Contents

Dedication *V*

Preface *XIX*

1 **Introduction** *1*
1.1 Computational Physics and Computational Science *1*
1.2 This Book's Subjects *3*
1.3 This Book's Problems *4*
1.4 This Book's Language: The Python Ecosystem *8*
1.4.1 Python Packages (Libraries) *9*
1.4.2 This Book's Packages *10*
1.4.3 The Easy Way: Python Distributions (Package Collections) *12*
1.5 Python's Visualization Tools *13*
1.5.1 Visual (VPython)'s 2D Plots *14*
1.5.2 VPython's Animations *17*
1.5.3 Matplotlib's 2D Plots *17*
1.5.4 Matplotlib's 3D Surface Plots *22*
1.5.5 Matplotlib's Animations *24*
1.5.6 Mayavi's Visualizations Beyond Plotting *26*
1.6 Plotting Exercises *30*
1.7 Python's Algebraic Tools *31*

2 **Computing Software Basics** *33*
2.1 Making Computers Obey *33*
2.2 Programming Warmup *35*
2.2.1 Structured and Reproducible Program Design *36*
2.2.2 Shells, Editors, and Execution *37*
2.3 Python I/O *39*
2.4 Computer Number Representations (Theory) *40*
2.4.1 IEEE Floating-Point Numbers *41*
2.4.2 Python and the IEEE 754 Standard *47*
2.4.3 Over and Underflow Exercises *48*
2.4.4 Machine Precision (Model) *49*

Bibliografische Informationen
http://d-nb.info/1065330790

digitalisiert durch DEUTSCHE NATIONAL BIBLIOTHEK

2.4.5 Experiment: Your Machine's Precision *50*
2.5 Problem: Summing Series *51*
2.5.1 Numerical Summation (Method) *51*
2.5.2 Implementation and Assessment *52*

3 Errors and Uncertainties in Computations *53*
3.1 Types of Errors (Theory) *53*
3.1.1 Model for Disaster: Subtractive Cancelation *55*
3.1.2 Subtractive Cancelation Exercises *56*
3.1.3 Round-off Errors *57*
3.1.4 Round-off Error Accumulation *58*
3.2 Error in Bessel Functions (Problem) *58*
3.2.1 Numerical Recursion (Method) *59*
3.2.2 Implementation and Assessment: Recursion Relations *61*
3.3 Experimental Error Investigation *62*
3.3.1 Error Assessment *65*

4 Monte Carlo: Randomness, Walks, and Decays *69*
4.1 Deterministic Randomness *69*
4.2 Random Sequences (Theory) *69*
4.2.1 Random-Number Generation (Algorithm) *70*
4.2.2 Implementation: Random Sequences *72*
4.2.3 Assessing Randomness and Uniformity *73*
4.3 Random Walks (Problem) *75*
4.3.1 Random-Walk Simulation *76*
4.3.2 Implementation: Random Walk *77*
4.4 Extension: Protein Folding and Self-Avoiding Random Walks *79*
4.5 Spontaneous Decay (Problem) *80*
4.5.1 Discrete Decay (Model) *81*
4.5.2 Continuous Decay (Model) *82*
4.5.3 Decay Simulation with Geiger Counter Sound *82*
4.6 Decay Implementation and Visualization *84*

5 Differentiation and Integration *85*
5.1 Differentiation *85*
5.2 Forward Difference (Algorithm) *86*
5.3 Central Difference (Algorithm) *87*
5.4 Extrapolated Difference (Algorithm) *87*
5.5 Error Assessment *88*
5.6 Second Derivatives (Problem) *90*
5.6.1 Second-Derivative Assessment *90*
5.7 Integration *91*
5.8 Quadrature as Box Counting (Math) *91*
5.9 Algorithm: Trapezoid Rule *93*
5.10 Algorithm: Simpson's Rule *94*

