

*Rubin H. Landau, Manuel J. Páez,  
and Cristian C. Bordeianu*

# **Computational Physics**

## **Problem Solving with Computers**

2nd, Revised and Enlarged Edition



WILEY-VCH Verlag GmbH & Co. KGaA

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Computational Physics and Computational Science	1
1.2	How to Use this Book	3
<b>2</b>	<b>Computing Software Basics</b>	<b>7</b>
2.1	Making Computers Obey	7
2.2	Computer Languages	7
2.3	Programming Warmup	9
2.3.1	Java-Scanner Implementation	10
2.3.2	C Implementation	11
2.3.3	Fortran Implementation	12
2.4	Shells, Editors, and Programs	12
2.5	Limited Range and Precision of Numbers	13
2.6	Number Representation	13
2.7	IEEE Floating Point Numbers	14
2.8	Over/Underflows Exercise	20
2.9	Machine Precision	21
2.10	Determine Your Machine Precision	23
2.11	Structured Program Design	24
2.12	Summing Series	26
2.13	Numeric Summation	26
2.14	Good and Bad Pseudocode	27
2.15	Assessment	27
<b>3</b>	<b>Errors and Uncertainties in Computations</b>	<b>29</b>
3.1	Living with Errors	29
3.2	Types of Errors	29
3.3	Model for Disaster: Subtractive Cancellation	31
3.4	Subtractive Cancellation Exercises	32
3.5	Model for Roundoff Error Accumulation	34

3.6	Errors in Spherical Bessel Functions (Problem)	35
3.7	Numeric Recursion Relations (Method)	35
3.8	Implementation and Assessment: Recursion Relations	37
3.9	Experimental Error Determination	39
3.10	Errors in Algorithms	39
3.11	Minimizing the Error	41
3.12	Error Assessment	42
<b>4</b>	<b>Object-Oriented Programming: Kinematics</b> ☉	<b>45</b>
4.1	Problem: Superposition of Motions	45
4.2	Theory: Object-Oriented Programming	45
4.2.1	OOP Fundamentals	46
4.3	Theory: Newton's Laws, Equation of Motion	46
4.4	OOP Method: Class Structure	47
4.5	Implementation: Uniform 1D Motion, unim1d.cpp	48
4.5.1	Uniform Motion in 1D, Class Um1D	49
4.5.2	Implementation: Uniform Motion in 2D, Child Um2D, unimot2d.cpp	50
4.5.3	Class Um2D: Uniform Motion in 2D	51
4.5.4	Implementation: Projectile Motion, Child Accm2D, accm2d.cpp	53
4.5.5	Accelerated Motion in Two Directions	54
4.6	Assessment: Exploration, shms.cpp	56
<b>5</b>	<b>Integration</b>	<b>59</b>
5.1	Problem: Integrating a Spectrum	59
5.2	Quadrature as Box Counting (Math)	59
5.3	Algorithm: Trapezoid Rule	61
5.4	Algorithm: Simpson's Rule	63
5.5	Integration Error	65
5.6	Algorithm: Gaussian Quadrature	66
5.6.1	Mapping Integration Points	68
5.6.2	Gauss Implementation	69
5.7	Empirical Error Estimate (Assessment)	71
5.8	Experimentation	72
5.9	Higher Order Rules	72
<b>6</b>	<b>Differentiation</b>	<b>75</b>
6.1	Problem 1: Numerical Limits	75
6.2	Method: Numeric	75
6.3	Forward Difference	75
6.4	Central Difference	76
6.5	Extrapolated Difference	77

