

Simulating, Analyzing, and Animating Dynamical Systems

Software • Environments • Tools

The series includes handbooks and software guides as well as monographs on practical implementation of computational methods, environments, and tools. The focus is on making recent developments available in a practical format to researchers and other users of these methods and tools.

Editor-in-Chief

Jack J. Dongarra
University of Tennessee and Oak Ridge National Laboratory

Editorial Board

James W. Demmel, University of California, Berkeley
Dennis Gannon, Indiana University
Eric Grosse, AT&T Bell Laboratories
Ken Kennedy, Rice University
Jorge J. Moré, Argonne National Laboratory

Software, Environments, and Tools

Bard Ermentrout, *Simulating, Analyzing, and Animating Dynamical Systems: A Guide to XPPAUT for Researchers and Students*

V. A. Barker, L. S. Blackford, J. Dongarra, J. Du Croz, S. Hammarling, M. Marinova, J. Waśniewski, and P. Yalamov, *LAPACK95 Users' Guide*

Stefan Goedecker and Adolfo Hoisie, *Performance Optimization of Numerically Intensive Codes*

Zhaojun Bai, James Demmel, Jack Dongarra, Axel Ruhe, and Henk van der Vorst, *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*

Lloyd N. Trefethen, *Spectral Methods in MATLAB*

E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide, Third Edition*

Michael W. Berry and Murray Browne, *Understanding Search Engines: Mathematical Modeling and Text Retrieval*

Jack J. Dongarra, Iain S. Duff, Danny C. Sorensen, and Henk A. van der Vorst, *Numerical Linear Algebra for High-Performance Computers*

R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*

Randolph E. Bank, *PLTMG: A Software Package for Solving Elliptic Partial Differential Equations, Users' Guide 8.0*

L. S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R. C. Whaley, *ScaLAPACK Users' Guide*

Greg Astfalk, editor, *Applications on Advanced Architecture Computers*

Françoise Chaitin-Chatelin and Valérie Frayssé, *Lectures on Finite Precision Computations*

Roger W. Hockney, *The Science of Computer Benchmarking*

Richard Barrett, Michael Berry, Tony F. Chan, James Demmel, June Donato, Jack Dongarra, Victor Eijkhout, Roldan Pozo, Charles Romine, and Henk van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*

E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, *LAPACK Users' Guide, Second Edition*

Jack J. Dongarra, Iain S. Duff, Danny C. Sorensen, and Henk van der Vorst, *Solving Linear Systems on Vector and Shared Memory Computers*

J. J. Dongarra, J. R. Bunch, C. B. Moler, and G. W. Stewart, *Linpack Users' Guide*

Simulating, Analyzing, and Animating Dynamical Systems

A Guide to XPPAUT for Researchers and Students

Bard Ermentrout

University of Pittsburgh
Pittsburgh, Pennsylvania



Society for Industrial and Applied Mathematics
Philadelphia

Copyright © 2002 by the Society for Industrial and Applied Mathematics

10 9 8 7 6 5 4 3 2 1

All rights reserved. Printed in the United States of America. No part of this book may be reproduced, stored, or transmitted in any manner without the written permission of the publisher. For information, write to the Society for Industrial and Applied Mathematics, 3600 University City Science Center, Philadelphia, PA 19104-2688.

Library of Congress Cataloging-in-Publication Data

Ermentrout, Bard.

Simulating, analyzing, and animating dynamical systems : a guide to XPPAUT for researchers and students / Bard Ermentrout.

p. cm. — (Software, environments, tools)

Includes bibliographical references and index.

ISBN 0-89871-506-7 (pbk.)

Differential equations—Numerical solutions—Data processing. I. Title. II. Series.

QA371.5.D37 E76 2002

515'.352—dc21

2002017639

Exceed is a trademark of Hummingbird Ltd.

Excel, Paint, Windows, Windows NT, WinZip, Word, and MS-DOS are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries.

Ghostscript is a trademark of Artifex Software Inc.

HP is a trademark of the Hewlett-Packard Company.

Linux is a registered trademark of Linus Torvalds.

Mac and QuickTime are trademarks of Apple Computer, Inc., registered in the U.S. and other countries.

Simulating, Analyzing, and Animating Dynamical Systems: A Guide to XPPAUT for Researchers and Students is an independent publication and has not been authorized, sponsored, or otherwise approved by Apple Computer, Inc.