5.11 Integration Error (Assessment) *96*
5.12 Algorithm: Gaussian Quadrature *97*
5.12.1 Mapping Integration Points *98*
5.12.2 Gaussian Points Derivation *99*
5.12.3 Integration Error Assessment *100*
5.13 Higher Order Rules (Algorithm) *103*
5.14 Monte Carlo Integration by Stone Throwing (Problem) *104*
5.14.1 Stone Throwing Implementation *104*
5.15 Mean Value Integration (Theory and Math) *105*
5.16 Integration Exercises *106*
5.17 Multidimensional Monte Carlo Integration (Problem) *108*
5.17.1 Multi Dimension Integration Error Assessment *109*
5.17.2 Implementation: 10D Monte Carlo Integration *110*
5.18 Integrating Rapidly Varying Functions (Problem) *110*
5.19 Variance Reduction (Method) *110*
5.20 Importance Sampling (Method) *111*
5.21 von Neumann Rejection (Method) *111*
5.21.1 Simple Random Gaussian Distribution *113*
5.22 Nonuniform Assessment ⊙ *113*
5.22.1 Implementation ⊙ *114*

6 Matrix Computing *117*
6.1 Problem 3: N–D Newton–Raphson; Two Masses on a String *117*
6.1.1 Theory: Statics *118*
6.1.2 Algorithm: Multidimensional Searching *119*
6.2 Why Matrix Computing? *122*
6.3 Classes of Matrix Problems (Math) *122*
6.3.1 Practical Matrix Computing *124*
6.4 Python Lists as Arrays *126*
6.5 Numerical Python (NumPy) Arrays *127*
6.5.1 NumPy's linalg Package *132*
6.6 Exercise: Testing Matrix Programs *134*
6.6.1 Matrix Solution of the String Problem *137*
6.6.2 Explorations *139*

7 Trial-and-Error Searching and Data Fitting *141*
7.1 Problem 1: A Search for Quantum States in a Box *141*
7.2 Algorithm: Trial-and-Error Roots via Bisection *142*
7.2.1 Implementation: Bisection Algorithm *144*
7.3 Improved Algorithm: Newton–Raphson Searching *145*
7.3.1 Newton–Raphson with Backtracking *147*
7.3.2 Implementation: Newton–Raphson Algorithm *148*
7.4 Problem 2: Temperature Dependence of Magnetization *148*
7.4.1 Searching Exercise *150*
7.5 Problem 3: Fitting An Experimental Spectrum *150*

7.5.1 Lagrange Implementation, Assessment *152*
7.5.2 Cubic Spline Interpolation (Method) *153*
7.6 Problem 4: Fitting Exponential Decay *156*
7.7 Least-Squares Fitting (Theory) *158*
7.7.1 Least-Squares Fitting: Theory and Implementation *160*
7.8 Exercises: Fitting Exponential Decay, Heat Flow and Hubble's Law *162*
7.8.1 Linear Quadratic Fit *164*
7.8.2 Problem 5: Nonlinear Fit to a Breit–Wigner *167*

8 Solving Differential Equations: Nonlinear Oscillations *171*
8.1 Free Nonlinear Oscillations *171*
8.2 Nonlinear Oscillators (Models) *171*
8.3 Types of Differential Equations (Math) *173*
8.4 Dynamic Form for ODEs (Theory) *175*
8.5 ODE Algorithms *177*
8.5.1 Euler's Rule *177*
8.6 Runge–Kutta Rule *178*
8.7 Adams–Bashforth–Moulton Predictor–Corrector Rule *183*
8.7.1 Assessment: rk2 vs. rk4 vs. rk45 *185*
8.8 Solution for Nonlinear Oscillations (Assessment) *187*
8.8.1 Precision Assessment: Energy Conservation *188*
8.9 Extensions: Nonlinear Resonances, Beats, Friction *189*
8.9.1 Friction (Model) *189*
8.9.2 Resonances and Beats: Model, Implementation *190*
8.10 Extension: Time-Dependent Forces *190*

9 ODE Applications: Eigenvalues, Scattering, and Projectiles *193*
9.1 Problem: Quantum Eigenvalues in Arbitrary Potential *193*
9.1.1 Model: Nucleon in a Box *194*
9.2 Algorithms: Eigenvalues via ODE Solver + Search *195*
9.2.1 Numerov Algorithm for Schrödinger ODE ⊙ *197*
9.2.2 Implementation: Eigenvalues via ODE Solver + Bisection Algorithm *200*
9.3 Explorations *203*
9.4 Problem: Classical Chaotic Scattering *203*
9.4.1 Model and Theory *204*
9.4.2 Implementation *206*
9.4.3 Assessment *207*
9.5 Problem: Balls Falling Out of the Sky *208*
9.6 Theory: Projectile Motion with Drag *208*
9.6.1 Simultaneous Second-Order ODEs *209*
9.6.2 Assessment *210*
9.7 Exercises: 2- and 3-Body Planet Orbits and Chaotic Weather *211*

