THE NONLINEAR WORKBOOK

This page intentionally left blank

THE NONLINEAR WORKBOOK

4th Edition

Chaos, Fractals,
Cellular Automata,
Neural Networks,
Genetic Algorithms,
Gene Expression Programming,
Support Vector Machine,
Wavelets,
Hidden Markov Models,
Fuzzy Logic
with C++, Java and SymbolicC++ Programs

Willi-Hans Steeb
International School for Scientific Computing
University of Johannesburg, South Africa

in collaboration with

Yorick Hardy
International School for Scientific Computing, South Africa

Ruedi Stoop
Institute for Neuroinformatics, University of Zürich/ETHZ

World Scientific

NEW JERSEY • LONDON • SINGAPORE • BEIJING • SHANGHAI • HONG KONG • TAIPEI • CHENNAI

Published by

World Scientific Publishing Co. Pte. Ltd.
5 Toh Tuck Link, Singapore 596224
USA office: 27 Warren Street, Suite 401-402, Hackensack, NJ 07601
UK office: 57 Shelton Street, Covent Garden, London WC2H 9HE

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library.

THE NONLINEAR WORKBOOK — Fourth Edition
Chaos, Fractals, Cellular Automata, Neural Networks, Genetic Algorithms, Gene Expression Programming, Support Vector Machine, Wavelets, Hidden Markov Models, Fuzzy Logic with C++, Java and SymbolicC++ Programs

ISBN-13 978-981-281-852-2
ISBN-10 981-281-852-9
ISBN-13 978-981-281-853-9 (pbk)
ISBN-10 981-281-853-7 (pbk)

Printed in Singapore.

Preface

The study of chaos, fractals, cellular automata, neural networks, genetic algorithms and fuzzy logic is one of the most fascinating subjects in science. Most of these fields are interrelated. Chaotic attractors are used in neural networks. Genetic algorithms can be used to train neural networks. Fractals are used in data compression. Neural networks and fuzzy logic are often combined when the input values of the system are not crisp.

In this book we give all the basic concepts in these fields together with the definitions, theorems and algorithms. The algorithms are implemented using C++, Java and SymbolicC++. The level of presentation is such that one can study the subject early on in science. There is a balance between practical computation and the underlying mathematical theory.

In chapter 1 we consider one and two-dimensional nonlinear maps. All the relevant quantities to characterize chaotic systems are introduced. Algorithms are given for all the quantities which are used to describe chaos such as invariant density, Liapunov exponent, correlation integral, autocorrelation function, capacity, phase portrait, Poincaré section, Fourier transform, calculations of exact trajectories, fixed points and their stability, etc.. Chaotic repellers and encoding using one-dimensional chaotic maps are also investigated. Newton's method in one and two dimensions is derived. Periodic orbits and topological degree are introduced.

Quite often a dynamical system cannot be modelled by difference equations or differential equations, but an experiment provides a time series. In chapter 2 we consider quantities for the study of chaotic time-series. We also include the Hurst exponent which plays an important role in the study of financial markets. The related Higuchi algorithm is also provided.

In chapter 3 we describe the classification of fixed points in the plane. Furthermore the most important two-dimensional dynamical systems are studied, such as the pendulum, limit cycle systems and a Lotka-Volterra model. Homoclinic orbits are also introduced.

Chapter 4 reviews integrable and chaotic Hamilton systems. Among other concepts we introduce the Lax representation for integrable Hamilton systems, the Poincaré section and the Floquet theory.

In chapter 5 nonlinear dissipative systems are studied. The most famous dissipative system with chaotic behaviour, the Lorenz model, is introduced. We also discuss Hopf bifurcation and hyperchaotic systems.

Nonlinear driven systems play a central role in engineering, in particular in electronics. In most cases the driving force is periodic. Chapter 6 is devoted to these systems. As examples we consider among others the driven pendulum and the driven van der Pol equation. The concept of torsion number is also discussed.

Controlling of chaotic systems is very important in applications in engineering. In chapter 7 we discuss the different concepts of controlling chaos. The Ott-Grebogi-Yorke method for controlling chaotic systems is studied in detail.

