

Object-Oriented Implementation of Numerical Methods

An Introduction with Java and Smalltalk



Object-Oriented Implementation of Numerical Methods

An Introduction with Java and Smalltalk

Didier H. Besset

DB Consulting



MORGAN KAUFMANN PUBLISHERS

AN IMPRINT OF ACADEMIC PRESS

A Harcourt Science and Technology Company

SAN FRANCISCO SAN DIEGO NEW YORK BOSTON
LONDON SYDNEY TOKYO

Acquisitions Editor Diane D. Cerra
Production Editor Howard Severson
Production Assistant Mei Levenson
Editorial Assistant Stacie Pierce
Cover Design Yvo Reizebos
Cover Image ???
Text Design, Composition, and Illustrations Windfall Software, using ZzTeX
Copyeditor Laura Larson
Proofreader Ken DellaPenta
Indexer Bill Meyers
Printer Courier Corporation

Designations used by companies to distinguish their products are often claimed as trademarks or registered trademarks. In all instances where Morgan Kaufmann Publishers is aware of a claim, the product names appear in initial capital or all capital letters. Readers, however, should contact the appropriate companies for more complete information regarding trademarks and registration.

ACADEMIC PRESS
A Harcourt Science and Technology Company
525 B Street, Suite 1900, San Diego, CA 92101-4495, USA
<http://www.academicpress.com>

Academic Press
Harcourt Place, 32 Jamestown Road, London, NW1 7BY, United Kingdom
<http://www.academicpress.com>

Morgan Kaufmann Publishers
340 Pine Street, Sixth Floor, San Francisco, CA 94104-3205
<http://www.mkp.com>

© 2001 by Academic Press
All rights reserved
Printed in the United States of America

05 04 03 02 01 5 4 3 2 1

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means—electronic, mechanical, photocopying, recording, or otherwise—without the prior written permission of the publisher.

Library of Congress Cataloging-in-Publication Data

Beset, Didier H.
Object-oriented implementation of numerical methods : an introduction with Java and Smalltalk / Didier H. Beset
p. cm.
ISBN 1-55860-679-3
1. Computer software—Development. 2. Object-oriented programming (Computer science). I. Title.
QA76.76.D47 B475 2001
005.1'17—dc21 00-063399

This book is printed on acid-free paper.

Code Listings xvii

Preface xxv

1 Introduction 1

- 1.1 Object-Oriented Paradigm and Mathematical Objects 2
- 1.2 Object-Oriented Concepts in a Nutshell 3
- 1.3 Dealing with Numerical Data 4
 - 1.3.1 *Floating Point Representation* 4
 - 1.3.2 *Rounding Errors* 6
 - 1.3.3 *Real Example of Rounding Error* 7
 - 1.3.4 *Outsmarting Rounding Errors* 8
 - 1.3.5 *Wisdom from the Past* 9
- 1.4 Finding the Numerical Precision of a Computer 10
 - 1.4.1 *Computer Numerical Precision—General Implementation* 12
 - 1.4.2 *Computer Numerical Precision—Smalltalk Implementation* 12
 - 1.4.3 *Computer Numerical Precision—Java Implementation* 16
- 1.5 Comparing Floating-Point Numbers 22
 - 1.5.1 *Comparing Floating-Point Numbers—Smalltalk Code* 23
- 1.6 Speed Consideration 23
 - 1.6.1 *Smalltalk Particular* 25
- 1.7 Conventions 25
 - 1.7.1 *Class Diagrams* 26
 - 1.7.2 *Smalltalk Code* 27
 - 1.7.3 *Java Code* 27
- 1.8 Road Map 28

2 Function Evaluation 33

- 2.1 Function Concept 35
 - 2.1.1 *Function—Smalltalk Implementation* 35
 - 2.1.2 *Function—Java Implementation* 36
- 2.2 Polynomials 39
 - 2.2.1 *Mathematical Definitions* 39
 - 2.2.2 *Polynomials—General Implementation* 41
 - 2.2.3 *Polynomials—Smalltalk Implementation* 42
 - 2.2.4 *Polynomials—Java implementation* 48
- 2.3 Error Function 56
 - 2.3.1 *Mathematical Definitions* 56
 - 2.3.2 *Error Function—Smalltalk Implementation* 58
 - 2.3.3 *Error Function—Java Implementation* 60
- 2.4 Gamma Function 62
 - 2.4.1 *Mathematical Definitions* 62
 - 2.4.2 *Gamma Function—Smalltalk Implementation* 64
 - 2.4.3 *Gamma Function—Java Implementation* 66
- 2.5 Beta Function 69
 - 2.5.1 *Mathematical Definitions* 69
 - 2.5.2 *Beta Function—Smalltalk Implementation* 70
 - 2.5.3 *Beta Function—Java Implementation* 70

