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
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
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1.3 Cholesky factorization</td>
<td>40</td>
</tr>
<tr>
<td>3.1.4 QR decomposition</td>
<td>43</td>
</tr>
<tr>
<td>3.1.5 Singular value decomposition</td>
<td>44</td>
</tr>
<tr>
<td><strong>3.2 Iterative methods</strong></td>
<td>44</td>
</tr>
<tr>
<td>3.2.1 Jacobi, Gauss–Seidel, and SOR</td>
<td>45</td>
</tr>
<tr>
<td>Successive overrelaxation</td>
<td>46</td>
</tr>
<tr>
<td>3.2.2 Convergence of iterative methods</td>
<td>48</td>
</tr>
<tr>
<td>3.2.3 General structure of algorithms for iterative methods</td>
<td>49</td>
</tr>
<tr>
<td>3.2.4 Block iterative methods</td>
<td>52</td>
</tr>
<tr>
<td><strong>3.3 Sparse linear systems</strong></td>
<td>53</td>
</tr>
<tr>
<td>3.3.1 Tridiagonal systems</td>
<td>53</td>
</tr>
<tr>
<td>3.3.2 Irregular sparse matrices</td>
<td>55</td>
</tr>
<tr>
<td>3.3.3 Structural properties of sparse matrices</td>
<td>57</td>
</tr>
<tr>
<td><strong>3.4 The Least Squares problem</strong></td>
<td>60</td>
</tr>
<tr>
<td>3.4.1 Method of normal equations</td>
<td>61</td>
</tr>
<tr>
<td>3.4.2 Least Squares via QR factorization</td>
<td>64</td>
</tr>
<tr>
<td>3.4.3 Least Squares via SVD decomposition</td>
<td>65</td>
</tr>
<tr>
<td>3.4.4 Final remarks</td>
<td>67</td>
</tr>
<tr>
<td>The backslash operator in Matlab</td>
<td>67</td>
</tr>
<tr>
<td><strong>4. Finite difference methods</strong></td>
<td>69</td>
</tr>
<tr>
<td>4.1 An example of a numerical solution</td>
<td>69</td>
</tr>
<tr>
<td>A first numerical approximation</td>
<td>70</td>
</tr>
<tr>
<td>A second numerical approximation</td>
<td>71</td>
</tr>
<tr>
<td>4.2 Classification of differential equations</td>
<td>73</td>
</tr>
<tr>
<td><strong>4.3 The Black–Scholes equation</strong></td>
<td>74</td>
</tr>
<tr>
<td>4.3.1 Explicit, implicit, and $\theta$-methods</td>
<td>76</td>
</tr>
<tr>
<td>4.3.2 Initial and boundary conditions and definition of the grid</td>
<td>76</td>
</tr>
<tr>
<td>4.3.3 Implementation of the $\theta$-method with Matlab</td>
<td>82</td>
</tr>
<tr>
<td>4.3.4 Stability</td>
<td>85</td>
</tr>
<tr>
<td>4.3.5 Coordinate transformation of space variables</td>
<td>88</td>
</tr>
<tr>
<td><strong>4.4 American options</strong></td>
<td>90</td>
</tr>
<tr>
<td>4.4.1 A note on Matlab’s function <code>spdiags</code></td>
<td>101</td>
</tr>
<tr>
<td><strong>5. Binomial trees</strong></td>
<td>103</td>
</tr>
<tr>
<td>5.1 Motivation</td>
<td>103</td>
</tr>
<tr>
<td>Matching moments</td>
<td>104</td>
</tr>
<tr>
<td>5.2 Growing the tree</td>
<td>105</td>
</tr>
<tr>
<td>5.2.1 Implementing a tree</td>
<td>106</td>
</tr>
<tr>
<td>5.2.2 Vectorization</td>
<td>107</td>
</tr>
<tr>
<td>5.2.3 Binomial expansion</td>
<td>108</td>
</tr>
<tr>
<td><strong>5.3 Early exercise</strong></td>
<td>110</td>
</tr>
<tr>
<td><strong>5.4 Dividends</strong></td>
<td>111</td>
</tr>
<tr>
<td><strong>5.5 The Greeks</strong></td>
<td>113</td>
</tr>
<tr>
<td>Greeks from the tree</td>
<td>113</td>
</tr>
</tbody>
</table>
## Contents

### 8. A gentle introduction to financial simulation

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

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

10. Optimization problems in finance 271
  10.1 What to optimize?
12. Heuristic methods in a nutshell

12.1 Heuristics

12.2 Trajectory methods
   12.2.1 Stochastic local search
   12.2.2 Simulated Annealing
   12.2.3 Threshold Accepting
   12.2.4 Tabu Search

12.3 Population-based methods
   12.3.1 Genetic Algorithms
   12.3.2 Differential Evolution
   12.3.3 Particle Swarm Optimization
   12.3.4 Ant Colony Optimization

12.4 Hybrids

12.5 Constraints

12.6 The stochastics of heuristic search
   12.6.1 Stochastic solutions and computational resources
   12.6.2 An illustrative experiment

12.7 General considerations
   12.7.1 What technique to choose?
   12.7.2 Efficient implementations
   12.7.3 Parameter settings

12.8 Summary

12.A Implementing heuristic methods with Matlab
   12.A.1 Threshold Accepting
   12.A.2 Genetic Algorithm
   12.A.3 Differential Evolution
   12.A.4 Particle Swarm Optimization

13. Portfolio optimization

13.1 The investment problem

13.2 The classical case: mean–variance optimization
   13.2.1 The model
   13.2.2 Solving the model
   13.2.3 Mean–variance models
   13.2.4 True, estimated, and realized frontiers
   13.2.5 Repairing matrices

13.3 Heuristic optimization of one-period models
   13.3.1 Asset selection with local search
   13.3.2 Scenario optimization with Threshold Accepting
   13.3.3 Examples
   13.3.4 Diagnostics

13.A More implementation issues in R
   13.A.1 Scoping rules in R and objective functions
   13.A.2 Vectorized objective functions