



**Manchester
Metropolitan
University**

Lynch, Stephen (2018) Dynamical Systems with Applications using Python.
Birkhäuser. ISBN 9783319781457

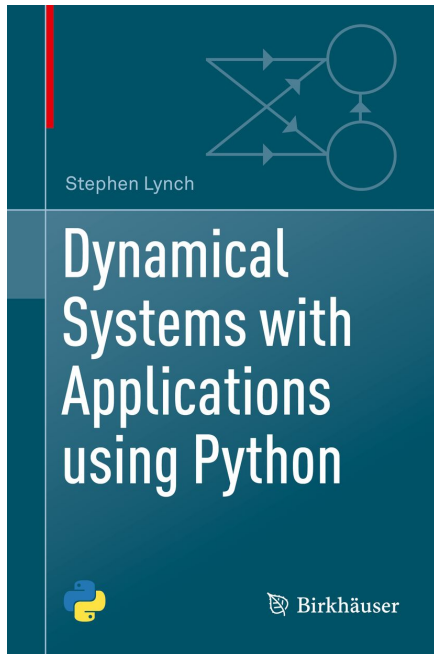
Downloaded from: <http://e-space.mmu.ac.uk/622016/>

Publisher: Birkhäuser

DOI: <https://doi.org/10.1007/978-3-319-78145-7>

Please cite the published version

<https://e-space.mmu.ac.uk>



1st ed. 2018, XVI, 665 p. 277 illus., 118 illus. in color.

Printed book

Hardcover

79,99 € | £69.99 | \$89.99

^[1]85,59 € (D) | 87,99 € (A) | CHF

94,50

eBook

67,82 € | £55.99 | \$69.99

^[2]67,82 € (D) | 67,82 € (A) | CHF

75,50

Available from your library or

springer.com/shop

MyCopy ^[3]

Printed eBook for just

€ | \$ 24.99

springer.com/mycopy

Stephen Lynch

Dynamical Systems with Applications using Python

- Designed for a broad audience of students in applied mathematics, physics, and engineering
- Represents dynamical systems with popular Python libraries like sympy, numpy, and matplotlib
- Explores a variety of advanced topics in dynamical systems, like neural networks, fractals, and nonlinear optics, at an undergraduate level

This textbook provides a broad introduction to continuous and discrete dynamical systems. With its hands-on approach, the text leads the reader from basic theory to recently published research material in nonlinear ordinary differential equations, nonlinear optics, multifractals, neural networks, and binary oscillator computing. Dynamical Systems with Applications Using Python takes advantage of Python's extensive visualization, simulation, and algorithmic tools to study those topics in nonlinear dynamical systems through numerical algorithms and generated diagrams. After a tutorial introduction to Python, the first part of the book deals with continuous systems using differential equations, including both ordinary and delay differential equations. The second part of the book deals with discrete dynamical systems and progresses to the study of both continuous and discrete systems in contexts like chaos control and synchronization, neural networks, and binary oscillator computing. These later sections are useful reference material for undergraduate student projects. The book is rounded off with example coursework to challenge students' programming abilities and Python-based exam questions. This book will appeal to advanced undergraduate and graduate students, applied mathematicians, engineers, and researchers in a range of disciplines, such as biology, chemistry, computing, economics, and physics.

Lifelong 40% discount for authors



Order online at springer.com / or for the Americas call (toll free) 1-800-SPRINGER / or email us at: customerservice@springernature.com. / For outside the Americas call +49 (0) 6221-345-4301 / or email us at: customerservice@springernature.com.

The first € price and the £ and \$ price are net prices, subject to local VAT. Prices indicated with [1] include VAT for books; the €(D) includes 7% for Germany, the €(A) includes 10% for Austria. Prices indicated with [2] include VAT for electronic products; 19% for Germany, 20% for Austria. All prices exclusive of carriage charges. Prices and other details are subject to change without notice. All errors and omissions excepted. [3] No discount for MyCopy.

Stephen Lynch

Dynamical Systems with Applications using Python

 Birkhäuser

Stephen Lynch
Manchester Metropolitan University
Manchester, UK

ISBN 978-3-319-78144-0 ISBN 978-3-319-78145-7 (eBook)
<https://doi.org/10.1007/978-3-319-78145-7>

Library of Congress Control Number: 2018952351

Mathematics Subject Classification (2010): 00A05, 00A69, 37-01, 34-01, 34Cxx, 34Dxx, 34H10, 34K18, 37-04, 37Nxx, 68U10, 78A60, 92B20

© Springer International Publishing AG, part of Springer Nature 2018

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This book is published under the imprint Birkhäuser, www.birkhauser-science.com by the registered company Springer Nature Switzerland AG

The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This book provides an introduction to the theory of dynamical systems with the aid of Python. It is written for both senior undergraduates and graduate students. Chapter 1 provides a tutorial introduction to Python—new users should go through this chapter carefully while those moderately familiar and experienced users will find this chapter a useful source of reference. The first part of the book deals with continuous systems using differential equations, including both ordinary and delay differential equations (Chapters 2–12), the second part is devoted to the study of discrete systems (Chapters 13–17), and Chapters 18–21 deal with both continuous and discrete systems. Chapter 22 gives examples of coursework and also lists three Python-based examinations to be sat in a computer laboratory with access to Python. Chapter 23 lists answers to all of the exercises given in the book. It should be pointed out that dynamical systems theory is not limited to these topics but also encompasses partial differential equations, integral and integro-differential equations, and stochastic systems, for instance. References [1–6] given at the end of the Preface provide more information for the interested reader. The author has gone for breadth of coverage rather than fine detail and theorems with proofs are kept at a minimum. The material is not clouded by functional analytic and group theoretical definitions, and so is intelligible to readers with a general mathematical background. Some of the topics covered are scarcely covered elsewhere. Most of the material in Chapters 9–12 and 16–21 is at postgraduate level and has been influenced by the author’s own research interests. There is more theory in these chapters than in the rest of the book since it is not easily accessed anywhere else. It has been found that these chapters are especially useful as reference material for senior undergraduate project work. The theory in other chapters of the book is dealt with more comprehensively in other texts, some of which may be found in the references section of the corresponding chapter. The book has a very hands-on approach and takes the reader from the basic theory right through to recently published research material.