3 Interpolation 73

- 3.1 General Remarks 74
 - 3.1.1 *Interpolation Concepts and Examples* 74
- 3.2 Lagrange Interpolation 78
 - 3.2.1 *Lagrange Interpolation—Smalltalk Implementation* 80
 - 3.2.2 *Lagrange Interpolation—Java Implementation* 83
- 3.3 Newton Interpolation 88
 - 3.3.1 *Newton Interpolation—General Implementation* 88
 - 3.3.2 *Newton Interpolation—Smalltalk Implementation* 89
 - 3.3.3 *Newton Interpolation—Java Implementation* 90
- 3.4 Neville Interpolation 92
 - 3.4.1 *Neville Interpolation—General Implementation* 93
 - 3.4.2 *Neville Interpolation—Smalltalk Implementation* 94

3.4.3	<i>Neville Interpolation—Java Implementation</i>	96
3.5	Bulirsch-Stoer Interpolation	98
3.5.1	<i>Bulirsch-Stoer Interpolation—General Implementation</i>	100
3.5.2	<i>Bulirsch-Stoer Interpolation—Smalltalk Implementation</i>	100
3.5.3	<i>Bulirsch-Stoer Interpolation—Java Implementation</i>	101
3.6	Cubic Spline Interpolation	102
3.6.1	<i>Cubic Spline Interpolation—General Implementation</i>	103
3.6.2	<i>Cubic Spline Interpolation—Smalltalk Implementation</i>	104
3.6.3	<i>Cubic Spline Interpolation—Java Implementation</i>	107
3.7	Which Method to Choose?	110

4 Iterative Algorithms 113

4.1	Successive Approximations	113
4.1.1	<i>Iterative Process—Smalltalk Implementation</i>	117
4.1.2	<i>Iterative Process—JAVA Implementation</i>	121
4.2	Evaluation with Relative Precision	126
4.2.1	<i>Relative Precision—Smalltalk Implementation</i>	126
4.2.2	<i>Relative Precision—Java Implementation</i>	128
4.3	Examples	130

5 Finding the Zero of a Function 133

5.1	Introduction	133
5.2	Finding the Zeroes of a Function—Bisection Method	134
5.2.1	<i>Bisection Algorithm—General Implementation</i>	136
5.2.2	<i>Bisection Algorithm—Smalltalk Implementation</i>	136
5.2.3	<i>Bisection Algorithm—Java Implementation</i>	138
5.3	Finding the Zero of a Function—Newton's Method	140
5.3.1	<i>Newton's Method—Smalltalk Implementation</i>	142
5.3.2	<i>Newton's Method—Java Implementation</i>	144
5.4	Example of Zero Finding—Roots of Polynomials	147
5.4.1	<i>Roots of Polynomials—Smalltalk Implementation</i>	148
5.4.2	<i>Roots of Polynomials—Java Implementation</i>	149
5.5	Which Method to Choose?	150

6 Integration of Functions **153**

- 6.1 Introduction 153
- 6.2 General Framework—Trapeze Integration Method 155
 - 6.2.1 *End Game Strategy* 157
 - 6.2.2 *Trapeze Integration—General Implementation* 157
 - 6.2.3 *Trapeze Integration—Smalltalk Implementation* 158
 - 6.2.4 *Trapeze Integration—Java Implementation* 160
- 6.3 Simpson Integration Algorithm 162
 - 6.3.1 *Simpson Integration—General Implementation* 163
 - 6.3.2 *Simpson Integration—Smalltalk Implementation* 163
 - 6.3.3 *Simpson Integration—Java Implementation* 164
- 6.4 Romberg Integration Algorithm 165
 - 6.4.1 *Romberg Integration—General Implementation* 166
 - 6.4.2 *Romberg Integration—Smalltalk Implementation* 167
 - 6.4.3 *Romberg Integration—Java Implementation* 168
- 6.5 Evaluation of Open Integrals 170
 - 6.5.1 *Bag of Tricks* 170
- 6.6 Which Method to Choose? 171
 - 6.6.1 *Smalltalk Comparison* 171
 - 6.6.2 *Java Comparison* 173

7 Series **177**

- 7.1 Introduction 177
- 7.2 Infinite Series 178
 - 7.2.1 *Infinite Series—Smalltalk Implementation* 181
 - 7.2.2 *Infinite Series—Java Implementation* 183
- 7.3 Continued Fractions 184
 - 7.3.1 *Continued Fractions—Smalltalk Implementation* 185
 - 7.3.2 *Continued Fractions—Java Implementation* 186
- 7.4 Incomplete Gamma Function 188
 - 7.4.1 *Mathematical Definitions* 189
 - 7.4.2 *Incomplete Gamma Function—Smalltalk Implementation* 190
 - 7.4.3 *Incomplete Gamma Function—Java Implementation* 193
- 7.5 Incomplete Beta Function 198
 - 7.5.1 *Mathematical Definitions* 199