Netscape is a registered trademark of Netscape Communications Corporation in the United States and other countries.

PostScript is a registered trademark of Adobe Systems Inc. in the United States and/or other countries.

RealPlayer is a trademark of RealNetworks, Inc.

Sun, Sun Microsystems, the Sun logo, Java, and Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Times Roman is a trademark of Heidelberger Druckmaschinen AG, which may be registered in certain jurisdictions, exclusively licensed through Linotype Library GmbH, a wholly owned subsidiary of Heidelberger Druckmaschinen AG.

UNIX and the X Window System are either trademarks or registered trademarks of The Open Group in the United States and other countries.

XPPAUT is distributed as is. The author makes no claims as to the performance of the program. Anyone is allowed to modify and distribute XPPAUT as long as the original code is also made available.

About the cover: Pictured in the foreground is the manufacturer's schematic of the design for the Zipper, a popular carnival ride. This drawing also appears in Figure 8.3 of Chapter 8, "Animation." The background photograph shows the actual Zipper ride. Used with permission of Chance Rides, Wichita, KS.

Contents

List of Figures	ix
Preface	xiii
1 Installation	1
1.1 Installation on UNIX	1
1.1.1 Installation from the source code	1
1.1.2 Installation from binaries	2
1.1.3 Additional UNIX setup	2
1.2 Native Version for Windows 95, 98, 2000, and Windows NT	3
1.3 X Window version on Windows	3
1.4 Installation on Mac OS X	5
1.5 Environment variables	6
1.5.1 Resource file	6
2 A Very Brief Tour of <i>XPPAUT</i>	9
2.1 Creating the ODE file	9
2.2 Running the program	11
2.2.1 The main window	11
2.2.2 Quitting the program	12
2.3 Solving the equation, graphing, and plotting	12
2.4 Changing parameters and initial data	16
2.5 Looking at the numbers: The data viewer	18
2.6 Saving and restoring the state of <i>XPPAUT</i>	18
2.6.1 Command summary	19
2.7 A nonlinear equation	20
2.7.1 Direction fields	21
2.7.2 Nullclines and fixed points	22
2.7.3 Command summary	24
2.8 The most important numerical parameters	24
2.9 Exercises	25
3 Writing ODE Files for Differential Equations	27
3.1 Introduction	27

3.2	ODEs and maps	28
3.2.1	Nonautonomous systems	29
3.3	Functions	30
3.3.1	User-defined functions	31
3.4	Auxiliary and temporary quantities	32
3.4.1	Fixed variables	33
3.4.2	Exercises	34
3.5	Discontinuous differential equations	35
3.5.1	Integrate-and-fire models	35
3.5.2	Clocks: Regular and irregular	37
3.5.3	The dripping faucet	40
3.5.4	Exercises	41
4	<i>XPPAUT</i> in the Classroom	43
4.1	Plotting functions	43
4.2	Discrete dynamics in one dimension	46
4.2.1	Bifurcation diagrams	47
4.2.2	Periodic points	49
4.2.3	Liapunov exponents for one-dimensional maps	51
4.2.4	The devil's staircase	52
4.2.5	Complex maps in one dimension	54
4.2.6	Iterated function systems	59
4.3	One-dimensional ODEs	62
4.3.1	Nonautonomous one-dimensional systems	65
4.4	Planar dynamical systems	67
4.5	Nonlinear systems	69
4.5.1	Conservative dynamical systems in the plane	72
4.5.2	Exercises	75
4.6	Three and higher dimensions	77
4.6.1	Poincaré maps, FFTs, and chaos	79
4.6.2	Poincaré maps	81
4.6.3	Exercises	82
5	More Advanced Differential Equations	87
5.1	Functional equations	87
5.1.1	Delay equations	87
5.1.2	Integral equations	90
5.1.3	Making three-dimensional movies	92
5.1.4	Singular integral equations	93
5.1.5	Some amusing tricks	93
5.1.6	Exercises	96
5.2	Stochastic equations	98
5.2.1	Markov processes	99
5.2.2	Gillespie's method	102
5.2.3	Ratchets and games	107
5.2.4	Spike-time statistics	110