6.6	Error Analysis	78
6.7	Error Analysis (Implementation and Assessment)	79
6.8	Second Derivatives	80
6.8.1	Second Derivative Assessment	80
<b>7</b>	<b>Trial and Error Searching</b>	<b>81</b>
7.1	Quantum States in Square Well	81
7.2	Trial-and-Error Root Finding via Bisection Algorithm	83
7.2.1	Bisection Algorithm Implementation	84
7.3	Newton–Raphson Algorithm	84
7.3.1	Newton–Raphson with Backtracking	86
7.3.2	Newton–Raphson Implementation	87
<b>8</b>	<b>Matrix Computing and N-D Newton Raphson</b>	<b>89</b>
8.1	Two Masses on a String	90
8.1.1	Statics	91
8.1.2	Multidimensional Newton–Raphson Searching	92
8.2	Classes of Matrix Problems	95
8.2.1	Practical Aspects of Matrix Computing	96
8.2.2	Implementation: Scientific Libraries, WWW	100
8.2.3	Exercises for Testing Matrix Calls	106
8.2.4	Matrix Solution of Problem	108
8.2.5	Explorations	108
<b>9</b>	<b>Data Fitting</b>	<b>111</b>
9.1	Fitting Experimental Spectrum	111
9.1.1	Lagrange Interpolation	112
9.1.2	Lagrange Implementation and Assessment	114
9.1.3	Explore Extrapolation	116
9.1.4	Cubic Splines	116
9.1.5	Spline Fit of Cross Section	118
9.2	Fitting Exponential Decay	120
9.2.1	Theory to Fit	120
9.3	Theory: Probability and Statistics	121
9.4	Least-Squares Fitting	124
9.4.1	Goodness of Fit	126
9.4.2	Least-Squares Fits Implementation	126
9.4.3	Exponential Decay Fit Assessment	128
9.4.4	Exercise: Fitting Heat Flow	129
9.4.5	Nonlinear Fit of Breit–Wigner to Cross Section	130
9.5	Appendix: Calling LAPACK from C	132
9.5.1	Calling LAPACK Fortran from C	134
9.5.2	Compiling C Programs with Fortran Calls	134

<b>10</b>	<b>Deterministic Randomness</b>	<b>137</b>
10.1	Random Sequences	137
10.1.1	Random-Number Generation	138
10.1.2	Implementation: Random Sequence	140
10.1.3	Assessing Randomness and Uniformity	141
<b>11</b>	<b>Monte Carlo Applications</b>	<b>145</b>
11.1	A Random Walk	145
11.1.1	Simulation	145
11.1.2	Implementation: Random Walk	147
11.2	Radioactive Decay	148
11.2.1	Discrete Decay	148
11.2.2	Continuous Decay	150
11.2.3	Simulation	150
11.3	Implementation and Visualization	151
11.4	Integration by Stone Throwing	152
11.5	Integration by Rejection	153
11.5.1	Implementation	154
11.5.2	Integration by Mean Value	154
11.6	High-Dimensional Integration	155
11.6.1	Multidimensional Monte Carlo	156
11.6.2	Error in N-D Integration	156
11.6.3	Implementation: 10D Monte Carlo Integration	157
11.7	Integrating Rapidly Varying Functions $\odot$	157
11.7.1	Variance Reduction $\odot$ (Method)	157
11.7.2	Importance Sampling $\odot$	158
11.7.3	Implementation: Nonuniform Randomness $\odot$	158
11.7.4	von Neumann Rejection $\odot$	162
11.7.5	Nonuniform Assessment $\odot$	163
<b>12</b>	<b>Thermodynamic Simulations: Ising Model</b>	<b>165</b>
12.1	Statistical Mechanics	165
12.2	An Ising Chain (Model)	166
12.2.1	Analytic Solutions	169
12.3	The Metropolis Algorithm	169
12.3.1	Implementation	173
12.3.2	Equilibration	173
12.3.3	Thermodynamic Properties	175
12.3.4	Beyond Nearest Neighbors and 1D	177