7.5.2 *Incomplete Beta Function—Smalltalk Implementation* 199

7.5.3 *Incomplete Beta Function—Java Implementation* 202

8 Linear Algebra 207

8.1 Vectors and Matrices 209

8.1.1 *Vector and Matrix—Smalltalk Implementation* 212

8.1.2 *Vector and Matrix—Java Implementation* 222

8.2 Linear Equations 242

8.2.1 *Backward Substitution* 243

8.2.2 *Gaussian Elimination* 244

8.2.3 *Fine Points* 245

8.2.4 *Linear Equations—General Implementation* 246

8.2.5 *Linear Equations—Smalltalk Implementation* 246

8.2.6 *Linear Equations—Java Implementation* 250

8.3 LUP Decomposition 257

8.3.1 *LUP Decomposition—General Implementation* 259

8.3.2 *LUP Decomposition—Smalltalk Implementation* 260

8.3.3 *LUP Decomposition—Java Implementation* 264

8.4 Computing the Determinant of a Matrix 272

8.4.1 *Computing the Determinant of Matrix—General Implementation* 273

8.4.2 *Computing the Determinant of Matrix—Smalltalk Implementation* 273

8.4.3 *Computing the Determinant of Matrix—Java Implementation* 274

8.5 Matrix Inversion 274

8.5.1 *Implementation Strategy* 275

8.5.2 *Matrix Inversion—Smalltalk Implementation* 277

8.5.3 *Matrix Inversion—Java Implementation* 279

8.5.4 *Matrix Inversion—Rounding Problems* 287

8.6 Matrix Eigenvalues and Eigenvectors of a Nonsymmetrical Matrix 288

8.6.1 *Finding the Largest Eigenvalue—General Implementation* 289

8.6.2 *Finding the Largest Eigenvalue—Smalltalk Implementation* 290

8.6.3 *Finding the Largest Eigenvalue—Java Implementation* 292

8.7 Matrix Eigenvalues and Eigenvectors of a Symmetrical Matrix 296

8.7.1 *Jacobi's Algorithm* 299

- 8.7.2 *Jacobi's Algorithm—General Implementation* 300
- 8.7.3 *Jacobi's Algorithm—Smalltalk Implementation* 300
- 8.7.4 *Jacobi's Algorithm—Java Implementation* 305

9 Elements of Statistics 311

- 9.1 Statistical Moments 311
 - 9.1.1 *Statistical Moments—General Implementation* 314
 - 9.1.2 *Statistical Moments—Smalltalk Implementation* 315
 - 9.1.3 *Statistical Moments—Java Implementation* 317
- 9.2 Robust Implementation of Statistical Moments 320
 - 9.2.1 *Robust Central Moments—General Implementation* 322
 - 9.2.2 *Robust Central Moments—Smalltalk Implementation* 322
 - 9.2.3 *Robust Central Moments—Java Implementation* 326
- 9.3 Histograms 331
 - 9.3.1 *Histograms—General Implementation* 333
 - 9.3.2 *Histograms—Smalltalk Implementation* 333
 - 9.3.3 *Histograms—Java Implementation* 343
- 9.4 Random Number Generator 358
 - 9.4.1 *Linear Congruential Random Generators* 358
 - 9.4.2 *Additive Sequence Generators* 359
 - 9.4.3 *Bit-Pattern Generators* 360
 - 9.4.4 *Random Number Generator—Smalltalk Implementation* 361
 - 9.4.5 *Random Number Generator—Java Implementation* 366
- 9.5 Probability Distributions 369
 - 9.5.1 *Probability Distributions—General Implementation* 371
 - 9.5.2 *Probability Distributions—Smalltalk Implementation* 371
 - 9.5.3 *Probability Distributions—Java Implementation* 376
- 9.6 Normal Distribution 384
 - 9.6.1 *Normal Distribution—Smalltalk Implementation* 384
 - 9.6.2 *Normal Distribution—Java Implementation* 387
- 9.7 Gamma Distribution 392
 - 9.7.1 *Gamma Distribution—Smalltalk Implementation* 392
 - 9.7.2 *Gamma Distribution—Java Implementation* 397
- 9.8 Experimental Distribution 403
 - 9.8.1 *Experimental Distribution—General Implementation* 403
 - 9.8.2 *Experimental Distribution—Smalltalk Implementation* 404