10 High-Performance Hardware and Parallel Computers *215*
10.1 High-Performance Computers *215*

10.2 Memory Hierarchy *216*
10.3 The Central Processing Unit *219*
10.4 CPU Design: Reduced Instruction Set Processors *220*
10.5 CPU Design: Multiple-Core Processors *221*
10.6 CPU Design: Vector Processors *222*
10.7 Introduction to Parallel Computing *223*
10.8 Parallel Semantics (Theory) *224*
10.9 Distributed Memory Programming *226*
10.10 Parallel Performance *227*
10.10.1 Communication Overhead *229*
10.11 Parallelization Strategies *230*
10.12 Practical Aspects of MIMD Message Passing *231*
10.12.1 High-Level View of Message Passing *233*
10.12.2 Message Passing Example and Exercise *234*
10.13 Scalability *236*
10.13.1 Scalability Exercises *238*
10.14 Data Parallelism and Domain Decomposition *239*
10.14.1 Domain Decomposition Exercises *242*
10.15 Example: The IBM Blue Gene Supercomputers *243*
10.16 Exascale Computing via Multinode-Multicore GPUs *245*

11 Applied HPC: Optimization, Tuning, and GPU Programming *247*
11.1 General Program Optimization *247*
11.1.1 Programming for Virtual Memory (Method) *248*
11.1.2 Optimization Exercises *249*
11.2 Optimized Matrix Programming with NumPy *251*
11.2.1 NumPy Optimization Exercises *254*
11.3 Empirical Performance of Hardware *254*
11.3.1 Racing Python vs. Fortran/C *255*
11.4 Programming for the Data Cache (Method) *262*
11.4.1 Exercise 1: Cache Misses *264*
11.4.2 Exercise 2: Cache Flow *264*
11.4.3 Exercise 3: Large-Matrix Multiplication *265*
11.5 Graphical Processing Units for High Performance Computing *266*
11.5.1 The GPU Card *267*
11.6 Practical Tips for Multicore and GPU Programming ⊙ *267*
11.6.1 CUDA Memory Usage *270*
11.6.2 CUDA Programming ⊙ *271*

12 Fourier Analysis: Signals and Filters *275*
12.1 Fourier Analysis of Nonlinear Oscillations *275*
12.2 Fourier Series (Math) *276*
12.2.1 Examples: Sawtooth and Half-Wave Functions *278*
12.3 Exercise: Summation of Fourier Series *279*
12.4 Fourier Transforms (Theory) *279*

12.5 The Discrete Fourier Transform *281*
12.5.1 Aliasing (Assessment) *285*
12.5.2 Fourier Series DFT (Example) *287*
12.5.3 Assessments *288*
12.5.4 Nonperiodic Function DFT (Exploration) *290*
12.6 Filtering Noisy Signals *290*
12.7 Noise Reduction via Autocorrelation (Theory) *290*
12.7.1 Autocorrelation Function Exercises *293*
12.8 Filtering with Transforms (Theory) *294*
12.8.1 Digital Filters: Windowed Sinc Filters (Exploration) ⊙ *296*
12.9 The Fast Fourier Transform Algorithm ⊙ *299*
12.9.1 Bit Reversal *301*
12.10 FFT Implementation *303*
12.11 FFT Assessment *304*

13 Wavelet and Principal Components Analyses: Nonstationary Signals and Data Compression *307*
13.1 Problem: Spectral Analysis of Nonstationary Signals *307*
13.2 Wavelet Basics *307*
13.3 Wave Packets and Uncertainty Principle (Theory) *309*
13.3.1 Wave Packet Assessment *311*
13.4 Short-Time Fourier Transforms (Math) *311*
13.5 The Wavelet Transform *313*
13.5.1 Generating Wavelet Basis Functions *313*
13.5.2 Continuous Wavelet Transform Implementation *316*
13.6 Discrete Wavelet Transforms, Multiresolution Analysis ⊙ *317*
13.6.1 Pyramid Scheme Implementation ⊙ *323*
13.6.2 Daubechies Wavelets via Filtering *327*
13.6.3 DWT Implementation and Exercise *330*
13.7 Principal Components Analysis *332*
13.7.1 Demonstration of Principal Component Analysis *334*
13.7.2 PCA Exercises *337*