5.2.5	Exercises	113
5.3	Differential algebraic equations	114
5.3.1	Exercises	116
6	Spatial Problems, PDEs, and BVPs	119
6.1	BVPs	119
6.1.1	Solving the BVP	122
6.1.2	Infinite domains	125
6.1.3	Exercises	132
6.2	PDEs and arrays	136
6.2.1	An animation file	139
6.2.2	The issue of stiffness	140
6.2.3	Special operators	142
6.2.4	Exercises	155
7	Using AUTO: Bifurcation and Continuation	161
7.1	Standard examples	163
7.1.1	A limit cycle	166
7.1.2	A “real” example	169
7.1.3	Exercises	173
7.2	Maps, BVPs, and forced systems	174
7.2.1	Maps	175
7.2.2	Exercises	177
7.3	BVPs	178
7.3.1	Homoclinics and heteroclinics	181
7.3.2	Exercises	187
7.4	Periodically forced equations	188
7.4.1	Exercises	190
7.5	Importing the diagram into <i>XPPAUT</i>	191
7.5.1	Exercise	193
8	Animation	195
8.1	Introduction	195
8.2	Some first examples	195
8.2.1	The pendulum	195
8.2.2	Important notes	199
8.2.3	A spatial problem revisited	199
8.2.4	A glider and a fancy glider	200
8.2.5	Coupled oscillators	203
8.3	My favorites	205
8.3.1	More pendula	205
8.3.2	A rollercoaster	207
8.3.3	The Zipper	209
8.3.4	The Lorenz equations	212
8.4	The animator scripting language	214

9	Tricks and Advanced Methods	217
9.1	Introduction	217
9.2	Graphics tricks	217
9.2.1	Better plots	217
9.2.2	Plotting results from range integration	219
9.3	Fitting a simulation to external data	220
9.4	The data browser as a spreadsheet	221
9.5	Oscillators, phase models, and averaging	223
9.5.1	Computing a limit cycle and the adjoint	224
9.5.2	Averaging	225
9.5.3	Phase response curves	226
9.5.4	Phase models	228
9.6	Arcana	232
9.6.1	Iterating with fixed variables	232
9.6.2	Timers	233
9.6.3	Initial data depending on parameters	235
9.6.4	Poincaré maps revisited	240
9.7	Don't forget...	241
9.8	Dynamic linking with external C routines	244
9.8.1	An array example	247
	Appendix A. Colors and Linestyles	249
	Appendix B. The Options	251
	Appendix C. Numerical Methods	255
C.1	Fixed points and stability	255
C.2	Integrators	256
C.2.1	Delay equations	259
C.2.2	The Volterra integrator	260
C.3	How AUTO works	260
	Appendix D. Structure of ODE Files	263
	Appendix E. Complete Command List	269
E.1	Main menu	269
E.2	AUTO	270
E.3	Browser commands	271
	Appendix F. Error Messages	273
	Appendix G. Cheat Sheet	275
	References	283
	Index	287

List of Figures

2.1	The main <i>XPPAUT</i> window.	11
2.2	The equation window.	12
2.3	Phase-plane for the linear two-dimensional problem.	13
2.4	The initial conditions window.	14
2.5	File selector.	15
2.6	The parameter window.	17
2.7	Top: Unused parameter slider. Bottom: Used parameter slider.	17
2.8	The data viewer.	19
2.9	Direction fields and some trajectories for the Fitzhugh–Nagumo equations	21
2.10	Nullclines, direction fields, and trajectories for $I = 0.4$ in the Fitzhugh–Nagumo equations.	22
3.1	Phase-plane of the forced Duffing equation.	30
3.2	A pair of integrate-and-fire neurons coupled with alpha-function synapses. With slow excitatory coupling, synchrony is unstable; with fast inhibition, synchrony is also unstable; with slow inhibition, synchrony is stable. . .	38
3.3	A kicked clock and its chaotic brother.	39
4.1	Three-dimensional plot illustrating a projection on the coordinate plane. .	46
4.2	A cobweb plot of the logistic map.	48
4.3	The logistic map showing an orbit diagram on the left and a plot of the periodic orbits of period 3 and period 5 on the right.	49
4.4	Maximal Liapunov exponent as a function of the parameter a for the logistic map.	53
4.5	Rotation number for the standard map and an expanded view over a small range of the parameter b	55
4.6	Mandelbrot set.	60
4.7	Direction fields and sample trajectories for the equation $x' = x(1 - x)$. .	66
4.8	Phase-plane showing trajectories and nullclines for the two-dimensional nonlinear example. Right: The stable and unstable manifolds of the fixed points.	71
4.9	Direction fields and nullclines of a predator–prey example.	72
4.10	Colorized conservative system: Greyscale indicates log of the integral. .	74
4.11	Double well potential.	75
4.12	The Lorenz attractor.	79

