Contents

Preface xxi

Acknowledgments xxv

Acronyms xxvii

Part I GENERALITIES

1 Introduction 3

1.1 The Nature of Computational Science 3

1.1.1 How Computational Scientists Do It 4

1.2 Aims of This Book 5

1.3 Using this Book with the Disk and Web 6

2 Computing Software Basics 9

2.1 Problem 1: Making Computers Obey 9

2.2 Theory: Computer Languages 9

2.3 Implementation: Programming Concepts 11

2.4 Implementation: Fortran, area.f 12

2.5 Implementation: C, area.c 13

2.6 Implementation: Shells, Editors, and Programs 13

2.7 Theory: Program Design 14
2.8 Method: Structured Programming 16
2.9 Method: Programming Hints 18
2.10 Problem 2: Limited Range of Numbers 20
2.11 Theory: Number Representation 20
2.12 Method: Fixed and Floating 21
2.13 Implementation: Over/Underflows, over.f (.c) 23
2.14 Model: Machine Precision 23
2.15 Implementation: limit.f (.c) 24
2.16 Problem 3: Complex Numbers, Inverse Functions 25
2.17 Theory: Complex Numbers 25
2.18 Implementation: complex.c (.f) 27
2.19 Exploration: Quantum Complex Energies 28
2.20 Problem 4: Summing Series 28
2.21 Method: Numeric 29
2.22 Implementation: Pseudocode 29
2.23 Implementation: Good Algorithm, exp-good.f (.c) 29
2.24 Implementation: Bad Algorithm, exp-bad.f (.c) 30
2.25’ Assessment 30

3 Errors and Uncertainties in Computations 31
3.1 Problem: Living with Errors 31
3.2 Theory: Types of Errors 32
3.3 Model: Subtractive Cancellation 33
3.4 Assessment: Cancellation Experiment 34
3.5 Model: Multiplicative Errors 36
3.6 Problem 1: Errors in Spherical Bessel Functions 37
3.7 Method: Numeric Recursion Relations 38
3.8 Implementation: bessel.f (.c) 40
3.9 Assessment 40
3.10 Problem 2: Errors in Algorithms 40
3.11 Model: Errors in Algorithms 41
3.11.1 Total Error 41
3.12 Method: Optimizing with Known Error Behavior 42
3.13 Method: Empirical Error Analysis 43
3.14 Assessment: Experiment 44

4 Integration 47
4.1 Problem: Integrating a Spectrum 47

4.2 Model: Quadrature, Summing Boxes 47
4.3 Method: Trapezoid Rule 50
4.4 Method: Simpson's Rule 51
4.5 Assessment: Integration Error, Analytic 52
4.6 Method: Gaussian Quadrature 55
4.6.1 Scaling Integration Points 56
4.7 Implementation: integ.f (.c) 57
4.8 Assessment: Empirical Error Estimate 58
4.9 Assessment: Experimentation 59
4.10 Method: Romberg Extrapolation 59
4.10.1 Other Closed Newton–Cotes Formulas 60

Part II APPLICATIONS

5 Data Fitting 63
5.1 Problem: Fitting an Experimental Spectrum 63
5.2 Theory: Curve Fitting 64
5.3 Method: Lagrange Interpolation 65
5.3.1 Example 66
5.4 Implementation: lagrange.f (.c) 66
5.5 Assessment: Interpolating a Resonant Spectrum 67
5.6 Assessment: Exploration 68
5.7 Method: Cubic Splines 68
5.7.1 Cubic Spline Boundary Conditions 69
5.7.2 Exploration: Cubic Spline Quadrature 70
5.8 Implementation: spline.f 71
5.9 Assessment: Spline Fit of Cross Section 71
5.10 Problem: Fitting Exponential Decay 71
5.11 Model: Exponential Decay 71
5.12 Theory: Probability Theory 73
5.13 Method: Least-Squares Fitting 74
5.14 Theory: Goodness of Fit 76
5.15 Implementation: Least-Squares Fits, fit.f (.c) 77
5.16 Assessment: Fitting Exponential Decay 78
5.17 Assessment: Fitting Heat Flow 79
5.18 Implementation: Linear Quadratic Fits 80
5.19 Assessment: Quadratic Fit 81
5.20 Method: Nonlinear Least-Squares Fitting 81