Synchronization of chaotic systems is described in chapter 8 and a number of applications are given, such as the coupled Rikitake dynamos.

Fractals have become of increasing interest, not only in art, but also in many different areas of science such as compression algorithms. In chapter 9 we introduce iterated function systems, the Mandelbrot set, the Julia set and the Weierstrass function. The famous Cantor set is considered as an example as well as the Koch curve, fern and the Sierpinski gasket. We also derive the construction of fractals using the Kronecker product of matrices. Grey level maps are also described.

Cellular automata are discrete dynamical systems. We describe in chapter 10 one and two-dimensional cellular automata. The famous game of life with a C++ implementation and the button game with a Java implementation are also considered. The Sznajd model is studied as an application.

Chapter 11 is about integration of differential equations. We describe the Euler method, the Runge-Kutta method, the Lie series technique, symplectic integration, Verlet method, etc.. Furthermore we discuss ghost solutions, invisible chaos and integration in the complex domain.

Chapter 12 is devoted to neural networks. We introduce the Hopfield algorithm, the Kohonen self-organizing map, the back propagation algorithm and radial basis function networks. One of the applications is the traveling salesman problem. Neural oscillator models are also introduced.

Genetic algorithms are used to solve optimization problems. Chapter 13 is devoted to this technique. We discuss optimization problems with and without constraints. A detailed discussion of bitwise operations is given. We also study simulated annealing.

Gene expression programming is a new genetic algorithm that uses encoded individuals. Gene expression programming individuals are encoded in linear chromosomes which are expressed or translated into expression trees. The linear chromosome is the genetic material that is passed on with modifications to the next generation.

Chapter 14 gives an introduction to this technique together with a C++ program. As an alternative to gene expression programming we also describe multi-expression programming together with a C++ program.

In chapter 15 we consider the Lagrange multiplier method for optimization problems and also describe an alternative method using differential forms. For problems with inequality constraints the Karush-Kuhn-Tucker condition is provided and the support vector machine is studied. The Kernel-Adatron algorithm, a fast and simple learning procedure for support vector machines, is also implemented. As an application the kernel Fisher discriminant is studied.

Wavelet theory is a form of mathematical transformation, similar to the Fourier transform in that it takes a signal in time domain, and represents it in frequency domain. Wavelet functions are distinguished from other transformations in that they not only dissect signals into their component frequencies, they also vary the scale at which the component frequencies are analyzed. Chapter 16 provides an introduction. Filtering is given as an example application. As examples the Haar wavelet and Daubechies wavelet are studied. Two-dimensional wavelets are also considered.

Discrete Hidden Markov Models are introduced in chapter 17. The forward-backward algorithm, Viterbi algorithm, and Baum-Welch algorithm are described. The application concentrates on speech recognition.

Since its inception 40 years ago the theory of fuzzy sets has advanced in a variety of ways and in many disciplines, not only in science. Chapter 18 is devoted to fuzzy logic. Fuzzy numbers and arithmetic are also considered. Furthermore decision making problems and controlling problems using fuzzy logic are also described. Fuzzy clustering is also included as well as a definition for the fuzzy XOR.

In each chapter we give C++, Java and SymbolicC++ implementations of the algorithms.

Without doubt, this book can be extended. If you have comments or suggestions, I would be pleased to have them. The author can be contacted via e-mail:

`steeb_wh@yahoo.com`
`steebwilli@gmail.com`

The web page of the author is:

`http://issc.uj.ac.za`

The International School for Scientific Computing (ISSC) provides certificate courses for these subjects. Please contact the author if you want to do any of these courses.