14 Nonlinear Population Dynamics *339*
14.1 Bug Population Dynamics *339*
14.2 The Logistic Map (Model) *339*
14.3 Properties of Nonlinear Maps (Theory and Exercise) *341*
14.3.1 Fixed Points *342*
14.3.2 Period Doubling, Attractors *343*
14.4 Mapping Implementation *344*
14.5 Bifurcation Diagram (Assessment) *345*
14.5.1 Bifurcation Diagram Implementation *346*
14.5.2 Visualization Algorithm: Binning *347*
14.5.3 Feigenbaum Constants (Exploration) *348*
14.6 Logistic Map Random Numbers (Exploration) ⊙ *348*

14.7 Other Maps (Exploration) *348*
14.8 Signals of Chaos: Lyapunov Coefficient and Shannon Entropy ⊙ *349*
14.9 Coupled Predator–Prey Models *353*
14.10 Lotka–Volterra Model *354*
14.10.1 Lotka–Volterra Assessment *356*
14.11 Predator–Prey Chaos *356*
14.11.1 Exercises *359*
14.11.2 LVM with Prey Limit *359*
14.11.3 LVM with Predation Efficiency *360*
14.11.4 LVM Implementation and Assessment *361*
14.11.5 Two Predators, One Prey (Exploration) *362*

15 Continuous Nonlinear Dynamics *363*
15.1 Chaotic Pendulum *363*
15.1.1 Free Pendulum Oscillations *364*
15.1.2 Solution as Elliptic Integrals *365*
15.1.3 Implementation and Test: Free Pendulum *366*
15.2 Visualization: Phase-Space Orbits *367*
15.2.1 Chaos in Phase Space *368*
15.2.2 Assessment in Phase Space *372*
15.3 Exploration: Bifurcations of Chaotic Pendulums *374*
15.4 Alternate Problem: The Double Pendulum *375*
15.5 Assessment: Fourier/Wavelet Analysis of Chaos *377*
15.6 Exploration: Alternate Phase-Space Plots *378*
15.7 Further Explorations *379*

16 Fractals and Statistical Growth Models *383*
16.1 Fractional Dimension (Math) *383*
16.2 The Sierpiński Gasket (Problem 1) *384*
16.2.1 Sierpiński Implementation *384*
16.2.2 Assessing Fractal Dimension *385*
16.3 Growing Plants (Problem 2) *386*
16.3.1 Self-Affine Connection (Theory) *386*
16.3.2 Barnsley's Fern Implementation *387*
16.3.3 Self-Affinity in Trees Implementation *389*
16.4 Ballistic Deposition (Problem 3) *390*
16.4.1 Random Deposition Algorithm *390*
16.5 Length of British Coastline (Problem 4) *391*
16.5.1 Coastlines as Fractals (Model) *392*
16.5.2 Box Counting Algorithm *392*
16.5.3 Coastline Implementation and Exercise *393*
16.6 Correlated Growth, Forests, Films (Problem 5) *395*
16.6.1 Correlated Ballistic Deposition Algorithm *395*
16.7 Globular Cluster (Problem 6) *396*
16.7.1 Diffusion-Limited Aggregation Algorithm *396*

16.7.2 Fractal Analysis of DLA or a Pollock *399*
16.8 Fractals in Bifurcation Plot (Problem 7) *400*
16.9 Fractals from Cellular Automata *400*
16.10 Perlin Noise Adds Realism ⊙ *402*
16.10.1 Ray Tracing Algorithms *404*
16.11 Exercises *407*