Python is extremely popular with a wide range of researchers from all sorts of disciplines; it has a very user-friendly interface and has extensive visualization and numerical computation capabilities. It is an ideal package to adopt for the study of nonlinear dynamical systems; the numerical algorithms work very quickly, and complex pictures can be plotted within seconds.

The first chapter provides an efficient tutorial introduction to Python. Simple Python programming is introduced using three basic programming structures: defining functions, for loops, and if, then, else constructs. New users will find the tutorials will enable them to become familiar with Python within a few days. Both engineering and mathematics students appreciate this method of teaching and I have found that it generally works well with one staff member to about twenty students in a computer laboratory. In most cases, I have chosen to list the Python commands at the end of each chapter; this avoids unnecessary cluttering in the text. The Python programs have been kept as simple as possible and should run under later versions of the package. All Python files for the book (including updates and extra files) can even be downloaded from the Web via GitHub at:

<https://github.com/springer-math/dynamical-systems-with-applications-using-python>

Readers will find that they can reproduce the figures given in the text, and then it is not too difficult to change parameters or equations to investigate other systems.

Chapters 2–12 deal with continuous dynamical systems. Chapters 2 and 3 cover some theory of ordinary differential equations and applications to models in the real world are given. The theory of differential equations applied to chemical kinetics and electric circuits is introduced in some detail. The memristor is introduced and one of the most remarkable stories in the history of mathematics is relayed. Chapter 2 ends with the existence and uniqueness theorem for the solutions of certain types of differential equations. The theory behind the construction of phase plane portraits for two-dimensional systems is dealt with in Chapter 3. Applications are taken from chemical kinetics, economics, electronics, epidemiology, mechanics, and population dynamics. The modeling of the populations of interacting species is discussed in some detail in Chapter 4 and domains of stability are discussed for the first time. Limit cycles, or isolated periodic solutions, are introduced in Chapter 5. Since we live in a periodic world, these are the most common type of solution found when modeling nonlinear dynamical systems. They appear extensively when modeling both the technological and natural sciences. Hamiltonian, or conservative, systems and stability are discussed in Chapter 6, and Chapter 7 is concerned with how planar systems vary depending upon a parameter. Bifurcation, bistability, multistability, and normal forms are discussed.

The reader is first introduced to the concept of chaos in continuous systems in Chapters 8 and 9, where three-dimensional systems and Poincaré maps are investigated. These higher-dimensional systems can exhibit strange attractors and chaotic dynamics. One can rotate the three-dimensional objects in Python and plot time series plots to get a better understanding of the dynamics involved. Once again, the theory can be applied to chemical kinetics (including stiff systems), electric circuits, and epidemiology; a simplified model for the weather is also briefly discussed. Chapter 9 deals with Poincaré first return maps that can be used to untangle complicated interlacing trajectories in higher-dimensional spaces. A periodically driven nonlinear pendulum is also investigated by means of a nonautonomous differential equation. Both local and global bifurcations are investigated in Chapter 10. The main results and statement of the famous second part of David Hilbert's sixteenth problem are listed in Chapter 11. In order to understand these results, Poincaré compactification is introduced. The study of continuous systems ends with one of the authors specialities—limit cycles of Liénard systems. There is some detail on Liénard systems, in particular, in this part of the book, but they do have a ubiquity for systems in the plane. Chapter 12 provides an introduction to delay differential equations with applications in biology and nonlinear optics.

Chapters 13–17 deal with discrete dynamical systems. Chapter 13 starts with a general introduction to iteration and linear recurrence (or difference) equations. The bulk of the chapter is concerned with the Leslie model used to investigate the population of a single species split into different age classes. Harvesting and culling policies are then investigated and optimal solutions are sought. Nonlinear discrete dynamical systems are dealt with in Chapter 14. Bifurcation diagrams, chaos, intermittency, Lyapunov exponents, periodicity, quasiperiodicity, and universality are some of the topics introduced. The theory is then applied to real-world problems from a broad range of disciplines including population dynamics, biology, economics, nonlinear optics, and neural networks. Chapter 15 is concerned with complex iterative maps in the Argand plane, where Julia sets and the now-famous Mandelbrot set are plotted. Basins of attraction are investigated for these complex systems and Newton fractals are introduced. As a simple introduction to optics, electromagnetic waves and Maxwell's equations are studied at the beginning of Chapter 16. Complex iterative equations are used to model the propagation of light waves through nonlinear optical fibers. A brief history of nonlinear bistable optical resonators is discussed and the simple fiber ring resonator is analyzed in particular. Chapter 16 is devoted to the study of these optical resonators, and there is discussion on phenomena such as bistability, chaotic attractors, feedback, hysteresis, instability, linear stability analysis, multistability, nonlinearity, and steady states. The first and second iterative methods are defined in this chapter. Some simple fractals may be constructed

using pencil and paper in Chapter 17, and the concept of fractal dimension is introduced. Fractals may be thought of as identical motifs repeated on ever-reduced scales. Unfortunately, most of the fractals appearing in nature are not homogeneous but are more heterogeneous, hence the need for the multifractal theory given later in the chapter. It has been found that the distribution of stars and galaxies in our universe is multifractal, and there is even evidence of multifractals in rainfall, stock markets, and heartbeat rhythms. Applications in geoscience, materials science, microbiology, and image processing are briefly discussed. Chapter 18 provides a brief introduction to image processing which is being used more and more by a diverse range of scientific disciplines, especially medical imaging. The fast Fourier transform is introduced and has a wide range of applications throughout the realms of science.