9.8.3 *Experimental Distribution—Java Implementation* 406

10 Statistical Analysis 411

- 10.1 *F*-Test and the Fisher-Snedecor Distribution 413
 - 10.1.1 *Fisher-Snedecor Distribution—Smalltalk Implementation* 414
 - 10.1.2 *Fisher-Snedecor Distribution—Java Implementation* 419
- 10.2 *t*-Test and the Student Distribution 424
 - 10.2.1 *Student Distribution—Smalltalk Implementation* 426
 - 10.2.2 *Student Distribution—Java Implementation* 431
- 10.3 χ^2 -Test and χ^2 Distribution 436
 - 10.3.1 χ^2 Distribution—Smalltalk Implementation 439
 - 10.3.2 χ^2 Distribution—Java Implementation 440
 - 10.3.3 *Weighted-Point Implementation* 442
- 10.4 χ^2 -Test on Histograms 448
 - 10.4.1 χ^2 -Test on Histograms—Smalltalk Implementation 450
 - 10.4.2 χ^2 -Test on Histograms—Java Implementation 453
- 10.5 Definition of Estimation 456
 - 10.5.1 *Maximum-Likelihood Estimation* 457
 - 10.5.2 *Least-Square Estimation* 457
- 10.6 Least-Square Fit with Linear Dependence 459
- 10.7 Linear Regression 460
 - 10.7.1 *Linear Regression—General Implementation* 462
 - 10.7.2 *Linear Regression—Smalltalk Implementation* 462
 - 10.7.3 *Linear Regression—Java Implementation* 466
- 10.8 Least-Square Fit with Polynomials 471
 - 10.8.1 *Polynomial Least-Square Fits—Smalltalk Implementation* 474
 - 10.8.2 *Polynomial Least-Square Fits—Java Implementation* 478
- 10.9 Least-Square Fit with Nonlinear Dependence 482
 - 10.9.1 *Nonlinear Fit—General Implementation* 484
 - 10.9.2 *Nonlinear Fit—Smalltalk Implementation* 485
 - 10.9.3 *Nonlinear Fit—Java Implementation* 489
- 10.10 Maximum-Likelihood Fit of a Probability Density Function 499
 - 10.10.1 *Maximum-Likelihood Fit—General Implementation* 501
 - 10.10.2 *Maximum-Likelihood Fit—Smalltalk Implementation* 502
 - 10.10.3 *Maximum-Likelihood Fit—Java Implementation* 505

11 Optimization 511

- 11.1 Introduction 514
- 11.2 Extended Newton Algorithms 515
- 11.3 Hill-Climbing Algorithms 516
 - 11.3.1 *Optimizing—General Implementation* 517
 - 11.3.2 *Common Optimizing Classes—Smalltalk Implementation* 518
 - 11.3.3 *Common Optimizing Classes—Java Implementation* 524
- 11.4 Optimizing in One Dimension 540
 - 11.4.1 *Optimizing in One Dimension—Smalltalk Implementation* 540
 - 11.4.2 *Optimizing in One Dimension—Java Implementation* 543
- 11.5 Bracketing the Optimum in One Dimension 546
 - 11.5.1 *Bracketing the Optimum—Smalltalk Implementation* 546
 - 11.5.2 *Bracketing the Optimum—Java Implementation* 548
- 11.6 Powell's Algorithm 550
 - 11.6.1 *Powell's Algorithm—General Implementation* 551
 - 11.6.2 *Powell's Algorithm—Smalltalk Implementation* 552
 - 11.6.3 *Powell's Algorithm—Java Implementation* 554
- 11.7 Simplex Algorithm 558
 - 11.7.1 *Simplex Algorithm—General Implementation* 560
 - 11.7.2 *Simplex Algorithm—Smalltalk Implementation* 560
 - 11.7.3 *Simplex Algorithm—Java Implementation* 563
- 11.8 Genetic Algorithm 568
 - 11.8.1 *Mapping the Search Space on Chromosomes* 569
 - 11.8.2 *Genetic Algorithm—General Implementation* 569
 - 11.8.3 *Genetic Algorithm—Smalltalk Implementation* 573
 - 11.8.4 *Genetic Algorithm—Java Abstract Implementation* 578
 - 11.8.5 *Genetic Algorithm—Java Implementation with Vectors* 585
- 11.9 Multiple Strategy Approach 592
 - 11.9.1 *Multiple Strategy Approach—General Implementation* 593

12 Data Mining 599

- 12.1 Data Server 601
 - 12.1.1 *Data Server—Smalltalk Implementation* 602
 - 12.1.2 *Data Server—Java Implementation* 603
- 12.2 Covariance and Covariance Matrix 605

12.2.1	<i>Using Covariance Information</i>	607
12.2.2	<i>Covariance Matrix—General Implementation</i>	607
12.2.3	<i>Covariance Matrix—Smalltalk Implementation</i>	608
12.2.4	<i>Covariance Matrix—Java Implementation</i>	611
12.3	<i>Multidimensional Probability Distribution</i>	616
12.4	<i>Covariance Data Reduction</i>	617
12.5	<i>Mahalanobis Distance</i>	617
12.5.1	<i>Examples of Use</i>	618
12.5.2	<i>Mahalanobis Distance—General Implementation</i>	619
12.5.3	<i>Mahalanobis Distance—Smalltalk Implementation</i>	620
12.5.4	<i>Mahalanobis Distance—Java Implementation</i>	622
12.6	<i>Cluster Analysis</i>	625
12.6.1	<i>Algorithm Details</i>	625
12.6.2	<i>Cluster Analysis—General Implementation</i>	627
12.6.3	<i>Cluster Analysis—Smalltalk Implementation</i>	629
12.6.4	<i>Cluster Analysis—Java Implementation</i>	634
12.7	<i>Covariance Clusters</i>	643
12.7.1	<i>Covariance Clusters—General Implementation</i>	643