5.21 Assessment: Nonlinear Fitting 82

6 Deterministic Randomness 83

6.1 Problem: Deterministic Randomness 83
6.2 Theory: Random Sequences 83
6.3 Method: Pseudo-Random-Number Generators 84
6.4 Assessment: Random Sequences 86
6.5 Implementation: Simple and Not random.f (.c); call.f (.c) 87
6.6 Assessment: Randomness and Uniformity 87
6.7 Assessment: Tests of Randomness, Uniformity 88
6.8 Problem: A Random Walk 89
6.9 Model: Random Walk Simulation 89
6.10 Method: Numerical Random Walk 90
6.11 Implementation: walk.f (.c) 91
6.12 Assessment: Different Random Walkers 91

7 Monte Carlo Applications 93

7.1 Problem: Radioactive Decay 93
7.2 Theory: Spontaneous Decay 93
7.3 Model: Discrete Decay 94
7.4 Model: Continuous Decay 95
7.5 Method: Decay Simulation 95
7.6 Implementation: decay.f (.c) 97
7.7 Assessment: Decay Visualization 97
7.8 Problem: Measuring by Stone Throwing 97
7.9 Theory: Integration by Rejection 97
7.10 Implementation: Stone Throwing, pond.f (.c) 98
7.11 Problem: High-Dimensional Integration 99
7.12 Method: Integration by Mean Value 99
7.12.1 Multidimensional Monte Carlo 101
7.13 Assessment: Error in N-D Integration 101
7.14 Implementation: 10-D Integration, int_10d.f (.c) 101
7.15 Problem: Integrate a Rapidly Varying Function@ 102
7.16 Method: Variance Reduction@ 102
7.17 Method: Importance Sampling@ 103
7.18 Implementation: Nonuniform Randomness@ 103
7.18.1 Inverse Transform Method 103
7.18.2 Uniform Weight Function w 104

7.18.3 Exponential Weight 105
7.18.4 Gaussian (Normal) Distribution 105
7.18.5 Alternate Gaussian Distribution 106
7.19 Method: von Neumann Rejection@ 107
7.20 Assessment@ 108

8 Differentiation 109
8.1 Problem 1: Numerical Limits 109
8.2 Method: Numeric 109
8.2.1 Method: Forward Difference 109
8.2.2 Method: Central Difference 111
8.2.3 Method: Extrapolated Difference 111
8.3 Assessment: Error Analysis 112
8.4 Implementation: Differentiation, diff.f (.c) 113
8.5 Assessment: Error Analysis, Numerical 113
8.6 Problem 2: Second Derivatives 114
8.7 Theory: Newton II 114
8.8 Method: Numerical Second Derivatives 114
8.9 Assessment: Numerical Second Derivatives 115

9 Differential Equations and Oscillations 117
9.1 Problem: A Forced Nonlinear Oscillator 117
9.2 Theory, Physics: Newton's Laws 117
9.3 Model: Nonlinear Oscillator 118
9.4 Theory, Math: Types of Equations 119
9.4.1 Order 119
9.4.2 Ordinary and Partial 120
9.4.3 Linear and Nonlinear 121
9.4.4 Initial and Boundary Conditions 121
9.5 Theory, Math, and Physics: The Dynamical Form for ODEs 122
9.5.1 Second-Order Equation 122
9.6 Implementation: Dynamical Form for Oscillator 123
9.7 Numerical Method: ODE Algorithms 124
9.8 Method (Numerical): Euler's Algorithm 124
9.9 Method (Numerical): Second-Order Runge-Kutta 126
9.10 Method (Numerical): Fourth-Order Runge-Kutta 127
9.11 Implementation: ODE Solver, rk4.f (.c) 127
9.12 Assessment: rk4 and Linear Oscillations 128

9.13 Assessment: rk4 and Nonlinear Oscillations 128
9.14 Exploration: Energy Conservation 129

10 Quantum Eigenvalues; Zero-Finding and Matching 131
10.1 Problem: Binding A Quantum Particle 131
10.2 Theory: Quantum Waves 132
10.3 Model: Particle in a Box 133
10.4 Solution: Semianalytic 133
10.5 Method: Finding Zero via Bisection Algorithm 136
10.6 Method: Eigenvalues from an ODE Solver 136
10.6.1 Matching 138
10.7 Implementation: ODE Eigenvalues, numerov.c 139
10.8 Assessment: Explorations 141
10.9 Extension: Newton's Rule for Finding Roots 141

11 Anharmonic Oscillations 143
11.1 Problem 1: Perturbed Harmonic Oscillator 143
11.2 Theory: Newton II 144
11.3' Implementation: ODE Solver, rk4.f (.c) 144
11.4 Assessment: Amplitude Dependence of Frequency 145
11.5 Problem 2: Realistic Pendulum 145
11.6 Theory: Newton II for Rotations 146
11.7 Method, Analytic: Elliptic Integrals 147
11.8 Implementation, rk4 for Pendulum 147
11.9 Exploration: Resonance and Beats 148
11.10 Exploration: Phase-Space Plot 149
11.11 Exploration: Damped Oscillator 150