Chapter 19 is devoted to the new and exciting theory behind chaos control and synchronization. For most systems, the maxim used by engineers in the past has been “stability good, chaos bad,” but more and more nowadays this is being replaced with “stability good, chaos better.” There are exciting and novel applications in cardiology, communications, engineering, laser technology, and space research, for example. A brief introduction to the enticing field of neural networks is presented in Chapter 20. Imagine trying to make a computer mimic the human brain. One could ask the question: In the future will it be possible for computers to think and even be conscious? The human brain will always be more powerful than traditional, sequential, logic-based digital computers and scientists are trying to incorporate some features of the brain into modern computing. Neural networks perform through learning and no underlying equations are required. Mathematicians and computer scientists are attempting to mimic the way neurons work together via synapses; indeed, a neural network can be thought of as a crude multidimensional model of the human brain. The expectations are high for future applications in a broad range of disciplines. Neural networks are already being used in machine learning and pattern recognition (computer vision, credit card fraud, prediction and forecasting, disease recognition, facial and speech recognition), the consumer home entertainment market, psychological profiling, predicting wave over-topping events, and control problems, for example. They also provide a parallel architecture allowing for very fast computational and response times. In recent years, the disciplines of neural networks and nonlinear dynamics have increasingly coalesced and a new branch of science called neurodynamics is emerging. Lyapunov functions can be used to determine the stability of certain types of neural network. There is also evidence of chaos, feedback, nonlinearity, periodicity, and chaos synchronization in the brain.

Chapter 21 focuses on binary oscillator computing, the subject of UK, International, and Taiwanese patents. The author and his co-inventor, Jon

Borresen, came up with the idea when modeling connected biological neurons. Binary oscillator technology can be applied to the design of arithmetic logic units (ALUs), memory, and other basic computing components. It has the potential to provide revolutionary computational speed-up, energy saving, and novel applications and may be applicable to a variety of technological paradigms including biological neurons, complementary metal-oxide-semiconductor (CMOS), memristors, optical oscillators, and superconducting materials. The research has the potential for MMU and industrial partners to develop super fast, low-power computers and may provide an assay for neuronal degradation for brain malfunctions such as Alzheimer's, epilepsy, and Parkinson's disease!

Examples of coursework and three examination-type papers are listed in Chapter 22, and a complete set of solutions for the book is listed in Chapter 23.

Both textbooks and research papers are presented in the list of references. The textbooks can be used to gain more background material, and the research papers have been given to encourage further reading and independent study.

This book is informed by the research interests of the author, which are currently nonlinear ordinary differential equations, nonlinear optics, multifractals, neural networks, and binary oscillator computing. Some references include recently published research articles by the author along with two patents.

The prerequisites for studying dynamical systems using this book are undergraduate courses in linear algebra, real and complex analysis, calculus, and ordinary differential equations; a knowledge of a computer language such as Basic, C, or Fortran would be beneficial but not essential.

Recommended Textbooks

- [1] H.P. Langtangen and A. Logg, *Solving PDEs in Python: The FEniCS Tutorial I* (Simula SpringerBriefs on Computing), Springer, New York, 2017.
- [2] B. Bhattacharya and M. Majumdar, *Random Dynamical Systems in Finance*, Chapman & Hall/CRC, New York, 2016.
- [3] L.C. de Barros, R.C. Bassanezi and W.A. Lodwick, *A First Course in Fuzzy Logic, Fuzzy Dynamical Systems, and Biomathematics: Theory and Applications*, Springer, New York, 2016.
- [4] V. Volterra, *Theory of Functionals and of Integral and Integro-Differential Equations*, Dover Publications, New York, 2005.

[5] J. Mallet-Paret (Editor), J. Wu (Editor), H. Zhu (Editor), Y. Yi (Editor), *Infinite Dimensional Dynamical Systems (Fields Institute Communications)*, Springer, New York, 2013.

[6] C. Bernido, M.V. Carpio-Bernido, M. Grothaus et al., *Stochastic and Infinite Dimensional Analysis*, Birkhäuser, New York, 2016.

Special thanks go to Ben Nuttall (Python guru), Community Manager, the Raspberry Pi Foundation, Cambridge, UK (www.raspberrypi.org), for reviewing this book. I would also like to express my sincere thanks to all of the reviewers of this book and the other editions of my books. As always, thanks also go to Birkhäuser and Springer, especially Samuel DiBella (Assistant Editor, Springer Nature). Finally, thanks to my family and especially my wife Gaynor, and our children, Sebastian and Thalia, for their continuing love, inspiration, and support.

Manchester, UK

Stephen Lynch FIMA SFHEA

Contents

1	A Tutorial Introduction to Python	1
1.1	The IDLE Integrated Development Environment for Python	2
1.1.1	Tutorial One: Using Python as a Powerful Calculator	4
1.1.2	Tutorial Two: Simple Programming with Python . . .	6
1.1.3	Tutorial Three: Simple Plotting Using the Turtle Module	9
1.2	Anaconda, Spyder and the Libraries, SymPy, Numpy, and Matplotlib	14
1.2.1	Tutorial One: A Tutorial Introduction to SymPy . . .	15
1.2.2	Tutorial Two: A Tutorial Introduction to Numpy and Matplotlib	18
1.2.3	Tutorial Three: Simple Programming, Solving ODEs, and More Detailed Plots	20
1.3	Exercises	27
2	Differential Equations	33
2.1	Simple Differential Equations and Applications	34
2.1.1	Linear Differential Equations	34
2.1.2	Separable Differential Equations	35
2.1.3	Exact Differential Equations	39
2.1.4	Homogeneous Differential Equations	40
2.2	Applications to Chemical Kinetics	44
2.3	Applications to Electric Circuits	48
2.4	Existence and Uniqueness Theorem	53
2.5	Python Programs	57
2.6	Exercises	59
3	Planar Systems	65
3.1	Canonical Forms	66
3.1.1	Real Distinct Eigenvalues	68
3.1.2	Complex Eigenvalues ($\lambda = \alpha \pm i\beta$)	69
3.1.3	Repeated Real Eigenvalues	70