This page intentionally left blank

Contents

1 Nonlinear and Chaotic Maps **1**
1.1 One-Dimensional Maps . . . 1
1.1.1 Exact and Numerical Trajectories . . . 3
1.1.2 Fixed Points and Stability . . . 14
1.1.3 Invariant Density . . . 16
1.1.4 Liapunov Exponent . . . 20
1.1.5 Autocorrelation Function . . . 23
1.1.6 Discrete Fourier Transform . . . 25
1.1.7 Fast Fourier Transform . . . 28
1.1.8 Logistic Map and Liapunov Exponent for $r \in [3,4]$ . . . 33
1.1.9 Logistic Map and Bifurcation Diagram . . . 34
1.1.10 Random Number Map and Invariant Density . . . 36
1.1.11 Random Number Map and Random Integration . . . 38
1.1.12 Circle Map and Rotation Number . . . 40
1.1.13 Newton Method . . . 41
1.1.14 Feigenbaum's Constant . . . 43
1.1.15 Symbolic Dynamics . . . 45
1.1.16 Chaotic Repeller . . . 47
1.1.17 Chaos and Encoding . . . 48
1.2 Two-Dimensional Maps . . . 54
1.2.1 Introduction . . . 54
1.2.2 Phase Portrait . . . 57
1.2.3 Fixed Points and Stability . . . 64
1.2.4 Liapunov Exponents . . . 65
1.2.5 Correlation Integral . . . 67
1.2.6 Capacity . . . 68
1.2.7 Hyperchaos . . . 70
1.2.8 Domain of Attraction . . . 74
1.2.9 Newton Method in the Complex Domain . . . 75
1.2.10 Newton Method in Higher Dimensions . . . 77
1.2.11 Ruelle-Takens-Newhouse Scenario . . . 78
1.2.12 Periodic Orbits and Topological Degree . . . 80
1.2.13 JPEG file . . . 82

2 Time Series Analysis 85
2.1 Introduction . . . 85
2.2 Correlation Coefficient . . . 86
2.3 Liapunov Exponent from Time Series . . . 87
2.3.1 Jacobian Matrix Estimation Algorithm . . . 88
2.3.2 Direct Method . . . 89
2.4 Hurst Exponent . . . 96
2.4.1 Introduction . . . 96
2.4.2 Implementation for the Hurst Exponent . . . 98
2.4.3 Random Walk . . . 102
2.5 Higuchi's Algorithm . . . 106
2.6 Complexity . . . 107

3 Autonomous Systems in the Plane 111
3.1 Classification of Fixed Points . . . 111
3.2 Homoclinic Orbit . . . 113
3.3 Pendulum . . . 114
3.4 Limit Cycle Systems . . . 116
3.5 Lotka-Volterra Systems . . . 119

4 Nonlinear Hamilton Systems 123
4.1 Hamilton Equations of Motion . . . 123
4.1.1 Hamilton System and Variational Equation . . . 126
4.2 Integrable Hamilton Systems . . . 127
4.2.1 Hamilton Systems and First Integrals . . . 127
4.2.2 Lax Pair and Hamilton Systems . . . 128
4.2.3 Floquet Theory . . . 130
4.3 Chaotic Hamilton Systems . . . 133
4.3.1 Hénon-Heiles Hamilton Function and Trajectories . . . 133
4.3.2 Hénon Heiles and Surface of Section Method . . . 135
4.3.3 Quartic Potential and Surface of Section Technique . . . 136

5 Nonlinear Dissipative Systems 139
5.1 Fixed Points and Stability . . . 139
5.2 Trajectories . . . 144
5.3 Phase Portrait . . . 148
5.4 Liapunov Exponents . . . 150
5.5 Generalized Lotka-Volterra Model . . . 153
5.6 Hyperchaotic Systems . . . 155
5.7 Hopf Bifurcation . . . 158
5.8 Time-Dependent First Integrals . . . 161

6 Nonlinear Driven Systems 163
6.1 Introduction . . . 163
6.2 Driven Anharmonic Systems . . . 166

6.2.1 Phase Portrait . . . 166
6.2.2 Poincaré Section . . . 167
6.2.3 Liapunov Exponent . . . 169
6.2.4 Autocorrelation Function . . . 170
6.2.5 Power Spectral Density . . . 173
6.3 Driven Pendulum . . . 174
6.3.1 Phase Portrait . . . 174
6.3.2 Poincaré Section . . . 176
6.4 Parametrically Driven Pendulum . . . 178
6.4.1 Phase Portrait . . . 178
6.4.2 Poincaré Section . . . 179
6.5 Driven Van der Pol Equation . . . 181
6.5.1 Phase Portrait . . . 181
6.5.2 Liapunov Exponent . . . 183
6.6 Parametrically and Externally Driven Pendulum . . . 185
6.7 Torsion Numbers . . . 187

