

11.6.2 Fixed point methods 325

11.6.3 Newton's method 328

11.6.4 Quasi-Newton methods 332

11.6.5 Further approaches 334

11.7 Synoptic view of solution methods 336

12. Heuristic methods in a nutshell 337

12.1 Heuristics 337

12.2 Trajectory methods 341

12.2.1 Stochastic local search 341

12.2.2 Simulated Annealing 342

12.2.3 Threshold Accepting 344

12.2.4 Tabu Search 344

12.3 Population-based methods 345

12.3.1 Genetic Algorithms 345

12.3.2 Differential Evolution 346

12.3.3 Particle Swarm Optimization 347

12.3.4 Ant Colony Optimization 348

12.4 Hybrids 349

12.5 Constraints 352

12.6 The stochastics of heuristic search 354

12.6.1 Stochastic solutions and computational resources 354

12.6.2 An illustrative experiment 356

12.7 General considerations 359

12.7.1 What technique to choose? 359

12.7.2 Efficient implementations 359

12.7.3 Parameter settings 362

12.8 Summary 363

12.A Implementing heuristic methods with Matlab 364

12.A.1 Threshold Accepting 367

12.A.2 Genetic Algorithm 371

12.A.3 Differential Evolution 375

12.A.4 Particle Swarm Optimization 377

13. Portfolio optimization 381

13.1 The investment problem 381

13.2 The classical case: mean–variance optimization 383

13.2.1 The model 383

13.2.2 Solving the model 385

13.2.3 Mean–variance models 386

13.2.4 True, estimated, and realized frontiers 393

13.2.5 Repairing matrices 396

13.3 Heuristic optimization of one-period models 403

13.3.1 Asset selection with local search 403

13.3.2 Scenario optimization with Threshold Accepting 409

13.3.3 Examples 421

13.3.4 Diagnostics 437

13.A More implementation issues in R 440

13.A.1 Scoping rules in R and objective functions 440

13.A.2 Vectorized objective functions 441

试读结束，需要全本PDF请购买 www.ertongbook.com

14. Econometric models 445
14.1 Term structure models 445
14.1.1 Yield curves 445
14.1.2 The Nelson–Siegel model 451
14.1.3 Calibration strategies 456
14.1.4 Experiments 474
14.2 Robust and resistant regression 480
14.2.1 The regression model 484
14.2.2 Estimation 486
14.2.3 An example 491
14.2.4 Numerical experiments 494
14.2.5 Final remarks 500
14.A Maximizing the Sharpe ratio 502

15. Calibrating option pricing models 505
15.1 Implied volatility with Black–Scholes 506
The smile 509
15.2 Pricing with the characteristic function 510
15.2.1 A pricing equation 510
15.2.2 Numerical integration 516
15.3 Calibration 541
15.3.1 Techniques 541
15.3.2 Organizing the problem and implementation 544
15.3.3 Two experiments 552
15.4 Final remarks 557
15.A Quadrature rules for infinity 558

Bibliography 563
Index 577

List of Algorithms

1	Forward substitution	37
2	Back-substitution	38
3	Cholesky factorization	42
4	Jacobi iteration	46
5	Gauss–Seidel iteration	46
6	SOR method	48
7	General structure of iterative methods	50
8	Block Jacobi method	52
9	Factorization of tridiagonal matrix	54
10	Forward and back-substitution for tridiagonal system	54
11	Least Squares with normal equations	63
12	Pseudocode for how \ works in Matlab	68
13	θ-method	82
14	Projected successive overrelaxation (PSOR)	92
15	Explicit payout method (EP)	93
16	European call for S, X, r, σ, T, and M time steps	106
17	Testing for early exercise: An American put	110
18	American call for S, X, r, σ, T, T_D, D and M time steps	112
19	Linear congruential random number generator	122
20	Simplified acceptance–rejection for standard normal variates	129
21	Taylor–Thompson algorithm	151
22	Generate variates with specific rank correlation	169
23	Metropolis algorithm	176
24	Metropolis–Hastings	178
25	Direct sampling	184
26	Monte Carlo simulation for a European call option	192
27	Agent-based simulation of prices in a market with chartists and fundamentalists	231
28	One-pass algorithm for computing the variance	250
29	Bracketing	290
30	Bisection	291
31	Fixed point iteration	292
32	Newton's method for zero finding	298
33	Newton's method for unconstrained optimization	304
34	Golden section search	305
35	Steepest descent method	307
36	Newton's method for unconstrained optimization in n dimensions	309
37	Quasi-Newton for unconstrained optimization in n dimensions	311
38	Nelder–Mead simplex direct search	316
39	Gauss–Newton method	320
40	Levenberg–Marquardt method	320