3.2	Eigenvectors Defining Stable and Unstable Manifolds	72
3.3	Phase Portraits of Linear Systems in the Plane	74
3.4	Linearization and Hartman’s Theorem	78
3.5	Constructing Phase Plane Diagrams	79
3.6	Python Programs	87
3.7	Exercises	90
4	Interacting Species	95
4.1	Competing Species	96
4.2	Predator-Prey Models	99
4.3	Other Characteristics Affecting Interacting Species	104
4.4	Python Programs	107
4.5	Exercises	108
5	Limit Cycles	113
5.1	Historical Background	114
5.2	Existence and Uniqueness of Limit Cycles in the Plane	117
5.3	Nonexistence of Limit Cycles in the Plane	123
5.4	Perturbation Methods	127
5.5	Python Programs	136
5.6	Exercises	139
6	Hamiltonian Systems, Lyapunov Functions, and Stability	145
6.1	Hamiltonian Systems in the Plane	146
6.2	Lyapunov Functions and Stability	151
6.3	Python Programs	157
6.4	Exercises	158
7	Bifurcation Theory	163
7.1	Bifurcations of Nonlinear Systems in the Plane	164
7.1.1	A Saddle-Node Bifurcation	165
7.1.2	A Transcritical Bifurcation	167
7.1.3	A Pitchfork Bifurcation	167
7.1.4	A Hopf Bifurcation	170
7.2	Normal Forms	170
7.3	Multistability and Bistability	174
7.4	Python Programs	178
7.5	Exercises	180
8	Three-Dimensional Autonomous Systems and Chaos	185
8.1	Linear Systems and Canonical Forms	186
8.2	Nonlinear Systems and Stability	190
8.3	The Rössler System and Chaos	194

8.3.1	The Rössler Attractor	194
8.3.2	Chaos	196
8.4	The Lorenz Equations, Chua’s Circuit, and the Belousov-Zhabotinski Reaction	199
8.4.1	The Lorenz Equations	199
8.4.2	Chua’s Circuit	201
8.4.3	The Belousov-Zhabotinski (BZ) Reaction	204
8.5	Python Programs	207
8.6	Exercises	211
9	Poincaré Maps and Nonautonomous Systems in the Plane	215
9.1	Poincaré Maps	216
9.2	Hamiltonian Systems with Two Degrees of Freedom	221
9.3	Nonautonomous Systems in the Plane	227
9.4	Python Programs	235
9.5	Exercises	239
10	Local and Global Bifurcations	245
10.1	Small-Amplitude Limit Cycle Bifurcations	246
10.2	Gröbner Bases	252
10.3	Melnikov Integrals and Bifurcating Limit Cycles from a Center	258
10.4	Bifurcations Involving Homoclinic Loops	260
10.5	Python Programs	262
10.6	Exercises	266
11	The Second Part of Hilbert’s Sixteenth Problem	271
11.1	Statement of Problem and Main Results	272
11.2	Poincaré Compactification	275
11.3	Global Results for Liénard Systems	281
11.4	Local Results for Liénard Systems	290
11.5	Python Programs	291
11.6	Exercises	292
12	Delay Differential Equations	297
12.1	Introduction and the Method of Steps	298
12.2	Applications in Biology	304
12.3	Applications in Nonlinear Optics	310
12.4	Other Applications	313
12.5	Python Programs	315
12.6	Exercises	320

13 Linear Discrete Dynamical Systems	327
13.1 Recurrence Relations	328
13.2 The Leslie Model	333
13.3 Harvesting and Culling Policies	337
13.4 Python Programs	342
13.5 Exercises	343
14 Nonlinear Discrete Dynamical Systems	347
14.1 The Tent Map and Graphical Iterations	348
14.2 Fixed Points and Periodic Orbits	353
14.3 The Logistic Map, Bifurcation Diagram, and Feigenbaum Number	360
14.4 Gaussian and Hénon Maps	368
14.5 Applications	373
14.6 Python Programs	376
14.7 Exercises	380
15 Complex Iterative Maps	385
15.1 Julia Sets and the Mandelbrot Set	386
15.2 Boundaries of Periodic Orbits	391
15.3 The Newton Fractal	395
15.4 Python Programs	396
15.5 Exercises	399
16 Electromagnetic Waves and Optical Resonators	403
16.1 Maxwell's Equations and Electromagnetic Waves	404
16.2 Historical Background	406
16.3 The Nonlinear SFR Resonator	412
16.4 Chaotic Attractors and Bistability	413
16.5 Linear Stability Analysis	417
16.6 Instabilities and Bistability	420
16.7 Python Programs	424
16.8 Exercises	428
17 Fractals and Multifractals	433
17.1 Construction of Simple Examples	434
17.2 Calculating Fractal Dimensions	441
17.3 A Multifractal Formalism	448
17.4 Multifractals in the Real World and Some Simple Examples	452
17.5 Python Programs	459
17.6 Exercises	464

18 Image Processing with Python	471
18.1 Image Processing and Matrices	472
18.2 The Fast Fourier Transform	477
18.3 The Fast Fourier Transform on Images	484
18.4 Exercises	487
19 Chaos Control and Synchronization	491
19.1 Historical Background	492
19.2 Controlling Chaos in the Logistic Map	497
19.3 Controlling Chaos in the Hénon Map	498
19.4 Chaos Synchronization	505
19.5 Python Programs	509
19.6 Exercises	513
20 Neural Networks	519
20.1 Introduction	520
20.2 The Delta Learning Rule and Backpropagation	526
20.3 The Hopfield Network and Lyapunov Stability	531
20.4 Neurodynamics	541
20.5 Python Programs	545
20.6 Exercises	550
21 Binary Oscillator Computing	557
21.1 Brain Inspired Computing	558
21.2 Oscillatory Threshold Logic	563
21.3 Applications and Future Work	571
21.4 An Assay for Neuronal Degradation	577
21.5 Python Programs	579
21.6 Exercises	584
22 Coursework and Examination-Type Questions	591
22.1 Examples of Coursework Questions	592
22.2 Examination 1	607
22.3 Examination 2	610
22.4 Examination 3	613
23 Solutions to Exercises	619
23.1 Chapter 1	619
23.2 Chapter 2	622
23.3 Chapter 3	623
23.4 Chapter 4	625
23.5 Chapter 5	626
23.6 Chapter 6	628
23.7 Chapter 7	628