App. A *Decimal Floating-Point Simulation* **647**

App. B *Smalltalk Primer for Java Programmers* **651**

B.1	<i>Syntax in a Nutshell</i>	651
B.1.1	<i>Smalltalk Expressions</i>	651
B.1.2	<i>Precedence</i>	652
B.1.3	<i>Assignment, Equality, and Identity</i>	653
B.2	<i>Class and Methods</i>	653
B.2.1	<i>Instance Methods</i>	654
B.2.2	<i>Class Methods</i>	655
B.2.3	<i>Block Context</i>	656
B.3	<i>Iterator Methods</i>	657
B.3.1	<i>do:</i>	657
B.3.2	<i>collect:</i>	657
B.3.3	<i>inject:into:</i>	657
B.4	<i>Double Dispatching</i>	658
B.5	<i>Multiple Dispatching</i>	660

App. C Java Primer for Smalltalk Programmers 661

- C.1 Remarks on the Syntax 661
 - C.1.1 *Classes* 662
 - C.1.2 *Instance Variables* 662
 - C.1.3 *Method Declaration and Method Calling* 662
 - C.1.4 *Objects and Nonobjects* 663
 - C.1.5 *Packages* 663
 - C.1.6 *Scope qualifiers* 664
 - C.1.7 *Static Qualifier* 666
- C.2 Abstract Class and Interface 666
- C.3 Exception Handling 668
- C.4 Collections and Related Topics 669

App. D Additional Probability Distributions 671

- D.1 Beta Distribution 671
 - D.1.1 *Beta Distribution—Smalltalk Implementation* 671
 - D.1.2 *Beta Distribution—Java Implementation* 675
- D.2 Cauchy Distribution 680
 - D.2.1 *Cauchy Distribution—Smalltalk Implementation* 681
 - D.2.2 *Cauchy Distribution—Java Implementation* 683
- D.3 Exponential Distribution 687
 - D.3.1 *Exponential Distribution—Smalltalk Implementation* 688
 - D.3.2 *Exponential Distribution—Java Implementation* 690
- D.4 Fisher-Tippett Distribution 694
 - D.4.1 *Fisher-Tippett Distribution—Smalltalk Implementation* 694
 - D.4.2 *Fisher-Tippett Distribution—Java Implementation* 698
- D.5 Laplace Distribution 702
 - D.5.1 *Laplace Distribution—Smalltalk Implementation* 702
 - D.5.2 *Laplace Distribution—Java Implementation* 705
- D.6 Log Normal Distribution 710
 - D.6.1 *Log Normal Distribution—Smalltalk Implementation* 710
 - D.6.2 *Log Normal Distribution—Java Implementation* 713
- D.7 Triangular Distribution 716
 - D.7.1 *Triangular Distribution—Smalltalk Implementation* 716
 - D.7.2 *Triangular Distribution—Java Implementation* 719

- D.8 Uniform Distribution 723
 - D.8.1 *Uniform Distribution—Smalltalk Implementation* 723
 - D.8.2 *Uniform Distribution—Java Implementation* 725
- D.9 Weibull Distribution 728
 - D.9.1 *Weibull Distribution—Smalltalk Implementation* 730
 - D.9.2 *Weibull Distribution—Java Implementation* 732

App. E Accurate Accumulation of Expectation Values 737

- E.1 Accurate Accumulation of Central Moments 737
- E.2 Accurate Accumulation of the Covariance 739

App. F Accompanying CD ROM 741

- F.1 Smalltalk code 741
 - F.1.1 *non-ENVY users* 741
 - F.1.2 *ENVY users* 743
- F.2 Java code 744
 - F.2.1 *Visual Age for Java* 744
 - F.2.2 *Other Java systems* 744
- F.3 Utilities 745
 - F.3.1 *Distribution demo* 745
 - F.3.2 *File reader* 745

References 747

Index 749

┌

┐

—

—

└

- 1**
 - 1.1 Smalltalk code to find the machine precision 12
 - 1.2 Java code to find the machine precision 16
 - 1.3 Comparison of floating-point numbers in Smalltalk 23

- 2**
 - 2.1 Java implementation of the interface `OneVariableFunction` 36
 - 2.2 Java implementation of a generic derivative evaluation 37
 - 2.3 Smalltalk implementation of the polynomial class 43
 - 2.4 Method of class `Number` related to polynomials 47
 - 2.5 Java implementation of the polynomial class 48
 - 2.6 Smalltalk implementation of the Error function 59
 - 2.7 Java implementation of the Error function (partial listing) 61
 - 2.8 Smalltalk implementation of the gamma function 64
 - 2.9 Java implementation of the gamma function 67
 - 2.10 Smalltalk implementation of the beta function 70