7 Controlling of Chaos **191**
7.1 Introduction . . . 191
7.2 Ott-Yorke-Grebogi Method . . . 191
7.2.1 One-Dimensional Maps . . . 191
7.2.2 Systems of Difference Equations . . . 195
7.3 Small Periodic Perturbation . . . 199
7.4 Resonant Perturbation and Control . . . 201

8 Synchronization of Chaos **203**
8.1 Introduction . . . 203
8.2 Synchronization of Chaos . . . 203
8.2.1 Synchronization Using Control . . . 203
8.2.2 Synchronizing Subsystems . . . 206
8.3 Synchronization of Coupled Dynamos . . . 209
8.4 Phase Coupled Systems . . . 211

9 Fractals **217**
9.1 Introduction . . . 217
9.2 Iterated Function System . . . 219
9.2.1 Introduction . . . 219
9.2.2 Cantor Set . . . 220
9.2.3 Heighway's Dragon . . . 223
9.2.4 Sierpinski Gasket . . . 225
9.2.5 Koch Curve . . . 227
9.2.6 Fern . . . 229
9.2.7 Grey Level Maps . . . 231
9.3 Mandelbrot Set . . . 232
9.4 Julia Set . . . 234

9.5 Fractals and Kronecker Product . . . 236
9.6 Lindenmayer Systems and Fractals . . . 240
9.7 Weierstrass Function . . . 243

10 Cellular Automata 245
10.1 Introduction . . . 245
10.2 One-Dimensional Cellular Automata . . . 248
10.3 Sznajd Model . . . 249
10.4 Conservation Laws . . . 252
10.5 Two-Dimensional Cellular Automata . . . 253
10.6 Button Game . . . 257

11 Solving Differential Equations 261
11.1 Introduction . . . 261
11.2 Euler Method . . . 262
11.3 Lie Series Technique . . . 264
11.4 Runge-Kutta-Fehlberg Technique . . . 268
11.5 Ghost Solutions . . . 269
11.6 Symplectic Integration . . . 272
11.7 Verlet Method . . . 277
11.8 Störmer Method . . . 279
11.9 Invisible Chaos . . . 280
11.10 First Integrals and Numerical Integration . . . 281

12 Neural Networks 283
12.1 Introduction . . . 283
12.2 Hopfield Model . . . 287
12.2.1 Introduction . . . 287
12.2.2 Synchronous Operations . . . 289
12.2.3 Energy Function . . . 291
12.2.4 Basins and Radii of Attraction . . . 293
12.2.5 Spurious Attractors . . . 293
12.2.6 Hebb's Law . . . 294
12.2.7 Hopfield Example . . . 296
12.2.8 Hopfield C++ Program . . . 298
12.2.9 Asynchronous Operation . . . 302
12.2.10 Translation Invariant Pattern Recognition . . . 303
12.3 Similarity Metrics . . . 305
12.4 Kohonen Network . . . 309
12.4.1 Introduction . . . 309
12.4.2 Kohonen Algorithm . . . 310
12.4.3 Kohonen Example . . . 312
12.4.4 Traveling Salesman Problem . . . 318
12.5 Perceptron . . . 322
12.5.1 Introduction . . . 322

12.5.2 Boolean Functions 324
12.5.3 Linearly Separable Sets 325
12.5.4 Perceptron Learning 326
12.5.5 Perceptron Learning Algorithm 330
12.5.6 One and Two Layered Networks 333
12.5.7 XOR Problem and Two-Layered Networks 335
12.6 Multilayer Perceptrons 339
12.6.1 Introduction 339
12.6.2 Cybenko's Theorem 340
12.6.3 Back-Propagation Algorithm 340
12.6.4 Recursive Deterministic Perceptron Neural Networks 348
12.7 Chaotic Neural Networks 350
12.8 Neuronal-Oscillator Models 351
12.9 Radial Basis Function Networks 353
12.10 Neural Network, Matrices and Eigenvalues 355