23.8	Chapter 8	630
23.9	Chapter 9	631
23.10	Chapter 10	631
23.11	Chapter 11	632
23.12	Chapter 12	634
23.13	Chapter 13	634
23.14	Chapter 14	636
23.15	Chapter 15	637
23.16	Chapter 16	638
23.17	Chapter 17	638
23.18	Chapter 18	639
23.19	Chapter 19	639
23.20	Chapter 20	640
23.21	Chapter 21	640
23.22	Chapter 22	641
Appendix A Index of Python Programs		645
A.1	IDLE Python Programs	645
A.2	Anaconda Python Programs	646
Index		651

Appendix A

Index of Python Programs

Readers can download the Python program files via GitHub:

<https://github.com/springer-math/dynamical-systems-with-applications-using-python>

These files will be kept up-to-date and extra files will be added in the forthcoming years.

A.1 IDLE Python Programs

These files include solutions to the Exercises listed in Chapter 1.

euclid_algorithm.py --- See Exercise 10.
F2C.py --- See Exercise 1(a).
F2K.py --- Converts degrees Fahrenheit to Kelvin.
fibonacci.py --- Lists first n terms of the Fibonacci sequence.
fmu.py --- The logistic function.
fractal_tree.py --- Plots a fractal tree.
fractal_tree_color.py --- Plots a color fractal tree.
grade.py --- Converts a score to a grade.
guess_number.py --- Guess the number game.
koch_snowflake.py --- See Exercise 1(d).
koch_square.py --- Plots a Koch square fractal.
Pythag_Triples.py --- See Exercise 1(c).
sierpinski.py --- Plots a Sierpinski triangle fractal.
sierpinski_square.py --- Plots a Sierpinski square fractal.

sum_primes.py --- See Exercise 1(b).
 sum_n.py --- Sums the natural numbers to n.

A.2 Anaconda Python Programs

If you have difficulty with the animation programs in Spyder, you have to change the backend to run an animation in the IPython console. You can do that by running

```
In[1]: %matplotlib qt5
```

before the animation. If you don't want to use this command every time, you can go to: Tools, Preferences, IPython Console, Graphics, Backend, and change it from "Inline" to "Automatic."

```
Program_01a.py --- Solve a simple ODE.
Program_01b.py --- Solve a second order ODE.
Program_01c.py --- Plot two curves on one graph.
Program_01d.py --- Subplots.
Program_01e.py --- Surface and contour plot in 3D.
Program_01f.py --- A parametric curve in 3D.
Program_01g.py --- Animation of a simple curve.

Program_02a.py --- Solve a separable ODE.
Program_02b.py --- Solve the logistic ODE.
Program_02c.py --- Power series solution.
Program_02d.py --- Power series solution for van der Pol.
Program_02e.py --- Plot series solution against numerical
                  solution.
Program_02f.py --- Solve a linear first order ODE.
Program_02g.py --- Solve a linear second order ODE.

Program_03a.py --- Plot the phase portrait of a linear system.
Program_03b.py --- Plot the phase portrait of a nonlinear
                  system.
Program_03c.py --- Finding critical points.

Program_04a.py --- Phase portrait and time series of Holling-
                  Tanner model.
```

Program_05a.py --- Limit cycle of a Fitzhugh-Nagumo system.
Program_05b.py --- Approximate and numerical solutions to ODEs.
Program_05c.py --- Error between one-term and numerical
solution.
Program_05d.py --- Lindstedt-Poincare technique.

Program_06a.py --- Contour plot.
Program_06b.py --- Surface plot.

Program_07a.py --- Animation of a simple curve.
Program_07b.py --- Animation of a subcritical Hopf bifurcation.
Program_07c.py --- Animation of a SNIC bifurcation.

Program_08a.py --- The Rossler attractor.
Program_08b.py --- The Lorenz Attractor.
Program_08c.py --- The Belousov-Zhabotinsky reaction.
Program_08d.py --- Animation of a Chua circuit bifurcation.

Program_09a.py --- Simple Poincare return map.
Program_09b.py --- Hamiltonian with two degrees of freedom plot.
Program_09c.py --- Phase portrait and Poincare map for the
Duffing system.
Program_09d.py --- Bifurcation diagram of Duffing equation.

Program_10a.py --- Computing Lyapunov quantities.
Program_10b.py --- Division algorithm for multivariate
polynomials.
Program_10c.py --- S-polynomial.
Program_10d.py --- Computing the Groebner basis.
Program_10e.py --- Computing Groebner basis of Lyapunov
quantities.
Program_10f.py --- Animation of a homoclinic limit cycle
bifurcation.
Program_10g.py --- Animation of a homoclinic limit cycle
bifurcation.

Program_11a.py --- Animation of a Lienard limit cycle.

Program_12a.py --- The method of steps.
Program_12b.py --- Plot of solution by method of steps.
Program_12c.py --- The Mackey-Glass DDE.
Program_12d.py --- The Lang-Kobayashi DDEs.

Program_13a.py --- Computing bank interest.
Program_13b.py --- Solving a second order recurrence relation.
Program_13c.py --- The Leslie matrix, eigenvalues and
eigenvectors.

Program_14a.py --- Graphical iteration of the tent map.
Program_14b.py --- Bifurcation diagram of the logistic map.
Program_14c.py --- Computing Lyapunov exponents for the
logistic map.
Program_14d.py --- Iteration of the Henon map.
Program_14e.py --- Lyapunov exponents of the Henon map.

Program_15a.py --- Point plot for a Julia set.
Program_15b.py --- Colormap of a Julia set.
Program_15c.py --- Color Mandelbrot set.
Program_15d.py --- Color Newton fractal Julia set.

Program_16a.py --- Intersection of implicit curves.
Program_16b.py --- Chaotic Attractor of the Ikeda map
Program_16c.py --- Bifurcation diagram of the Ikeda map.