<b>13</b>	<b>Computer Hardware Basics: Memory and CPU</b>	<b>179</b>
13.1	High-Performance Computers	179
13.1.1	Memory Hierarchy	180
13.2	The Central Processing Unit	184
13.2.1	CPU Design: RISC	185
13.2.2	Vector Processor	186
<b>14</b>	<b>High-Performance Computing: Profiling and Tuning</b>	<b>189</b>
14.1	Rules for Optimization	189
14.1.1	Programming for Virtual Memory	190
14.1.2	Optimizing Programs; Java vs. Fortran/C	190
14.1.3	Good, Bad Virtual Memory Use	192
14.1.4	Experimental Effects of Hardware on Performance	193
14.1.5	Java versus Fortran/C	195
14.2	Programming for Data Cache	203
14.2.1	Exercise 1: Cache Misses	204
14.2.2	Exercise 2: Cache Flow	204
14.2.3	Exercise 3: Large Matrix Multiplication	205
<b>15</b>	<b>Differential Equation Applications</b>	<b>207</b>
15.1	UNIT I. <i>Free Nonlinear Oscillations</i>	207
15.2	Nonlinear Oscillator	208
15.3	Math: Types of Differential Equations	209
15.4	Dynamical Form for ODEs	212
15.5	ODE Algorithms	213
15.5.1	Euler's Rule	215
15.5.2	Runge–Kutta Algorithm	215
15.5.3	Assessment: rk2 v. rk4 v. rk45	221
15.6	Solution for Nonlinear Oscillations	223
15.6.1	Precision Assessment: Energy Conservation	224
15.7	Extensions: Nonlinear Resonances, Beats and Friction	225
15.7.1	Friction: Model and Implementation	225
15.7.2	Resonances and Beats: Model and Implementation	226
15.8	Implementation: Inclusion of Time-Dependent Force	226
15.9	UNIT II. <i>Balls, not Planets, Fall Out of the Sky</i>	228
15.10	Theory: Projectile Motion with Drag	228
15.10.1	Simultaneous Second Order ODEs	229
15.10.2	Assessment	230
15.11	Exploration: Planetary Motion	231
15.11.1	Implementation: Planetary Motion	232

<b>16</b>	<b>Quantum Eigenvalues via ODE Matching</b>	<b>235</b>
16.1	Theory: The Quantum Eigenvalue Problem	236
16.1.1	Model: Nucleon in a Box	236
16.1.2	Algorithm: Eigenvalues via ODE Solver + Search	238
16.1.3	Implementation: ODE Eigenvalues Solver	242
16.1.4	Explorations	243
<b>17</b>	<b>Fourier Analysis of Linear and Nonlinear Signals</b>	<b>245</b>
17.1	Harmonics of Nonlinear Oscillations	245
17.2	Fourier Analysis	246
17.2.1	Example 1: Sawtooth Function	248
17.2.2	Example 2: Half-Wave Function	249
17.3	Summation of Fourier Series(Exercise)	250
17.4	Fourier Transforms	250
17.5	Discrete Fourier Transform Algorithm (DFT)	252
17.6	Aliasing and Antialiasing⊙	257
17.7	DFT for Fourier Series	259
17.8	Assessments	260
17.9	DFT of Nonperiodic Functions (Exploration)	261
17.10	Model Independent Data Analysis ⊙	262
17.11	Assessment	264
<b>18</b>	<b>Unusual Dynamics of Nonlinear Systems</b>	<b>267</b>
18.1	The Logistic Map	267
18.2	Properties of Nonlinear Maps	269
18.2.1	Fixed Points	269
18.2.2	Period Doubling, Attractors	270
18.3	Explicit Mapping Implementation	271
18.4	Bifurcation Diagram	272
18.4.1	Implementation	273
18.4.2	Visualization Algorithm: Binning	274
18.5	Random Numbers via Logistic Map	275
18.6	Feigenbaum Constants	276
18.7	Other Maps	276
<b>19</b>	<b>Differential Chaos in Phase Space</b>	<b>277</b>
19.1	Problem: A Pendulum Becomes Chaotic (Differential Chaos)	277
19.2	Equation of Chaotic Pendulum	278
19.2.1	Oscillations of a Free Pendulum	279
19.2.2	Pendulum's "Solution" as Elliptic Integrals	280
19.2.3	Implementation and Test: Free Pendulum	280
19.3	Visualization: Phase-Space Orbits	282
19.3.1	Chaos in Phase Space	285