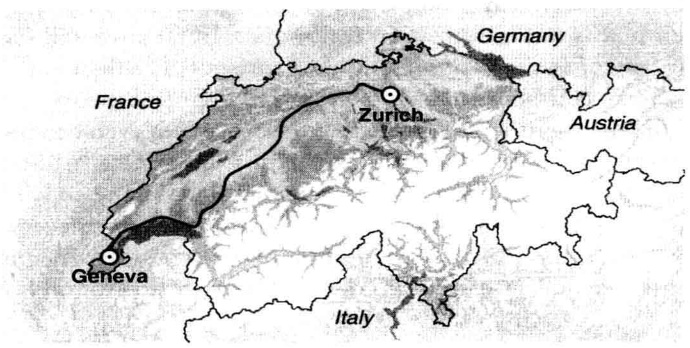
41 Jacobi, Gauss–Seidel, and SOR for nonlinear systems 326
42 Newton's method for nonlinear systems 329
43 Broyden's method for nonlinear systems 333
44 Local search 342
45 Simulated Annealing 343
46 Threshold Accepting 344
47 Tabu Search 345
48 Genetic Algorithms 346
49 Differential Evolution 347
50 Particle Swarm Optimization 348
51 Ant Colony Optimization 349
52 Repair mechanism for correlation matrix 400
53 Local Search for asset selection 404
54 Neighborhood (asset selection) 407
55 Threshold Accepting 417
56 Computing the threshold sequence—Variant 1 420
57 Computing the threshold sequence—Variant 2 420
58 Neighborhood (budget constraint) 422
59 Neighborhood (holding sizes) 423
60 Threshold Accepting—scenario updating 428
61 Neighborhood (budget constraint, cardinality constraints) 432
62 Differential Evolution for yield curve models 459
63 Particle Swarm Optimization for robust regression 487
64 Computing the prices for a given surface 537
65 Hybrid search 542
66 Repairing a solution x by reflection 551

CHAPTER ONE

Introduction

1.1. About this book

The origin of this book lies in the city of Geneva in Switzerland, so it seems only appropriate to start there. In case you do not know Switzerland, here is a map.



Imagine you stand at the wonderful lakefront of Geneva, with a view on Mont Blanc, and you decide to take a run to Lake Zurich. We admit that this is unlikely, given that the distance is 250 km. But it is just an example. Also, imagine you could run the whole distance with a speed of 20 km per hour which is about the speed that a world-class marathon runner could manage (we ignore the fact that Switzerland is quite a mountainous region). It would still take you more than 12 hours to arrive in Zurich. Now, imagine you took a plane with an average speed of 800 km per hour. This is about 40 times faster, and you would arrive in less than half an hour. What is our point? First, few people would run from Geneva to Zurich. But what we wanted to give you is an idea of speedup. It is just forty in this case, not a large number, but it makes a difference.

This book is not about traveling but about quantitative methods in a specific field, computational finance. Quantitative analysis has become ever more important in scientific research and the industry over the last decades,

© 2011 Elsevier Inc. All rights reserved.

and in our view, it has much changed. On the one hand, ever more data is collected and stored today, waiting to be analyzed. At the same time, computing power has increased dramatically. If we measure the speed of a computer by the number of operations it can perform in a second, then computers have improved by a factor of perhaps 1,000,000 since the early 1980s.[1] If traveling had matched this speedup, we could go from Geneva to Zurich in 4/100 of a second. Better yet, we do not talk about supercomputers here but about the kind of computers we have on our desktops. (And at least in the past, the power of what was called a supercomputer at one point in time was available for private users only a few years later.)