Program_17a.py --- The Koch curve.
Program_17b.py --- Chaos game and the Sierpinski triangle.
Program_17c.py --- Barnsley's fern.
Program_17d.py --- Subplots of τ , D_q and $f(\alpha)$
multifractal spectra.

Program_18a.py --- Generating a multifractal image
Program_18b.py --- Counting pixels in a color image.
Program_18c.py --- Image and statistical analysis on
microbes.png
Program_18d.py --- Fast Fourier transform of a noisy signal
Program_18e.py --- Iterative map and power spectra
Program_18f.py --- Fast Fourier transform of Lena image
Program_18g.py --- Edge detection in Lena image

Program_19a.py --- Chaos control in the logistic map.
Program_19b.py --- Chaos control in the Henon map.
Program_19c.py --- Chaos synchronization between two Lorenz
systems.
Program_19d.py --- Generalized synchronization.

Program_20a.py --- The generalized delta learning rule.

Program_20b.py --- The discrete Hopfield network.
Program_20c.py --- Iteration of a minimal chaotic neuromodule.
Program_20d.py --- Bifurcation diagram of neuromodule.

Program_21a.py --- The Hodgkin-Huxley equations.
Program_21b.py --- The Fitzhugh-Nagumo half-adder.
Program_21c.py --- Phase portrait Josephson junction limit
 cycle.
Program_21d.py --- Animated Josephson junction limit cycle.
Program_21e.py --- Pinched hysteresis of a memristor.

Index

- absorptive nonlinearity, 407
- action potential, 114, 559
- activation
 - function, 521, 550
 - level, 532
 - potential, 522
- ADALINE network, 524
- affine linear transformation, 440
- age class, 106, 333
- Airy equation, 61
- algebraicity of limit cycles, 283
- All or None principle, 560
- Alzheimer's disease, 577
- ampere, 48
- Ampere's law, 405
- Anaconda, 14
- Anaconda Python programs, 646
- anemia, 374
- angiogenesis, 308
- angular frequency of the wave,
406
- animation, 26
 - Spyder, 646
- ants and termites, 96
- aperiodic, 201, 360
 - behavior, 196
- append, 262
- applying a damper, 493
- arrhythmic, 495
- Artificial Intelligence Group, 526
- artificial neural networks, 520
- ArtistAnimation, 178
- assay for neuronal degradation,
577
- associative memory, 524, 531
- asymptotic expansion, 127
- asymptotically stable
 - critical point, 152
- asynchronous updating, 536
- attractor, 193
- attributes, 529
- autocatalysis, 204
- autonomous differential equation,
54
- autonomous system, 186
- auxiliary system approach, 507
- average Lyapunov exponent, 366
- ax.set title, 235
- Axes3D(fig), 235
- axial flow compressors, 175
- axon, 521, 558
- Baby computer, 571
- backpropagation, 526
 - algorithm, 528
- backward training, 388
- ballistic propagation, 571
- bandwidth, 407
- Barnsley's fern, 441
- basin of attraction, 159, 193, 372,
388
- basis, 253

- batch data, 530
- Belousov-Zhabotinski reaction, 204, 212
- Bendixson's criterion, 123
- bias, 521
- bifurcating limit cycles from a center, 258
- bifurcation
 - curve, 167
 - diagram, 166
 - at infinity, 273
 - point, 362
 - value, 164
- bifurcation diagram
 - CR resonator, 409
 - DDE
 - field components, 313
 - Mackey-Glass, 307
 - Duffing equation, 232
 - Gaussian map, 370
 - Ikeda map, 425
 - Josephson junction, 575
 - limit cycles, 177
 - logistic map, 365
 - neuromodule, 544
 - periodically forced
 - pendulum, 233
 - SFR resonator, 422
 - SNIC, 178
- binarize, 475
- binary half adder, 565
- biology, 374
- bipolar activation function, 522
- bistability, 174, 175, 368, 421
- bistable, 175, 206, 370, 408
 - cycle, 203
 - device, 407, 421
 - neuromodule, 543
 - optical resonator, 495
 - region, 233, 241, 409, 414
 - solution, 176
- bistable region, 544
- blit, 178
- blowflies, 360
- bluegill sunfish, 104
- Boston housing data, 529
- boundaries of periodic orbits, 391
- box-counting dimension, 443, 448
- brain functions, 521
- Briggs-Rauscher, 204
- bursting, 310
- butterfly effect, 200
- BZ reaction, 204
- canny edge detector, 477
- canonical form, 66, 67, 186
- Cantor
 - multifractal, 454
 - set, 235, 435
- capacitance, 49
- capacitor, 49
- cardioid, 393
- cardiology, 492
- carrying capacity, 38
- cavity ring (CR) resonator, 408
- cavity round-trip time, 408
- cell body, 558
- center, 69, 148, 246
 - manifold
 - theorem, 191
- changing the system parameters, 493
- chaologist, 492
- chaos, 194, 196, 350, 353, 409
 - control, 541, 552
 - synchronization, 505
- chaos control
 - OGY method, 493
 - periodic proportional pulses, 513
- chaos game, 439
- chaotic
 - attractor, 197, 229, 415
 - dynamics, 199
 - phenomena, 348