4.13	Poincaré map for the Lorenz attractor looking at successive values of the maximum of z . Period 1, 2, and 3 points are illustrated.	83
5.1	The initial condition window for delay equations.	89
5.2	Solutions to Ken Cooke's delay equation.	90
5.3	Three-dimensional reconstruction of the chaotic integral equation.	92
5.4	Solution to the Fabry–Perot equations in the chaotic regime.	96
5.5	The probability of existence of the mutant species as a function of time.	101
5.6	The Gillespie method applied to the Brusselator (left) and to a membrane model.	105
5.7	An asymmetric periodic potential.	107
5.8	Illustration of Parrondo's paradox—games one and two are losers but when played together become winners.	110
5.9	PSTH for the noisy theta neuron (left) and the spike count distributions for a Poisson process with rate 0.05 and the same process with a refractory period.	112
6.1	The boundary condition window.	121
6.2	The connection of the unstable (dashed) and stable (solid) manifolds of a fixed point as the parameter λ varies.	126
6.3	Stable and unstable manifolds for the fixed point (1,0) of the Fisher equation.	129
6.4	Establishing shooting sets for the bistable reaction-diffusion equation.	131
6.5	Steady-state behavior of a nonlinear dendrite. Top: Some representative solutions. Bottom: Flux at $x = 0$ as a function of \bar{g}	134
6.6	The array plot window showing the space-time behavior of the cable equation.	138
6.7	Solutions to the cable equation at four different times.	139
6.8	Solutions to the CGL equation.	142
6.9	Solutions to the delayed neural net equation with a delay of 1.	147
6.10	A dendritic tree rendered in the animation window.	153
6.11	Cellular automata simulation.	155
7.1	The AUTO window.	164
7.2	One- and two-parameter bifurcation diagrams for the cusp ODE, $x' = a + bx - x^3$. The one-parameter diagram has $b = 1$. The two-parameter diagram also shows the artifactual line at $b = 1$ from the one-parameter continuation.	166
7.3	The diagram for the isola example. The left figure shows the maximum and minimum of the stable (filled circles) and unstable (empty circles) limit cycles. The right figure shows the norm of the limit cycles and the stable fixed point at 0.	169
7.4	Various diagrams associated with the Morris–Lecar equation. Top left: One-parameter diagram showing fixed points and limit cycles. Top right: Frequency of the periodic orbits. Bottom: Two-parameter diagram showing the curve of Hopf points and (straight line) the lines of saddle-node points. These are independent of the parameter ϕ	171

7.5 Bifurcation diagram for period 8 orbits of the logistic map (left) and period 7 orbits for the delayed logistic map (right). 176

7.6 Solution to the BVPs from Example 1 (left) and Example 2 (right). . . . 179

7.7 Left: The continuation of the homoclinic orbit for the example problem. Right: Sample orbits computed for $a = 0, \pm 6$ in the (x, y) phase-plane. . 184

7.8 Bifurcation diagram for the periodically driven bistable system. 190

7.9 Bursting in a modified membrane model. Top left: The voltage as a function of time during a burst. Top right: The slow parameter I and the voltage. Bottom: Superimposition of the bifurcation diagram in the (I, v) -plane. 192

8.1 The animation window. 196

8.2 The fancy glider! 202

8.3 The Zipper carnival ride. Left shows the manufacturer’s schematic; right shows the Zipper as rendered by XPP. 210

9.1 Curve fit to experimental kinetics model. Top: The initial guess. Bottom: The parameters found by curve fitting. 222

9.2 Top: The PRC for the van der Pol oscillator with different amplitudes. Bottom: The PRC for the Morris–Lecar model. 229

9.3 The basin boundaries for the iteration which arises from the application of Newton’s method to the equation $z^3 = 1$ in the complex plane. Each point in the plane is colored according to the root that it converges to. . . 237