- 19.3.2 Assessment in Phase Space 286
- 19.4 Assessment: Fourier Analysis of Chaos 288
- 19.5 Exploration: Bifurcations in Chaotic Pendulum 290
- 19.6 Exploration: Another Type of Phase-Space Plot 291
- 19.7 Further Explorations 291
  
- 20 Fractals 293**
  - 20.1 Fractional Dimension 293
  - 20.2 The Sierpiński Gasket 294
    - 20.2.1 Implementation 295
    - 20.2.2 Assessing Fractal Dimension 295
  - 20.3 Beautiful Plants 297
    - 20.3.1 Self-Affine Connection 297
    - 20.3.2 Barnsley's Fern (fern.c) 298
    - 20.3.3 Self-Affinity in Trees (tree.c) 300
  - 20.4 Ballistic Deposition 301
    - 20.4.1 Random Deposition Algorithm (film.c) 301
  - 20.5 Length of British Coastline 303
    - 20.5.1 Coastline as Fractal 303
    - 20.5.2 Box Counting Algorithm 304
    - 20.5.3 Coastline Implementation 305
  - 20.6 Problem 5: Correlated Growth, Forests, and Films 306
    - 20.6.1 Correlated Ballistic Deposition Algorithm (column.c) 307
    - 20.6.2 Globular Cluster 308
    - 20.6.3 Diffusion-Limited Aggregation Algorithm (dla.c) 308
    - 20.6.4 Fractal Analysis of DLA Graph 310
  - 20.7 Problem 7: Fractals in Bifurcation Graph 311
  
- 21 Parallel Computing 313**
  - 21.1 Parallel Semantics 314
    - 21.1.1 Granularity 315
  - 21.2 Distributed Memory Programming 316
  - 21.3 Parallel Performance 317
    - 21.3.1 Communication Overhead 319
  
- 22 Parallel Computing with MPI 321**
  - 22.1 Running on a Beowulf 322
    - 22.1.1 An Alternative: BCCD = Your Cluster on a CD 326
  - 22.2 Running MPI 326
    - 22.2.1 MPI under a Queuing System 327
    - 22.2.2 Your First MPI Program 329
    - 22.2.3 MPIhello.c Explained 330



22.2.4	Send/Receive Messages	332
22.2.5	Receive More Messages	333
22.2.6	Broadcast Messages: MPIpi.c	334
22.2.7	Exercise	336
22.3	Parallel Tuning: TuneMPI.c	340
22.4	A String Vibrating in Parallel	342
22.4.1	MPIstring.c Exercise	345
22.5	Deadlock	346
22.5.1	Nonblocking Communication	347
22.5.2	Collective Communication	347
22.6	Supplementary Exercises	348
22.7	List of MPI Commands	349
<b>23</b>	<b>Electrostatics Potentials via Finite Differences (PDEs)</b>	<b>351</b>
23.1	PDE Generalities	351
23.2	Electrostatic Potentials	353
23.2.1	Laplace's Elliptic PDE	353
23.3	Fourier Series Solution of PDE	354
23.3.1	Shortcomings of Polynomial Expansions	356
23.4	Solution: Finite Difference Method	357
23.4.1	Relaxation and Over-Relaxation	359
23.4.2	Lattice PDE Implementation	361
23.5	Assessment via Surface Plot	362
23.6	Three Alternate Capacitor Problems	363
23.7	Implementation and Assessment	365
23.8	Other Geometries and Boundary Conditions	368
<b>24</b>	<b>Heat Flow</b>	<b>369</b>
24.1	The Parabolic Heat Equation	369
24.2	Solution: Analytic Expansion	370
24.3	Solution: Finite Time Stepping (Leap Frog)	371
24.4	von Neumann Stability Assessment	373
24.4.1	Implementation	374
24.5	Assessment and Visualization	376
<b>25</b>	<b>PDE Waves on Strings and Membranes</b>	<b>379</b>
25.1	The Hyperbolic Wave Equation	379
25.1.1	Solution via Normal Mode Expansion	381
25.1.2	Algorithm: Time Stepping (Leapfrog)	382
25.1.3	Implementation	386
25.1.4	Assessment and Exploration	386
25.1.5	Including Friction (Extension)	388