- chaotic attractor
 - Hénon map, 501
 - neuromodule, 542
 - Sierpiński, 440
- Chapman cycle, 211
- characteristic
 - equation, 302, 330
 - exponent, 283
 - multiplier, 219
- charge density, 405
- chemical
 - kinetics, 44, 86, 117, 204, 505, 561
 - reaction, 59
 - signals, 521
 - substance, 61
- chemical law of mass action, 44
- Chua's circuit, 51, 117, 201, 493, 514
- circle map, 221
- circular frequency of light, 410
- classical symmetry argument, 248
- classification of critical points, 71
- climate change, 314
- clipping problem, 451
- clockwise bistable cycle, 632
- clockwise hysteresis, 241, 410
- cluster, 524
- cmap, 396
- CMOS oscillators, 571
- coarse Hölder exponent, 450
- codimension-1 bifurcation, 182
- codimension-2 bifurcation, 182
- coexistence, 96
- coexisting chaotic attractors, 372
- col, 69
- collect, 128
- comb(k,s), 459
- common typing errors, 20
- commutative ring, 252
- competing species, 96, 110, 140
- complete synchronization, 506
- completely
 - integrable, 221
 - reduced, 254
- complex eigenvalues, 69, 330
- complex iterative equation, 417
- complex(x,y), 396
- compound interest, 329
- computer algebra, 252
- concentrations, 44
- conditional Lyapunov exponents, 506
- conductivity, 406
- conformal mapping, 386
- conservation of energy, 146
- conservation of mass, 86
- conservative, 146
- contact rate, 212
- content-addressable memory, 532
- continuation lines, 20
- continuous Hopfield model, 532
- contour plot, 24
- control curves, 499
- control engineering, 526
- control parameter, 495
- control region, 495
- controlling chaos
 - Hénon map, 498
 - logistic map, 497
- conversational agents, 526
- convex closed curve, 123
- convoluted surfaces, 191
- coordinates of an image, 474
- core area of the fiber, 413
- corollary to Poincaré-Bendixson theorem, 120
- correlation dimension, 450
- coulomb, 48
- counterclockwise hysteresis, 410
- coupler, 412
- critical point, 55, 72, 186, 191, 300
 - at infinity, 276

- culling, 106
 - policy, 337
- current, 48
 - density, 405
- cuspl, 84
- cylindrical polar coordinates, 193

- damping, 116
- damping coefficient, 282
- dangerous bifurcation, 174
- Daphnia dentifera*, 104
- dashed curve, 22
- data, 545
- data mining, 520
- databases, 529
- DDE, 298
- dde23, 315
- ddeint, 298
- def, 6
- defibrillator, 495
- defraction, 61
- degenerate
 - critical point, 148
 - node, 70
- degree, 273
- degree lexicographical order, 253
- delay differential equation, 298
- deleted neighborhood, 121
- delta learning rule, 524, 526
- dendrites, 521, 558
- depolarization, 559
- depolarize, 566
- derivative of the Poincaré map
 - test, 220
- desired vector, 527
- deterministic chaos, 195, 492
- deterministic system, 520
- D_f , 441
- dielectric, 406
- difference equation, 328, 551
- differential amplifier, 407
- diffusion limited aggregates (DLA), 453

- dimension, 448
- direction
 - field, 66
 - vector, 66
- discrete Fourier transform, 479
- discrete Hopfield model, 536
- dispersive nonlinearity, 407
- displacement function, 247
- distributive laws, 252
- divergence test, 248
- domain of stability, 97, 193, 388
- double-coupler fiber ring
 - resonator, 410, 428
- double Hopf bifurcation, 308, 322
- double-scroll attractor, 203
- double-well potential, 151
- D_q , 448
- driver system, 506, 507
- dsolve, 57
- Duffing
 - equation, 129, 227
 - system, 493, 611
- Dulac's criterion, 122
- Dulac's theorem, 273

- E_C , 191
- economic model, 117, 322, 332
- economics, 92, 374, 382, 453, 505
- edge detection
 - Roberts, 485
 - Sobel, 485
- eig, 342
- eigenvector, 72
- El Niño, 313
- electric
 - circuit, 48, 92, 117, 201, 532
 - displacement, 405
 - displacement vector, 405
 - field, 412, 417, 495
 - field strength, 405
 - flux density, 405
- electromotive force (EMF), 49
- elementary steps, 44

- elliptic integral, 259
- EMF, 49
- energy level, 227
- enrichment of prey, 106
- ENSO model, 313
- Enthought Canopy, 14
- environmental effects, 106
- environmental model, 313
- epidemic, 61, 85, 106, 117, 212
- epilepsy, 577
- epoch, 524
- equilibrium point, 55
- ergodicity, 366, 495
- error backpropagation rule, 528
- error function, 527
- erythrocytes, 374
- E_S , 72, 186, 191
- E_U , 72, 186, 191
- Euclidean dimension, 448
- Euclid's algorithm, 29
- exact, 39
- exact differential equation, 39
- excitatory, 521, 558
- existence and uniqueness
 - limit cycle, 117
- existence theorem, 53
- extinct, 96

- Fabry-Perot
 - interferometer, 407
 - resonator, 407
- farad, 49
- Faraday's law, 49
 - of induction, 404
- fast Fourier transform, 480
- feedback, 175, 407, 420
- feedback mechanism, 543
- feedforward single layer network, 523
- Feigenbaum constant, 365
- Ferranti Mark 1, 571
- FFT, 480
- fiber parameters, 423

- Fibonacci sequence, 343
- field, 253
- figsize, 107
- fine focus, 246
- first integral, 146
- first iterative method, 419, 422, 552
- first return map, 216
- first-order difference equation, 328
- fish population, 38, 181, 345
- Fitzhugh-Nagumo
 - equations, 116
 - oscillator, 114
 - system, 563
- fixed point, 55, 355
 - period m , 221
 - period N , 357
 - period one, 217, 414, 497
 - period two, 498
- fixed size box-counting algorithm, 451
- fixed weight box-counting algorithm, 451
- flow, 118
- focal values, 247
- fold bifurcation, 182
- for loop, 6
- forced system, 227
- forward rate constant, 46
- fossil dating, 59
- Fourier spectrum, 478
- Fourier transform, 477
- fractal, 434, 441
 - attractor, 197, 441
 - dimension, 441
 - Cantor set, 442
 - Koch curve, 442
 - Koch square, 442
 - Sierpiński triangle, 442
 - geometry, 434
 - structure, 196, 201, 386
- fragmentation ratios, 449

