

<<金融中的数值方法和优化>>

图书基本信息

书名：<<金融中的数值方法和优化>>

13位ISBN编号：9787510052651

10位ISBN编号：7510052653

出版时间：2013-1

出版时间：世界图书出版公司

作者：吉利

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## <<金融中的数值方法和优化>>

### 内容概要

《金融中的数值方法和优化(英文)》旨在为读者介绍金融计算工具—基本数值分析和计算技巧，如期权定价、并突出了模拟和优化的重要性，用许多章讲述投资组合保险和风险估计问题。特别地，有几章用于讲述优化探索和如何将他们应用于投资组合的选择、估值的校准和期权定价模型。这些具体的例子让读者学习了解决问题的具体步骤，以及将这些步骤举一反三。同时，这些应用使得《金融中的数值方法和优化(英文)》的参考价值大大提高。

<<金融中的数值方法和优化>>

作者简介

作者：(瑞士)吉利 (M.Gilli)

## 书籍目录

List of Algorithms Acknowledgements 1. Introduction 1.1 About this book 1.2 Principles 1.3 on software 1.4 on approximations and accuracy 1.5 Summary: the theme of the book Part One Fundamentals 2. Numerical analysis in a nutshell 2.1 Computer arithmetic Representation of real numbers Machine precision Example of limitations of floating point arithmetic 2.2 Measuring errors 2.3 Approximating derivatives with finite differences Approximating first-order derivatives Approximating second-order derivatives Partial derivatives How to choose  $h$  Truncation error for forward difference 2.4 Numerical instability and ill-conditioning Example of a numerically unstable algorithm Example of an ill-conditioned problem 2.5 Condition number of a matrix Comments and examples 2.6 A primer on algorithmic and computational complexity 2.6.1 Criteria for comparison Order of complexity and classification 2.A Operation count for basic linear algebra operations 3. Linear equations and Least Squares problems Choice of method 3.1 Direct methods 3.1.1 Triangular systems 3.1.2 LU factorization 3.1.3 Cholesky factorization 3.1.4 QR decomposition 3.1.5 Singular value decomposition 3.2 Iterative methods 3.2.1 Jacobi, Gauss-Seidel, and SOR Successive overrelaxation 3.2.2 Convergence of iterative methods 3.2.3 General structure of algorithms for iterative methods 3.2.4 Block iterative methods 3.3 Sparse linear systems 3.3.1 Tridiagonal systems 3.3.2 Irregular sparse matrices 3.3.3 Structural properties of sparse matrices 3.4 The Least Squares problem 3.4.1 Method of normal equations 3.4.2 Least Squares via QR factorization 3.4.3 Least Squares via SVD decomposition 3.4.4 Final remarks The backslash operator in Matlab 4. Finite difference methods 4.1 An example of a numerical solution A first numerical approximation A second numerical approximation 4.2 Classification of differential equations 4.3 The Black-Scholes equation 4.3.1 Explicit, implicit, and  $\theta$ -methods 4.3.2 Initial and boundary conditions and definition of the grid 4.3.3 Implementation of the  $\theta$ -method with Matlab 4.3.4 Stability 4.3.5 Coordinate transformation of space variables 4.4 American options 4.A A note on Matlab's function `spdiags` 5. Binomial trees 5.1 Motivation Matching moments 5.2 Growing the tree 5.2.1 Implementing a tree 5.2.2 Vectorization 5.2.3 Binomial expansion 5.3 Early exercise 5.4 Dividends 5.5 The Greeks Greeks from the tree Part Two Simulation 6. Generating random numbers 6.1 Monte Carlo methods and sampling 6.1.1 How it all began 6.1.2 Financial applications 6.2 Uniform random number generators 6.2.1 Congruential generators 6.2.2 Mersenne Twister 6.3 Nonuniform distributions 6.3.1 The inversion method 6.3.2 Acceptance-rejection method 6.4 Specialized methods for selected distributions 6.4.1 Normal distribution 6.4.2 Higher order moments and the Cornish-Fisher expansion 6.4.3 Further distributions 6.5 Sampling from a discrete set 6.5.1 Discrete uniform selection 6.5.2 Roulette wheel selection 6.5.3 Random permutations and shuffling 6.6 Sampling errors—and how to reduce them 6.6.1 The basic problem 6.6.2 Quasi-Monte Carlo 6.6.3 Stratified sampling 6.6.4 Variance reduction 6.7 Drawing from empirical distributions 6.7.1 Data randomization 6.7.2 Bootstrap 6.8 Controlled experiments and experimental design 6.8.1 Replicability and ceteris paribus analysis 6.8.2 Available random number generators in Matlab 6.8.3 Uniform random numbers from Matlab's `rand` function 6.8.4 Gaussian random numbers from Matlab's `randn` function 6.8.5 Remedies 7. Modeling dependence 7.1 Transformation methods 7.1.1 Linear correlation 7.1.2 Rank correlation 7.2 Markov chains 7.2.1 Concepts 7.2.2 The Metropolis algorithm 7.3 Copula models 7.3.1 Concepts 7.3.2 Simulation using copulas 8. A gentle introduction to financial simulation 8.1 Setting the stage 8.2 Single-period simulations 8.2.1 Terminal asset prices 8.2.2 I-over-N portfolios 8.2.3 European options 8.2.4 VaR of a covered put portfolio 8.3 Simple price processes 8.4 Processes with memory in the levels of returns 8.4.1 Efficient versus adaptive markets 8.4.2 Moving averages 8.4.3 Autoregressive models 8.4.4 Autoregressive moving average (ARMA) models 8.4.5 Simulating ARMA models 8.4.6 Models with long-term memory 8.5 Time-varying volatility 8.5.1 The concepts 8.5.2 Autocorrelated time-varying volatility 8.5.3 Simulating GARCH processes 8.5.4 Selected further autoregressive volatility models 8.6 Adaptive expectations and patterns in price processes 8.6.1 Price-earnings models 8.6.2 Models with learning 8.7 Historical simulation 8.7.1 Backtesting 8.7.2 Bootstrap 8.8 Agent-based models and complexity 9. Financial simulation at work: some case studies 9.1 Constant proportion portfolio insurance (CPPI) 9.1.1 Basic concepts 9.1.2 Bootstrap 9.2 VaR estimation with Extreme Value Theory 9.2.1 Basic concepts 9.2.2 Scaling the data 9.2.3 Using Extreme Value Theory 9.3 Option pricing