- 25.1.6 Variable Tension and Density 390
- 25.2 Realistic 1D Wave Exercises 391
- 25.3 Vibrating Membrane (2D Waves) 392
- 25.4 Analytical Solution 394
- 25.5 Numerical Solution for 2D Waves 396
  
- 26 Solitons; KdeV and Sine-Gordon 399**
- 26.1 Chain of Coupled Pendulums (Theory) 399
- 26.2 Wave Dispersion 400
- 26.2.1 Continuum Limit, the SGE 402
- 26.3 Analytic SGE Solution 403
- 26.4 Numeric Solution: 2D SGE Solitons 403
- 26.5 2D Soliton Implementation 406
- 26.6 Visualization 408
- 26.7 Shallow Water (KdeV) Solitons  $\odot$  409
- 26.8 Theory: The Korteweg–de Vries Equation 410
- 26.8.1 Analytic Solution: KdeV Solitons 411
- 26.8.2 Algorithm: KdeV Soliton Solution 412
- 26.8.3 Implementation: KdeV Solitons 413
- 26.8.4 Exploration: Two KdeV Solitons Crossing 415
- 26.8.5 Phase-Space Behavior 415
  
- 27 Quantum Wave Packets  $\odot$  417**
- 27.1 Time-Dependent Schrödinger Equation (Theory) 417
- 27.1.1 Finite Difference Solution 419
- 27.1.2 Implementation 419
- 27.1.3 Visualization and Animation 422
- 27.2 Wave Packets Confined to Other Wells (Exploration) 422
- 27.2.1 Algorithm for 2D Schrödinger Equation 423
  
- 28 Quantum Paths for Functional Integration 427**
- 28.1 Feynman’s Space–Time Propagation 427
- 28.1.1 Bound-State Wave Function 431
- 28.1.2 Lattice Path Integration (Algorithm) 432
- 28.1.3 Implementation 439
- 28.1.4 Assessment and Exploration 441
  
- 29 Quantum Bound States via Integral Equations 443**
- 29.1 Momentum–Space Schrödinger Equation 444
- 29.1.1 Integral to Linear Equations 445
- 29.1.2 Delta-Shell Potential (Model) 447
- 29.1.3 Implementation 448

29.1.4	Wave Function	449
<b>30</b>	<b>Quantum Scattering via Integral Equations</b>	<b>451</b>
30.1	Lippmann–Schwinger Equation	451
30.1.1	Singular Integrals	452
30.1.2	Numerical Principal Values	453
30.1.3	Reducing Integral to Matrix Equations	454
30.1.4	Solution via Inversion, Elimination	455
30.1.5	Solving $i\epsilon$ Integral Equations $\odot$	456
30.1.6	Delta-Shell Potential Implementation	456
30.1.7	Scattering Wave Function	458
<b>A</b>	<b>PtPlot: 2D Graphs within Java</b>	<b>461</b>
<b>B</b>	<b>Glossary</b>	<b>467</b>
<b>C</b>	<b>Fortran 95 Codes</b>	<b>479</b>
<b>D</b>	<b>Fortran 77 Codes</b>	<b>513</b>
<b>E</b>	<b>C Language Codes</b>	<b>547</b>
	<b>References</b>	<b>583</b>
	<b>Index</b>	<b>587</b>