12 Fourier Analysis of Nonlinear Oscillations 151
12.1 Problem 1: Harmonics in Nonlinear Oscillations 151
12.2 Theory: Fourier Analysis 152
12.2.1 Example 1: Sawtooth Function 154
12.2.2 Example 2: Half-Wave Function 154
12.3 Assessment: Summation of Fourier Series 155
12.4 Theory: Fourier Transforms 156
12.5 Method: Discrete Fourier Transform 157
12.6 Method: DFT for Fourier Series 161
12.7 Implementation: fourier.f (.c), invfour.c 162
12.8 Assessment: Simple Analytic Input 162

12.9 Assessment: Highly Nonlinear Oscillator 163
12.10 Assessment: Nonlinearly Perturbed Oscillator 163
12.11 Exploration: DFT of Nonperiodic Functions 163
12.12 Exploration: Processing Noisy Signals 164
12.13 Model: Autocorrelation Function 164
12.14 Assessment: DFT and Autocorrelation Function 165
12.15 Problem 2: Model Dependence of Data Analysis ⊙ 166
12.16 Method: Model-Independent Data Analysis 167
12.17 Assessment 168

13 Unusual Dynamics of Nonlinear Systems 171
13.1 Problem: Variability of Bug Populations 171
13.2 Theory: Nonlinear Dynamics 171
13.3 Model: Nonlinear Growth, The Logistic Map 172
13.3.1 The Logistic Map 173
13.4 Theory: Properties of Nonlinear Maps 174
13.4.1 Fixed Points 174
13.4.2 Period Doubling, Attractors 175
13.5 Implementation: Explicit Mapping 176
13.6 Assessment: Bifurcation Diagram 177
13.7 Implementation: bugs.f (.c) 178
13.8 Exploration: Random Numbers via Logistic Map 179
13.9 Exploration: Feigenbaum Constants 179
13.10 Exploration: Other Maps 180

14 Differential Chaos in Phase Space 181
14.1 Problem: A Pendulum Becomes Chaotic 181
14.2 Theory and Model: The Chaotic Pendulum 182
14.3 Theory: Limit Cycles and Mode Locking 183
14.4 Implementation 1: Solve ODE, rk4.f (.c) 184
14.5 Visualization: Phase-Space Orbits 184
14.6 Implementation 2: Free Oscillations 187
14.7 Theory: Motion in Phase Space 188
14.8 Implementation 3: Chaotic Pendulum 188
14.9 Assessment: Chaotic Structure in Phase Space 191
14.10 Assessment: Fourier Analysis 191
14.11 Exploration: Pendulum with Vibrating Pivot 192
14.11.1 Implementation: Bifurcation Diagram 192
14.12 Further Explorations 193

Part III APPLICATIONS
(HIGH PERFORMANCE COMPUTING)

15 Matrix Computing and Subroutine Libraries 197
15.1 Problem 1: Many Simultaneous Linear Equations 197
15.2 Formulation: Linear into Matrix Equation 198
15.3 Problem 2: Simple but Unsolvable Statics 198
15.4 Theory: Statics 199
15.5 Formulation: Nonlinear Simultaneous Equations 199
15.6 Theory: Matrix Problems 200
15.6.1 Classes of Matrix Problems 201
15.7 Method: Matrix Computing 203
15.8 Implementation: Scientific Libraries, WWW 207
15.9 Implementation: Determining Availability 209
15.9.1 Determining Contents of a Library 210
15.9.2 Determining the Needed Routine 210
15.9.3 Calling LAPACK from Fortran, lineq.c 211
15.9.4 Calling LAPACK from C 212
'15.9.5 Calling LAPACK Fortran from C 213
15.9.6 C Compiling Calling Fortran 214
15.10 Extension: More Netlib Libraries 214
15.10.1 SLATEC's Common Math Library 215
15.11 Exercises: Testing Matrix Calls 216
15.12 Implementation: LAPACK Short Contents 217
15.13 Implementation: Netlib Short Contents 221
15.14 Implementation: SLATEC Short Contents 222

16 Bound States in Momentum Space 231
16.1 Problem: Bound States in Nonlocal Potentials 231
16.2 Theory: k-Space Schrödinger Equation 232
16.3 Method: Reducing Integral to Linear Equations 233
16.4 Model: The Delta-Shell Potential 235
16.5 Implementation: Binding Energies, bound-c (.f) 236
16.6 Exploration: Wave Function 237