- 3**
 - 3.1 Smalltalk implementation of the Lagrange interpolation 81
 - 3.2 Java interface for point series 83
 - 3.3 A possible concrete implementation of the interface `PointSeries` 84
 - 3.4 Java implementation of the Lagrange interpolation 86
 - 3.5 Smalltalk implementation of the Newton interpolation 89
 - 3.6 Java implementation of the Newton interpolation 91
 - 3.7 Smalltalk implementation of Neville's algorithm 94
 - 3.8 Java implementation of Neville interpolation 96
 - 3.9 Smalltalk implementation of Bulirsch-Stoer interpolation 100
 - 3.10 Java implementation of Bulirsch-Stoer interpolation 101
 - 3.11 Smalltalk implementation of cubic spline interpolation 104
 - 3.12 Java implementation of cubic spline interpolation 107

4	4.1	Smalltalk implementation of an iterative process	119
	4.2	Java implementation of an iterative process	123
	4.3	Smalltalk implementation of the class <code>DhbFunctionalIterator</code>	127
	4.4	Java implementation of the class <code>FunctionalIterator</code>	129
5	5.1	Smalltalk implementation of the bisection algorithm	137
	5.2	Java implementation of a generic derivative evaluation	139
	5.3	Smalltalk implementation of Newton's zero-finding method	143
	5.4	Java implementation of Newton's zero-finding method	145
	5.5	Smalltalk implementation of finding the roots of a polynomial	148
6	6.1	Smalltalk implementation of trapeze integration	159
	6.2	Java implementation of trapeze integration	161
	6.3	Smalltalk implementation of the Simpson integration algorithm	164
	6.4	Java implementation of the Simpson integration algorithm	164
	6.5	Smalltalk implementation of Romberg integration	167
	6.6	Java implementation of Romberg integration	168
	6.7	Smalltalk comparison script for integration algorithms	172
	6.8	Java comparison method for integration algorithms	173
7	7.1	Smalltalk implementation of an infinite series	181
	7.2	Smalltalk implementation of a term server	182
	7.3	Java implementation of an infinite series	183
	7.4	Smalltalk implementation of a continued fraction	186
	7.5	Java implementation of an infinite series	186
	7.6	Smalltalk implementation of the incomplete gamma function	191
	7.7	Smalltalk implementation of the series term server for the incomplete gamma function	192
	7.8	Smalltalk implementation of the fraction term server for the incomplete gamma function	193
	7.9	Java implementation of the incomplete gamma function	194
	7.10	Java implementation of the infinite series term server for the incomplete gamma function	196
	7.11	Java implementation of the fraction term server for the incomplete gamma function	197
	7.12	Smalltalk implementation of the incomplete beta function	200
	7.13	Smalltalk implementation of the term server for the incomplete beta function	201
	7.14	Java implementation of the incomplete beta function	202

- 7.15 Java implementation of the continued fraction for the incomplete beta function 204

8

- 8.1 Vector class in Smalltalk 214
- 8.2 Matrix classes in Smalltalk 217
- 8.3 Symmetrical matrix classes in Smalltalk 221
- 8.4 Mathematical vector class in Java 224
- 8.5 Matrix class in Java 232
- 8.6 Smalltalk implementation of a system of linear equations 247
- 8.7 Java implementation of a system of linear equations 252
- 8.8 Smalltalk implementation of the LUP decomposition 261
- 8.9 Java implementation of the LUP decomposition 266
- 8.10 Smalltalk methods to compute a matrix determinant 273
- 8.11 Smalltalk implementation of matrix inversion 277
- 8.12 Java implementation of the class `SymmetricMatrix` 280
- 8.13 Smalltalk implementation of the search for the largest eigenvalue 291
- 8.14 Java implementation of the search for the largest eigenvalue 293
- 8.15 Smalltalk implementation of Jacobi's algorithm 301
- 8.16 Java implementation of Jacobi's algorithm 306

9

- 9.1 Smalltalk fast implementation of statistical moments 315
- 9.2 Java implementation of statistical moments 317
- 9.3 Smalltalk implementation of accurate statistical moments 322
- 9.4 Smalltalk implementation of accurate statistical moments with fixed orders 325
- 9.5 Java implementation of accurate statistical moments 326
- 9.6 Java implementation of accurate statistical moments with fixed order 330
- 9.7 Smalltalk implementation of histograms 336
- 9.8 Java implementation of histograms 345
- 9.9 Smalltalk implementation of congruential random number generators 363
- 9.10 Smalltalk implementation of an additive sequence random number generator 364
- 9.11 Smalltalk implementation of random number generators 366
- 9.12 Java implementation of an additive sequence random number generator 367
- 9.13 Smalltalk implementation of a probability distribution 372
- 9.14 Smalltalk implementation of a probability distribution with unknown distribution function 375
- 9.15 Smalltalk implementation of a probability distribution function 376
- 9.16 Java implementation of a probability distribution 377