17 Thermodynamic Simulations and Feynman Path Integrals *409*
17.1 Magnets via Metropolis Algorithm *409*
17.2 An Ising Chain (Model) *410*
17.3 Statistical Mechanics (Theory) *412*
17.3.1 Analytic Solution *413*
17.4 Metropolis Algorithm *413*
17.4.1 Metropolis Algorithm Implementation *416*
17.4.2 Equilibration, Thermodynamic Properties (Assessment) *417*
17.4.3 Beyond Nearest Neighbors, 1D (Exploration) *419*
17.5 Magnets via Wang–Landau Sampling ⊙ *420*
17.6 Wang–Landau Algorithm *423*
17.6.1 WLS Ising Model Implementation *425*
17.6.2 WLS Ising Model Assessment *428*
17.7 Feynman Path Integral Quantum Mechanics ⊙ *429*
17.8 Feynman's Space–Time Propagation (Theory) *429*
17.8.1 Bound-State Wave Function (Theory) *431*
17.8.2 Lattice Path Integration (Algorithm) *432*
17.8.3 Lattice Implementation *437*
17.8.4 Assessment and Exploration *440*
17.9 Exploration: Quantum Bouncer's Paths ⊙ *440*

18 Molecular Dynamics Simulations *445*
18.1 Molecular Dynamics (Theory) *445*
18.1.1 Connection to Thermodynamic Variables *449*
18.1.2 Setting Initial Velocities *449*
18.1.3 Periodic Boundary Conditions and Potential Cutoff *450*
18.2 Verlet and Velocity–Verlet Algorithms *451*
18.3 1D Implementation and Exercise *453*
18.4 Analysis *456*

19 PDE Review and Electrostatics via Finite Differences and Electrostatics via Finite Differences *461*
19.1 PDE Generalities *461*
19.2 Electrostatic Potentials *463*
19.2.1 Laplace's Elliptic PDE (Theory) *463*
19.3 Fourier Series Solution of a PDE *464*
19.3.1 Polynomial Expansion as an Algorithm *466*
19.4 Finite-Difference Algorithm *467*

19.4.1 Relaxation and Over-relaxation *469*
19.4.2 Lattice PDE Implementation *470*
19.5 Assessment via Surface Plot *471*
19.6 Alternate Capacitor Problems *471*
19.7 Implementation and Assessment *474*
19.8 Electric Field Visualization (Exploration) *475*
19.9 Review Exercise *476*

20 Heat Flow via Time Stepping *477*
20.1 Heat Flow via Time-Stepping (Leapfrog) *477*
20.2 The Parabolic Heat Equation (Theory) *478*
20.2.1 Solution: Analytic Expansion *478*
20.2.2 Solution: Time Stepping *479*
20.2.3 von Neumann Stability Assessment *481*
20.2.4 Heat Equation Implementation *483*
20.3 Assessment and Visualization *483*
20.4 Improved Heat Flow: Crank–Nicolson Method *484*
20.4.1 Solution of Tridiagonal Matrix Equations ⊙ *487*
20.4.2 Crank–Nicolson Implementation, Assessment *490*

21 Wave Equations I: Strings and Membranes *491*
21.1 A Vibrating String *491*
21.2 The Hyperbolic Wave Equation (Theory) *491*
21.2.1 Solution via Normal-Mode Expansion *493*
21.2.2 Algorithm: Time Stepping *494*
21.2.3 Wave Equation Implementation *496*
21.2.4 Assessment, Exploration *497*
21.3 Strings with Friction (Extension) *499*
21.4 Strings with Variable Tension and Density *500*
21.4.1 Waves on Catenary *501*
21.4.2 Derivation of Catenary Shape *501*
21.4.3 Catenary and Frictional Wave Exercises *503*
21.5 Vibrating Membrane (2D Waves) *504*
21.6 Analytical Solution *505*
21.7 Numerical Solution for 2D Waves *508*

22 Wave Equations II: Quantum Packets and Electromagnetic *511*
22.1 Quantum Wave Packets *511*
22.2 Time-Dependent Schrödinger Equation (Theory) *511*
22.2.1 Finite-Difference Algorithm *513*
22.2.2 Wave Packet Implementation, Animation *514*
22.2.3 Wave Packets in Other Wells (Exploration) *516*
22.3 Algorithm for the 2D Schrödinger Equation *517*
22.3.1 Exploration: Bound and Diffracted 2D Packet *518*
22.4 Wave Packet–Wave Packet Scattering *518*