13 Genetic Algorithms 357
13.1 Introduction 357
13.2 Sequential Genetic Algorithm 358
13.3 Schemata Theorem 362
13.4 Bitwise Operations 364
13.4.1 Introduction 364
13.4.2 Assembly Language 367
13.4.3 Floating Point Numbers and Bitwise Operations 369
13.4.4 Java Bitset Class 370
13.4.5 C++ bitset Class 371
13.5 Bit Vector Class 373
13.6 Penna Bit-String Model 376
13.7 Maximum of One-Dimensional Maps 378
13.8 Maximum of Two-Dimensional Maps 384
13.9 Finding a Fitness Function 392
13.10 Problems with Constraints 398
13.10.1 Introduction 398
13.10.2 Knapsack Problem 399
13.10.3 Traveling Salesman Problem 404
13.11 Simulated Annealing 412

14 Gene Expression Programming 415
14.1 Introduction 415
14.2 Example 418
14.3 Numerical-Symbolic Manipulation 430
14.4 Multi Expression Programming 435

15 Optimization **441**
15.1 Lagrange Multiplier Method . . . 441
15.2 Karush-Kuhn-Tucker Conditions . . . 449
15.3 Support Vector Machine . . . 453
15.3.1 Introduction . . . 453
15.3.2 Linear Decision Boundaries . . . 453
15.3.3 Nonlinear Decision Boundaries . . . 457
15.3.4 Kernel Fisher Discriminant . . . 461

16 Discrete Wavelets **465**
16.1 Introduction . . . 465
16.2 Multiresolution Analysis . . . 468
16.3 Pyramid Algorithm . . . 470
16.4 Biorthogonal Wavelets . . . 475
16.5 Two-Dimensional Wavelets . . . 480

17 Discrete Hidden Markov Processes **483**
17.1 Introduction . . . 483
17.2 Markov Chains . . . 485
17.3 Discrete Hidden Markov Processes . . . 489
17.4 Forward-Backward Algorithm . . . 493
17.5 Viterbi Algorithm . . . 496
17.6 Baum-Welch Algorithm . . . 497
17.7 Distances between HMMs . . . 498
17.8 Application of HMMs . . . 499
17.9 C++ Program . . . 502

18 Fuzzy Sets and Fuzzy Logic **513**
18.1 Introduction . . . 513
18.2 Operators for Fuzzy Sets . . . 521
18.2.1 Logical Operators . . . 521
18.2.2 Algebraic Operators . . . 524
18.2.3 Defuzzification Operators . . . 525
18.2.4 Fuzzy Concepts as Fuzzy Sets . . . 527
18.2.5 Hedging . . . 528
18.2.6 Quantifying Fuzzyness . . . 529
18.2.7 C++ Implementation of Discrete Fuzzy Sets . . . 530
18.2.8 Applications: Simple Decision-Making Problems . . . 549
18.3 Fuzzy Numbers and Fuzzy Arithmetic . . . 555
18.3.1 Introduction . . . 555
18.3.2 Algebraic Operations . . . 556
18.3.3 LR-Representations . . . 559
18.3.4 Algebraic Operations on Fuzzy Numbers . . . 562
18.3.5 C++ Implementation of Fuzzy Numbers . . . 563
18.3.6 Applications . . . 570

18.4 Fuzzy Rule-Based Systems . . . 571
18.4.1 Introduction . . . 571
18.4.2 Fuzzy If-Then Rules . . . 574
18.4.3 Inverted Pendulum Control System . . . 575
18.4.4 Fuzzy Controllers with B-Spline Models . . . 577
18.4.5 Application . . . 580
18.5 Fuzzy C-Means Clustering . . . 582
18.6 fXOR Fuzzy Logic Networks . . . 586
18.7 Fuzzy Hamming Distance . . . 588
18.8 Fuzzy Truth Values and Probabilities . . . 591