- 9.17 Java implementation of a probability distribution with unknown distribution 382
- 9.18 Java implementation of a probability distribution function 383
- 9.19 Smalltalk implementation of the normal distribution 384
- 9.20 Java implementation of the normal distribution 387
- 9.21 Smalltalk implementation of the gamma distribution 394
- 9.22 Java implementation of the gamma distribution 397
- 9.23 Smalltalk implementation of an experimental distribution 404
- 9.24 Java implementation of an experimental distribution 406

10

- 10.1 Smalltalk implementation of the Fisher-Snedecor distribution 416
- 10.2 Smalltalk implementation of the F -test 418
- 10.3 Java implementation of the Fisher-Snedecor distribution 420
- 10.4 Smalltalk implementation of the Student distribution 427
- 10.5 Smalltalk implementation of the t -test 430
- 10.6 Java implementation of the Student distribution 432
- 10.7 Smalltalk implementation of the χ^2 distribution 439
- 10.8 Java implementation of the χ^2 distribution 440
- 10.9 Smalltalk implementation of the weighted point class 444
- 10.10 Java implementation of the weighted-point class 446
- 10.11 Smalltalk implementation of a scaled probability density function 450
- 10.12 Smalltalk implementation of χ^2 -test on histograms 452
- 10.13 Java implementation of a scaled probability density function 454
- 10.14 Smalltalk implementation of linear regression 463
- 10.15 Java implementation of linear regression 467
- 10.16 Smalltalk implementation of a polynomial least-square fit 475
- 10.17 Smalltalk implementation of a polynomial with error 477
- 10.18 Java implementation of a polynomial least square fit 479
- 10.19 Java implementation of a polynomial with error 481
- 10.20 Smalltalk implementation of a nonlinear least-square fit 486
- 10.21 Java parametrized function interface 490
- 10.22 Java implementation of a nonlinear least-square fit 493
- 10.23 Smalltalk implementation of a maximum-likelihood fit 503
- 10.24 Java implementation of a maximum-likelihood fit 506

11

- 11.1 Smalltalk classes common to all optimizing classes 518
- 11.2 Smalltalk abstract class for all optimizing classes 520
- 11.3 Smalltalk projected function classes 522

- 11.4 Java optimizing point classes 524
- 11.5 Java optimizing vector classes 527
- 11.6 Java optimizing point factory classes 532
- 11.7 Java abstract class for all optimizing classes 534
- 11.8 Java projected function class 537
- 11.9 Smalltalk golden section optimum finder 541
- 11.10 Java implementation of the golden section optimum search 544
- 11.11 Smalltalk optimum bracket finder 547
- 11.12 Java optimum bracket finder 548
- 11.13 Smalltalk implementation of Powell's algorithm 552
- 11.14 Java implementation of Powell's algorithm 555
- 11.15 Smalltalk implementation of simplex algorithm 560
- 11.16 Java implementation of simplex algorithm 563
- 11.17 Smalltalk chromosome: abstract and concrete 573
- 11.18 Smalltalk implementation of genetic algorithm 576
- 11.19 Java abstract implementation of a chromosome 579
- 11.20 Java implementation of genetic algorithm 582
- 11.21 Java implementation of a vector chromosome 585
- 11.22 Java implementation of genetic algorithm for vectors 589
- 11.23 Smalltalk implementation of a general optimizer 593
- 11.24 Java implementation of a general optimizer 595

12

- 12.1 Smalltalk abstract data server 602
- 12.2 Smalltalk memory-based data server 602
- 12.3 Java abstract data server 603
- 12.4 Java memory-based data server 604
- 12.5 Smalltalk implementation of vector average 609
- 12.6 Smalltalk implementation of covariance matrix 610
- 12.7 Java implementation of vector average 611
- 12.8 Java implementation of covariance matrix 613
- 12.9 Smalltalk Mahalanobis center 621
- 12.10 Java Mahalanobis center 623
- 12.11 Smalltalk K -cluster algorithm 630
- 12.12 Smalltalk implementation of a Euclidean cluster 633
- 12.13 Java K -cluster algorithm 635
- 12.14 Java implementation of a Euclidean cluster 641
- 12.15 Smalltalk covariance cluster 643
- 12.16 Java covariance cluster 644

A

- A.1 Smalltalk code simulating decimal floating-number arithmetic 648

C

- C.1 Inheritance of private methods 664

D

- D.1 Smalltalk implementation of the beta distribution 671
D.2 Java implementation of the beta distribution 675
D.3 Smalltalk implementation of the Cauchy distribution 681
D.4 Java implementation of the Cauchy distribution 684
D.5 Smalltalk implementation of the exponential distribution 689
D.6 Java implementation of the exponential distribution 691
D.7 Smalltalk implementation of the Fisher-Tippett distribution 696
D.8 Java implementation of the Fisher-Tippett distribution 698
D.9 Smalltalk implementation of the Laplace distribution 704
D.10 Java implementation of the Laplace distribution 705
D.11 Smalltalk implementation of the log normal distribution 710
D.12 Java implementation of the log normal distribution 713
D.13 Smalltalk implementation of the triangular distribution 716
D.14 Java implementation of the triangular distribution 720
D.15 Smalltalk implementation of the uniform distribution 723
D.16 Java implementation of the uniform distribution 725
D.17 Smalltalk implementation of the Weibull distribution 730
D.18 Java implementation of the Weibull distribution 732