22.4.1 Algorithm *520*
22.4.2 Implementation *520*
22.4.3 Results and Visualization *522*
22.5 E&M Waves via Finite-Difference Time Domain *525*
22.6 Maxwell's Equations *525*
22.7 FDTD Algorithm *526*
22.7.1 Implementation *530*
22.7.2 Assessment *530*
22.7.3 Extension: Circularly Polarized Waves *531*
22.8 Application: Wave Plates *533*
22.9 Algorithm *534*
22.10 FDTD Exercise and Assessment *535*

23 Electrostatics via Finite Elements *537*
23.1 Finite-Element Method ⊙ *537*
23.2 Electric Field from Charge Density (Problem) *538*
23.3 Analytic Solution *538*
23.4 Finite-Element (Not Difference) Methods, 1D *539*
23.4.1 Weak Form of PDE *539*
23.4.2 Galerkin Spectral Decomposition *540*
23.5 1D FEM Implementation and Exercises *544*
23.5.1 1D Exploration *547*
23.6 Extension to 2D Finite Elements *547*
23.6.1 Weak Form of PDE *548*
23.6.2 Galerkin's Spectral Decomposition *548*
23.6.3 Triangular Elements *549*
23.6.4 Solution as Linear Equations *551*
23.6.5 Imposing Boundary Conditions *552*
23.6.6 FEM 2D Implementation and Exercise *554*
23.6.7 FEM 2D Exercises *554*

24 Shocks Waves and Solitons *555*
24.1 Shocks and Solitons in Shallow Water *555*
24.2 Theory: Continuity and Advection Equations *556*
24.2.1 Advection Implementation *558*
24.3 Theory: Shock Waves via Burgers' Equation *559*
24.3.1 Lax–Wendroff Algorithm for Burgers' Equation *560*
24.3.2 Implementation and Assessment of Burgers' Shock Equation *561*
24.4 Including Dispersion *562*
24.5 Shallow-Water Solitons: The KdeV Equation *563*
24.5.1 Analytic Soliton Solution *563*
24.5.2 Algorithm for KdeV Solitons *564*
24.5.3 Implementation: KdeV Solitons *565*
24.5.4 Exploration: Solitons in Phase Space, Crossing *567*
24.6 Solitons on Pendulum Chain *567*

24.6.1 Including Dispersion *568*
24.6.2 Continuum Limit, the Sine-Gordon Equation *570*
24.6.3 Analytic SGE Solution *571*
24.6.4 Numeric Solution: 2D SGE Solitons *571*
24.6.5 2D Soliton Implementation *573*
24.6.6 SGE Soliton Visualization *574*

25 Fluid Dynamics *575*
25.1 River Hydrodynamics *575*
25.2 Navier–Stokes Equation (Theory) *576*
25.2.1 Boundary Conditions for Parallel Plates *578*
25.2.2 Finite-Difference Algorithm and Overrelaxation *580*
25.2.3 Successive Overrelaxation Implementation *581*
25.3 2D Flow over a Beam *581*
25.4 Theory: Vorticity Form of Navier–Stokes Equation *582*
25.4.1 Finite Differences and the SOR Algorithm *584*
25.4.2 Boundary Conditions for a Beam *585*
25.4.3 SOR on a Grid *587*
25.4.4 Flow Assessment *589*
25.4.5 Exploration *590*

26 Integral Equations of Quantum Mechanics *591*
26.1 Bound States of Nonlocal Potentials *591*
26.2 Momentum–Space Schrödinger Equation (Theory) *592*
26.2.1 Integral to Matrix Equations *593*
26.2.2 Delta-Shell Potential (Model) *595*
26.2.3 Binding Energies Solution *595*
26.2.4 Wave Function (Exploration) *597*
26.3 Scattering States of Nonlocal Potentials ⊙ *597*
26.4 Lippmann–Schwinger Equation (Theory) *598*
26.4.1 Singular Integrals (Math) *599*
26.4.2 Numerical Principal Values *600*
26.4.3 Reducing Integral Equations to Matrix Equations (Method) *600*
26.4.4 Solution via Inversion, Elimination *602*
26.4.5 Scattering Implementation *603*
26.4.6 Scattering Wave Function (Exploration) *604*

Appendix A Codes, Applets, and Animations *607*

Bibliography *609*

Index *615*