A.1 Available PostScript linestyles. 249

C.1 Error from Euler (left) and improved Euler (right). 257

Preface

XPPAUT is a tool for simulating, animating, and analyzing dynamical systems. The program evolved from a DOS program that was originally written so that John Rinzel and I could easily illustrate the dynamics of a simple model for an excitable membrane. The DOS program, PHASEPLANE, became a commercial project and was used for many years by a number of patient folks. In the early 1990s, I spent a month in a beautiful office at the Mathematical Sciences Research Institute as part of a Mathematical Biology program. During the evenings, I ported the DOS program to X Window on a UNIX environment while enjoying the sunset and listening to the same cassette tape over and over. (I forget what it was.) The program has evolved a great deal from those early years and is available at no cost to anyone who wishes to download it. I have also successfully compiled the X version to run under various 32-bit flavors of Windows and also under the new Mac OS X.

I have added lots of integrators and tools as well as my own idiosyncratic interface to the amazing continuation package AUTO. Most things that you might want to do that concern dynamics—either discrete or continuous—can probably be done with *XPPAUT* if you know a few of the tricks. That is the point of this book; I suspect many users do not take full advantage of the features of the program. This is mostly my fault, as the users' manual that is distributed with the program, while comprehensive in its description of all the features, is hopelessly baroque in its organization.

Why should anyone want to use *XPPAUT*? There are plenty of packages that will integrate differential equations for you. Many people use MATLAB, MAPLE, or MATHEMATICA to study and analyze dynamical systems. These are all general purpose packages that have the capability to do most everything that is described in this book. However, the latter two symbolic packages are extremely slow when it comes to numerically solving differential equations. Furthermore, they do not offer much flexibility in the choice of integration methods, and the integration is not done interactively. That is, you cannot see the progress of the solution until it is computed. Standard qualitative tools such as direction fields and nullclines require running additional packages or writing by hand. MATLAB has great flexibility and can even integrate differential equations with discontinuities such as the integrate-and-fire equations. However, the numerical integration is generally slower than can be achieved with *XPPAUT*. None of the packages offers an interface to AUTO, the main reason that some people use *XPPAUT*. The syntax of *XPPAUT* for setting up differential equations is pretty simple compared to the other programs. Finally, *XPPAUT* is free—no license demons crashing once a year, no guilt copying to another computer, and the source code is always there for the taking. To download your copy, go to the *XPPAUT* homepage at <http://www.math.pitt.edu/~bard/xpp/xpp.html>.

How to use this book. This book is written to be used by either a researcher or modeler who wants to simulate and analyze a particular system or by students as an adjunct to a modeling class or a class in differential equations. I have used it in many such courses both at the sophomore engineering level up through the graduate level in a dynamical systems class. I have used *XPPAUT* in applied courses for students in neuroscience and physiology. The present book contains many examples and many exercises. Along the way, I hope that it can aid in teaching certain concepts in the analysis of the behavior of differential equations. Most of the problems and examples are taken from research papers. The emphasis, I am afraid, is skewed toward biological applications, as that is what I do.

If all you want to do is solve differential equations and graph the solutions, then most of the information you will need can be found in Chapter 3. Suggestions for how you can use *XPPAUT* in a classroom setting are found in Chapter 4. Research problems involve a more complete set of tools: see Chapter 5 for functional and stochastic differential equations, and Chapter 6 for how to discretize and solve partial differential equations. Boundary value problems are also covered in Chapter 6. Chapter 7 introduces bifurcation theory and the use of the AUTO interface in *XPPAUT*. Chapter 8 shows you how to make animations with the built-in animator, and Chapter 9 shows other ways to make animations. Tricks and special classes of differential equations also are described in Chapter 9.

Acknowledgments. I have benefitted a great deal from the many users of various versions of the program. To those one or two of you who sent me a note about how useful the program is rather than what new features you wished I'd put into it or which ones didn't work, I salute you. To the others, well, I reluctantly thank you, as your comments motivated new features and bugs. Mostly, I thank John Rinzel and Artie Sherman for being guinea pigs for many versions of the program that have appeared throughout the years. Finally, I want to thank my wife, Ellen, for her patience, and the boys, Kyle and Jordan, without whom this book would have a 2000 copyright.

Bard Ermentrout