Bibliography **593**

Index **601**

This page intentionally left blank

Symbol Index

$\emptyset$	empty set
$A \subset B$	the subset A of the set B
$A \cap B$	the intersection of the sets A and B
$A \cup B$	the union of the sets A and B
χ_A	indicator function
$\mathbf{Z}$	the set of integers
$\mathbf{N}$	the set of positive integers: natural numbers
$\mathbf{Q}$	the set of rational numbers
$\mathbf{R}$	the set of real numbers
$\mathbf{R}^+$	nonnegative real numbers
$\mathbf{C}$	the set of complex numbers
$\mathbf{R}^n$	the n-dimensional real linear space
$\mathbf{C}^n$	the n-dimensional complex linear space
$\{0,1\}^n$	n-dimensional binary hypercube
$[a,b]$	closed interval of real-valued numbers between a and b
$[0,1]$	unit interval
S^1	unit circle $\{ (x,y) : x^2 + y^2 = 1 \}$
f	mapping (map)
$f \circ g$	composition of mappings $(f \circ g)(x) = f(g(x))$
$f^{(n)}$	n-th iterate of mapping f
i	$:= \sqrt{-1}$
$\Re z$	real part of the complex number z
$\Im z$	imaginary part of the complex number z
$\mathbf{x} \in \mathbf{R}^n$	the element $\mathbf{x}$ of $\mathbf{R}^n$, column vector
$\mathbf{x}^T$	transpose of $\mathbf{x}$ (row vector)
$\mathbf{x}^T\mathbf{y}$	scalar product in $\mathbf{R}^n$
t	time: discrete and continuous depending on context
$\mathbf{x}$	dependent variable: discrete systems
$\mathbf{x}^*$	fixed point
$\mathbf{u}$	dependent variable: continuous systems
ρ	invariant density
r	bifurcation parameter, control parameter
$\| . \|$	norm
tr	trace of a square matrix
det	determinant of a square matrix
H	Hamilton function
δ_{jk}	Kronecker delta with $\delta_{jk} = 1$ for $j = k$ and $\delta_{jk} = 0$ for $j \neq k$
λ	eigenvalue

η	learning rate
I	identity matrix
$\mathbf{w}$	weight vector (column vector)
$\Delta\mathbf{w}$	small change applied to $\mathbf{w}$
W	weight matrix
$\boldsymbol{\Theta}$	bias vector
$\{\mathbf{x}_k, \mathbf{d}_k\}$	kth training pairs
net	weighted sum or $\mathbf{w}^T\mathbf{x}$
$f(net)$	differentiable activation function, usually a sigmoid function
$f'(net)$	derivative of f with respect to net
$\mu_{\widetilde{A}}$	membership function (fuzzy logic)
$\oplus$	XOR bitwise operation
$\otimes$	Kronecker product of matrices
$\wedge$	wedge product (exterior product, Grassmann product)
$[A, B]$	commutator of $n \times n$ matrices A and B

Comments to the Programs

The C++ programs comply to the ANSI C++ standard. Thus they should run under all compilers.

SymbolicC++ version 3 is a symbolic manipulation tool (Y. Hardy, W.-H. Steeb, Tan Kiat Shi) written completely in ANSI C++. It includes classes (abstract data types) to do symbolic and numeric manipulations. The classes include

`Symbolic, Rational, Verylong, Quaternion,`
`Derive, Vector, Matrix, Array, Polynomial .`

The classes from the Standard Template Library are also extensively used and so is the `string` class from C++. The symbolic manipulation is done using the class `Symbolic`.

SymbolicC++ version 3 is available at

`http://issc.uj.ac.za/symbolic/symbolic.html`

We have tested the C++ programs with GCC 4.1.3 and Microsoft Visual Studio.net (VC8).

In C++ graphics does not belong to the standard. Here we use GnuPlot (GNU-PLOT Copyright(c) 1986-1993, 1998, Colin Kelly and Thomas Williams) to draw the figures.

The Java programs have been tested with JDK 1.6. The JDK (Java Development Kit) is a product of Sun Microsystems, Inc. The JDK allows us to develop applets that will run in browsers supporting the Java platform 1.6. The Java tools we using are the Java Compiler (`javac`) which compiles programs written in the Java programming language into bytecodes and the Java Interpreter (`java`) that executes Java bytecodes. In other words, it runs programs written in the Java programming language. AppletViewer allows us to run one or more Java applets that are called references in a web page (HTML file) using the APPLET tag. The AppletViewer finds the APPLET tags in the HTML file and runs the applets (in separate windows) as specified by the tags.

Most of the programs are written so that they can be understood by beginners. Thus some of the programs can be improved and written in a more sophisticated manner.