Computational finance is not a well-defined discipline. It is the intersection of financial economics, scientific computing, econometrics, software engineering, and many other fields. Its goal is to better understand prices and markets (if you are an academic), and to make money (if you are more practically inclined). This book is about tools for computational finance, with an emphasis on optimization techniques. In fact, we will focus on a particular type of optimization methods: heuristics. The theme of the book is the practical, computational application of financial theory; we do not deal with the theory itself; in some cases, there may not even be much established theory. Hence, we will not develop ideas theoretically but consider quantitative analysis as a primarily computational discipline, in which applications are put into software form and tested empirically. For every topic in this book, we discuss the implementation of methods; algorithms will be given in pseudocode, Matlab or R code will be provided for all important examples.[2]

The readers we have in mind are students (Master or PhD level) in programs on quantitative and computational finance, and researchers in finance but also practitioners in banks and other financial companies. The book could be the main reference for courses in computational finance programs or an additional reference for quantitative or mathematical finance courses. Many of the chapters have the flavor of case studies. From a pedagogical view, this allows students to learn the required steps for tackling specific problems; these steps can then be applied to other problems. From the practical side the selected problems will, we hope, be relevant enough to make the book a reference on how to implement a given technique. In fact, the ideal, prototypic kind of reader that we have in mind—let us call him "the analyst"—does not only work on theory, but his job requires a close interaction between theoretical ideas and computation with data.

[1] The first IBM PC in the early 1980s did about 10 or 20 Kflops (one flop is one floating point operation per second, see page 32). And this does not take into account improvements in algorithms.

[2] Matlab and R are sufficiently similar so that code from one language can be translated into the other.

The book is structured into three parts. The first part, "Fundamentals," begins with an introduction to numerical analysis, so we discuss computer arithmetic, approximation errors, how to solve linear equations, how to approximate derivatives, and other topics. These discussions will serve as a reference to which later chapters often come back. For instance, the Greeks for option pricing models can often be computed by finite differences. Numerical methods are rarely discussed in standard finance books (Brandimarte, 2006, is an exception), even though they are the basis for empirically applying and testing models. Beyond the initial chapters, our book will not discuss numerical methods in an abstract way. For a general treatment see, for instance, Heath (2005) or the concise and very recommended summary given by Trefethen (2008b); we will only discuss numerical aspects that are relevant to our applications. Further chapters will explain deterministic numerical methods to price instruments, namely trees and finite difference methods, and their application to standard products.

The second part, "Simulation," starts with chapters on how to generate random numbers and how to model dependence. Chapter 8 discusses how time series processes, simple assets, and portfolios can be simulated. There is also a short example on how to implement an agent-based model. By means of several case studies, applications will illustrate how these models can be used for generating more realistic price processes and how simulation can help to develop an intuitive understanding of econometric models and solve practical problems.

The third part, "Optimization" deals with optimization problems in finance. Many relevant optimization models cannot be solved with standard methods that are readily available in software packages. Hence, standard methods will only be briefly touched. We will rather show how to apply another class of techniques, so-called heuristics. Chapter 11 will detail building blocks of numerical optimization algorithms. Though this includes gradient-based techniques like Newton's method, emphasis will be put on more robust techniques like direct search. Chapter 12, "Heuristic Methods in a Nutshell," will then move from these building blocks to specific techniques, and give an overview of heuristic methods. The remaining chapters will deal with specific problems from portfolio optimization, the estimation of econometric models, and the calibration of option pricing models.

1.2. Principles

We start with some principles, or guidelines, for the applications in the book.

1. *We don't know much in finance.* No, we are not being modest here. "We" does not only refer to the authors, but to finance and the people working in finance in general (which includes us). A number of people may object

试读结束，需要全本PDF请购买 www.ertongbook.com