17 **Quantum Scattering via Integral Equations@** 239
17.1 Problem: Quantum Scattering in k Space 239
17.2 Theory: Lippmann-Schwinger Equation 240
17.3 Theory (Mathematics): Singular Integrals 241

17.3.1 Numerical Principal Values 242
17.4 Method: Converting Integral to Matrix Equations 243
17.4.1 Solution via Inversion or Elimination 245
17.4.2 Solving $i\epsilon$ Integral Equations ⊙ 245
17.5 Implementation: Delta-Shell Potential, scatt.f 246
17.6 Exploration: Scattering Wave Function 248

18 Computing Hardware Basics: Memory and CPU 249
18.1 Problem: Speeding Up Your Program 249
18.2 Theory: High-Performance Components 250
18.2.1 Memory Hierarchy 250
18.2.2 The Central Processing Unit 253
18.2.3 CPU Design: RISC 254
18.2.4 CPU Design: Vector Processing 254
18.2.5 Virtual Memory 255
18.3 Method: Programming for Virtual Memory 256
18.4 Implementation: Good, Bad Virtual Memory Use 257
18.5 Method: Programming for Data Cache 258
18.6 Implementation 1: Cache Misses 260
18.7 Implementation 2: Cache Flow 260
18.8 Implementation 3: Large Matrix Multiplication 261

19 High-Performance Computing: Profiling and Tuning 263
19.1 Problem: Effect of Hardware on Performance 263
19.2 Method: Tabulating Speedups 263
19.3 Implementation 1: Baseline Program, tune.f 264
19.4 Method: Profiling 265
19.5 Implementation 2: Basic Optimization, tune1 .f 266
19.6 Implementation 2: Vector Tuning, tune2.f 269
19.7 Implementation 3: Vector Code on RISC, tune3.f 270
19.8 Implementation 4: Superscalar Tuning, tune4.f 271
19.9 Assessment 273

20 Parallel Computing and PVM 275
20.1 Problem: Speeding Up Your Program 275
20.2 Theory: Parallel Semantics 276
20.2.1 Parallel Instruction and Data Streams 276
20.2.2 Granularity 277
20.2.3 Parallel Performance 277

20.3 Method: Multitasking Programming 278
20.3.1 Method: Multitask Organization 278
20.4 Method: Distributed Memory Programming 279
20.5 Implementation: PVM Bug Populations, WWW 280
20.5.1 The Plan 280

21 Object-Oriented Programming: Kinematics ⊙ 283
21.1 Problem: Superposition of Motions 283
21.2 Theory: Object-Oriented Programming 284
21.2.1 OOP Fundamentals 284
21.3 Theory: Newton's Laws, Equation of Motion 285
21.4 OOP Method: Class Structure 285
21.5 Implementation: Uniform 1 -D Motion, unimld.cpp 286
21.5.1 Uniform Motion in I-D, Class Um1D 287
21.5.2 Implementation: Uniform Motion in 2-D, Child Um2D, unimot2d.cpp 288
21.5.3 Class Um2D: Uniform Motion in 2-D 289
'21.5.4 Implementation: Projectile Motion, Child Accm2D, accm2d.cpp 291
2 1.5.5 Accelerated Motion in Two Directions 293
21.6 Assessment: Exploration, shms.cpp 295

22 Thermodynamic Simulations: The Ising Model 297
22.1 Problem: Hot Magnets 297
22.2 Theory: Statistical Mechanics 297
22.3 Model: An Ising Chain 299
22.4 Solution, Analytic 301
22.5 Solution, Numerical: The Metropolis Algorithm 301
22.6 Implementation: ising.f (.c) 304
22.7 Assessment: Approach to Thermal Equilibrium 305
22.8 Assessment: Thermodynamic Properties 305
22.9 Exploration: Beyond Nearest Neighbors 307
22.10 Exploration: 2-D and 3-D Ising Models 307

23 Functional Integration on Quantum Paths@ 309
23.1 Problem: Relate Quantum to Classical Trajectory 309
23.2 Theory: Feynman's Spacetime Propagation 309
23.3 Analytic Method: Bound-State Wave Function 312

23.4 Numerical Method: Lattice Path Integration 314
23.5 Method: Lattice Computation of Propagators 316
23.6 Implementation: qmc.f (.c) 319
23.7 Assessment and Exploration 321