Preface

*Si je savais une chose utile à ma nation qui fût ruineuse à une autre,
je ne la proposerais pas à mon prince,
parce que je suis homme avant d'être Français,
parce que je suis nécessairement homme,
et que je ne suis Français que par hasard.¹*

—Charles de Montesquieu

When I first encountered object-oriented programming I immediately became highly enthusiastic about it, mainly because of my mathematical inclination. After all, I learned to use computers as a high-energy physicist. In mathematics, a new, high order concept is always based on previously defined, simpler concepts. Once a property is demonstrated for a given concept, it can be applied to any new concept sharing the same premises as the original one. In object-oriented language, this is called *reuse* and *inheritance*. Thus, numerical algorithms using mathematical concepts that can be mapped directly into objects.

This book is intended for object-oriented programmers who need to implement numerical methods in their applications. The algorithms exposed here are mostly fundamental numerical algorithms with a few advanced ones. The purpose of the book is to show that implementing these algorithms in an object-oriented language is feasible—and quite easily feasible. I expect readers to be able to implement their own favorite numerical algorithm after seeing the examples discussed in this book.

The scope of the book is limited. It is not a Bible about numerical algorithms. Such Bible-like books already exist and are quoted throughout the chapters. Instead, I wanted to illustrate mapping between mathematical concepts and computer objects. I have limited the book to algorithms, which I have implemented and used in real applications over 12 years of object-oriented programming. Thus, the reader can be certain that the algorithms have been tested in the field.

Because the book's intent is to show numerical methods to object-oriented programmers, the code presented here is described in depth. Each algorithm is presented with the same organization. First the necessary equations are introduced with short

1. If I knew some trade useful to my country but that would ruin another, I would not disclose it to my ruler, because I am a man before being French, because I belong to mankind while I am French only by a twist of fate.

explanations. This book is not about mathematics, so explanations are kept to a minimum. Then the general object-oriented architecture of the algorithm is presented. Finally, because this book is intended to be a practical one, the code implementation is exposed. First, I describe how to use it, for readers who are just interested in using the algorithm, and then I discuss and present the code implementation.

The question was which language to use?

I decided to select two languages: Smalltalk and Java. Java is an obvious choice today since the world seems to lean toward Java. In spite of all the current emphasis on Java, I personally prefer Smalltalk for any practical use. But, as the computer industry evolves, and because I am anything but a language fanatic, the reader must be aware that this last statement may no longer be true by the time these lines are read. New computer languages will certainly combine object-oriented features with novel concepts, and I am ready to try any serious new contender.

Having stated my preference for Smalltalk does not mean that I don't like Java. Java has brought forth interesting innovations and certainly paved a golden way for object-oriented programming. I personally have gained a lot of insight in learning Java.

The code in this book has been written in both languages and is presented in separate sections. My hope is that Smalltalk and Java programmers will be exposed to the other language by reading this book. If they keep an open mind and do indeed read both code sections, they certainly will gain something from the comparison as I did myself in writing them.

As far as possible, each algorithm is presented with examples of use. I did not want to build contrived examples and instead have used examples personally encountered in my professional life. Some people may think that some examples are coming from esoteric domains. This is not so. Each example has been selected for its generality. The reader should study each example regardless of the field of application and concentrate on the universal aspects of it.

Acknowledgments

I wish to express my thanks to the many people with whom I had interactions about the object-oriented approach—Smalltalk and Java in particular—on the various electronic forums. One special person is Kent Beck, whose controversial statements raised hell and started spirited discussions. I also thank Kent for showing me tricks about the Refactoring Browser and eXtreme Programming. I also would like to thank Eric Clayberg for pulling me out of a ditch more than once and for making such fantastic Smalltalk tools.

A special mention goes to Donald Knuth for being an inspiration to me and many other programmers with his series of books *The Art of Computer Programming* and for making this wonderful typesetting program T_EX. This book was typeset with T_EX, L^AT_EX, and ZzT_EX.

Furthermore, I would like to give credit to a few people without whom this book would never have been published. First, Joseph Pelrine persuaded me that what I was doing was worth sharing with the rest of the object-oriented community. I express

my most sincere thanks to the reviewers who toiled on the early manuscripts of this book. Without their open-mindedness, it never would have made it to the publisher.

Special thanks go to David N. Smith for triggering interesting thoughts about random number generators and to William Leo for checking the equations. Another special mention goes to An Lam, who did a terrific job at checking the orthodoxy of the Java code.

Finally, my immense gratitude is due to Stephane Ducasse of the University of Bern, who checked the orthodoxy of the Smalltalk code and did an excellent job of rendering the early manuscript into not only readable but entertaining prose.

┌

┐

└