9.3.1 Modeling prices 9.3.2 Pricingmodels 9.3.3 Greeks 9.3.4 Quasi-Monte Carlo Part Three Optimization

## 章节摘录

版权页：插图： When looking at these examples more closely, one can already spot some limitations of these methods for financial applications. The optimal solution for  $n + 1$  is usually not found by taking the solution for  $n$  and adding one extra point. Likewise, choosing the values for high  $n$  but using only some of them ( e.g., because one runs out of time ) will lead to a ( systematic ) bias. Hence, the designer has to decide in advance how many samples he or she wants to evaluate. Determining the draws for the multivariate case can become a tough optimization problem in its own right—even ( and in particular ) when the draws must be orthogonal ( i.e., uncorrelated ) in all dimensions. Fang, Tang, Maringer, and Winker ( 2006 ) provide bounds and show how heuristics can help in tackling this problem. Repeated experiments generate exactly the same sequence. On the one hand, this is good as it simplifies replicability. On the other hand, this is not so good because if an interesting spot in the range is missed once, then repeated experiments will not cover it either. Even more importantly, it is difficult to judge how stable the results are: Results with pseudorandom numbers will vary from experiment to experiment, but should converge with increasing sample size. If results are close together, this could indicate stability; if they are all over the place, then the results are obviously not very robust. With quasi-Monte Carlo numbers, the results will be identical unless one changes  $n$ ; and unless  $n$  changes dramatically ( in particular when  $n$  is already high ) , the variations in the points could be modest. In finance, one is often interested in extreme risks and rare events, and generating draws with QMC might lead to biased and unreliable results.

## <<金融中的数值方法和优化>>

### 编辑推荐

《金融中的数值方法和优化(英文)》由世界图书出版社出版。

<<金融中的数值方法和优化>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>