24 Fractals 323
24. I Problem: Fractals 323
24.2 Theory: Fractional Dimension 324
24.3 Problem 1: The Sierpinski Gasket 324
24.4 Implementation: sierpin.c 325
24.5 Assessment: Determining a Fractal Dimension 326
24.6 Problem 2: How to Grow Beautiful Plants 327
24.7 Theory: Self-Affine Connections 328
24.8 Implementation: Bamsley's Fern, fern.c 329
24.9 Exploration: Self-Affinity in Trees 329
24.10 Implementation: Nice Trees, tree.c 330
24.11 Problem 3: Ballistic Deposition 330
24.12 Method: Random Depositions 331
24.13 Implementation: Ballistic Deposition, film.c 333
24.14 Problem 4: Length of the Coastline of Britain 333
24.15 Model: The Coast as a Fractal 333
24.16 Method: Box Counting 334
24.17 Problem 5: Correlated Growth, Forests and Films 335
24.18 Method: Correlated Ballistic Deposition 336
24.19 Implementation: column.c 337
24.20 Problem 6: A Globular Cluster 337
24.21 Model: Diffusion-Limited Aggregation 337
24.22 Method: DLA Simulation 338
24.23 Implementation: dla.c 339
24.24 Assessment: Fractal Analysis of DLA Graph 339
24.25 Problem 7: Fractals in Bifurcation Graph 340

Part IV PARTIAL DIFFERENTIAL EQUATIONS

25 Electrostatic Potentials 343
25.1 Introduction: Types of PDEs 343
25.2 Problem: Determining an Electrostatic Potential 344
25.3 Theory: Laplace's Equation (Elliptic PDE) 344

25.4 Method, Numerical: Finite Difference 345
25.5 Method, Analytic: Polynomial Expansions 347
25.6 Implementation: Solution on Lattice, laplace.f 350
25.7 Assessment and Visualization 350
25.8 Exploration 351
25.9 Exploration: Parallel-Plate Capacitor 351
25.10 Exploration: Field Between Square Conductors 353

26 Heat Flow 355
26.1 Problem: Heat Flow in a Metal Bar 355
26.2 Model: The Heat (Parabolic) PDE 355
26.3 Method, Analytic: Polynomial Expansions 356
26.4 Method, Numerical: Finite Difference 358
26.5 Analytic Assessment: Algorithm 358
26.6 Implementation: Heat Equation, eqheat.f (.c) 360
26.7 Assessment: Continuity, Numeric vs Analytic 361
26.8 Assessment: Visualization 361
26.9 Exploration 362

27 Waves on a String 365
27.1 Problem: A Vibrating String 365
27.2 Model: The Wave Equation (Hyperbolic PDE) 365
27.3 Method, Numerical: Time Stepping 367
27.4 Method, Analytic: Normal Modes 368
27.5 Implementation: eqstring.f (.c) 370
27.6 Assessment: Visualization 371
27.7 Exploration 371

28 Solitons, the KdeV Equation ⊙ 379
28.1 Introduction 379
28.2 Problem: Solitons 380
28.3 Theory: The Korteweg-de Vries Equation 381
28.4 Method, Analytic: Traveling Waves 384
28.5 Method, Numeric: Finite Difference 384
28.6 Implementation: KdeV Solitons, soliton.f (.c) 386
28.7 Assessment: Visualization 387
28.8 Exploration: Two Solitons Crossing 387
28.9 Exploration: Phase-Space Behavior 388
28.10 Exploration: Shock Waves 388

29 Sine–Gordon Solitons ⊙ 389
29.1 Problem 1: Particles from Field Equations 389
29.2 Theory: Circular Ring Solitons (Pulsons) 389
29.3 Problem 2: Dispersionless Dispersive Chains 390
29.4 Theory: Coupled Pendula 390
29.4.1 Dispersion in Linear Chain 392
29.4.2 Continuum Limit, the SGE 393
29.5 Solution: Analytic 394
29.6 Solution: Numeric 394
29.7 Implementation: 2-D Solitons, twodsol.f (.c) 397
29.8 Visualization 398

30 Confined Electronic Wave Packets ⊙ 399
30.1 Problem: A Confined Electron 399
30.2 Model: Time-Dependent Schrödinger Equation 399
30.3 Method, Numeric: Finite Difference 401
30.4 Implementation: Wave Packet in Well, sqwell.f 403
30.5 Assessment: Visualization, and Animation 403
30.6 Exploration: 1 -D Harmonic Oscillator 404
30.7 Implementation: harmos.f 404
30.8 Problem: Two-Dimensional Confinement 404
30.9 Method: Numerical 406
30.10 Exploration: 2-D Harmonic Oscillator 407
30.11 Exploration: Single-Slit Diffraction, slit.f 408

Appendix A: Analogous Elements in Fortran and C 409

Appendix B: Programs on Floppy Diskette 411

Appendix C: Listing of C Programs 417

Appendix D: Listing of Fortran Programs 461

Appendix E: Typical Project Assignments 501

Glossary 503

References 509