- $f(\alpha)$ spectrum, 448
- FuncAnimation, 178, 207
- Function, 57
- function approximators, 526
- fundamental memory, 536
- fuzzy discs, 451
- Gauss's law
 - electricity, 405
 - magnetism, 405
- Gauss-Newton method, 528
- Gaussian input pulse, 421
- Gaussian map, 368
- Gaussian pulse, 638
- generalized delta rule, 528
- generalized fractal dimensions, 448
- generalized mixed Rayleigh
 - Liénard equations, 267
- generalized synchronization, 507
- gestation period, 304
- GitHub, 2
- global bifurcation, 260, 273
- global warming, 314
- globally asymptotically stable, 194, 211
- glucose in blood, 60
- Gröbner bases, 252
- gradient, 66
- gradient vector, 527
- graphene nano-ribbon, 573
- graphic, 120
- graphical method, 351, 417
- gray scale, 458, 473
- Green's theorem, 122
- Gross National Product (GNP), 374
- Guido van Rossum, 2
- Hénon-Heiles Hamiltonian, 225
- haematopoiesis, 306
- Hamiltonian, 146, 611
- Hamiltonian systems
 - with two degrees of freedom, 221
- handcrafted patterns, 541
- hard bifurcation, 174
- Hartman's theorem, 79
- harvesting, 106, 181
 - policy, 337
- Hausdorff dimension, 448
- Hausdorff index, 441
- Hausdorff-Besicovich dimension, 450
- Heaviside function, 523
- Hebb's learning law, 522
- Hebb's postulate of learning, 536
- help command, 20
- Hénon map, 370, 445, 451, 609, 612
- henry, 49
- heteroclinic
 - bifurcation, 234
 - orbit, 120, 150, 234, 274
 - tangle, 234
- heterogeneous, 448
- hidden layer, 524, 528, 603
- high pass filter, 484
- Hilbert numbers, 273
- Hints for programming, 20
- history, 176
- history function, DDEs, 298
- Hodgkin-Huxley equations, 114, 560
- Holling-Tanner model, 101, 139, 164
- homoclinic
 - bifurcation, 200, 234, 261
 - loop, 260, 261, 267
 - orbit, 150, 234, 274
 - tangle, 234
- homogeneous, 448
- homogeneous differential equation, 40

- Hopf
 - bifurcation, 170, 182, 303
 - singularity, 182
- Hopfield network, 156, 531, 551, 610, 613
- Hopfield neural network, 525
- horseshoe dynamics, 235
- host-parasite system, 104
- human population, 85, 343
- hyperbolic
 - attracting, 283
 - critical point, 79
 - fixed point, 220, 371
 - iterated function system, 440
 - repelling, 283
 - stable limit cycle, 220
 - unstable limit cycle, 220
- hyperpolarize, 566
- hyperpolarized, 559
- hysteresis, 175, 371, 421
 - curves, 313
 - Josephson junction, 574

- IDE, 14
- ideal, 252
- IDLE, 2
- IDLE Python programs, 645
- if, elif, else, 6
- Ikeda
 - DDE, 310
 - map, 375, 414, 428, 513
- im, 396
- image analysis, 453
- image compression, 434, 484
- incident, 407
- indentation level, 20
- index, 125
- inductance, 49
- infected population, 105
- infectives, 85
- inflation unemployment model, 382
- infodict, 107
- information dimension, 450
- inhibitory, 521, 558
- initial value problem, 38
- input vector, 521
- insect population, 109, 345, 616
- instability, 421
- instant physician, 526
- integrable, 223
- integrate and fire neuron, 116
- Integrated Development Environment, 2, 14
- integrating factor, 34
- intensity, 412
- interacting species, 95, 626
- intermittency, 203, 212, 363
 - route to chaos, 365
- invariant, 118, 201, 416
 - axes, 82, 98
- inverse discrete Fourier transform, 479
- inverted Koch snowflake, 609
- inverted Koch square, 438
- inverted pendulum, 322
- io.imsave, 474
- isoclines, 67
- isolated periodic solution, 114
- isothermal chemical reaction, 86
- iterated function system (IFS), 440
- iteration, 328

- Jacobian, 171
- Jacobian matrix, 78, 192, 205, 371, 502
- Jaynes-Cummings model, 595
- Jordan curve, 121, 286
- Josephson junction
 - mathematical model, 573
- Josephson junction (JJ), 571
- jth point of period i , 357
- Julia set, 386, 389, 434, 612
 - color, 390

- KAM
 - theorem, 227
 - tori, 227
- kernel machines, 525
- Kerr
 - effect, 407, 413
 - type, 413
- kinetic energy, 146
- Kirchhoff's
 - current law, 49
 - laws, 532
 - voltage law, 49
- Koch
 - curve, 436
 - snowflake, 464
 - square, 436

- ladybirds and aphids, 99
- lambdas, 315
- laminarize, 494
- Lang-Kobayashi equations, 310
- Laplace transform, 50
- large-amplitude limit cycle, 175
 - bifurcation, 175
- laser, 182, 374, 410, 494
 - model, 310
- LaTeX, 24
- law of mass action, 86
- learning process, 521
- learning rate, 527
- least mean squared (LMS)
 - algorithm, 524
- legend, 23, 107
- Legendre transformation, 450
- Leslie
 - matrix, 334
 - model, 333
- lexicographical order, 253
- lie detector, 526
- Liénard
 - equation, 248
 - plane, 282

- system, 116, 123, 139, 260, 281
 - large parameter, 286
 - local results, 290
 - theorem, 293
- limit cycle, 106, 114, 118, 206, 595, 611
 - hyperbolic, 259
 - neuron, 116
 - nonexistence, 608
 - 3-D, 195
- Lindstedt-Poincaré technique, 131
- linear differential equation, 34
- linear phase shift, 412, 423
- linear stability analysis, 56, 300, 417
- linear transformation, 187
- linearization, 78
- linearized system, 78
- Lipschitz
 - condition, 53
 - continuous, 53
- local bifurcation, 273
- log-log plot, 445
- logic gates, 407
- logic operations, 565
- logistic
 - equation, 38, 302
 - function, 6
 - growth, 101
 - map, 360, 497, 612
- Lorenz
 - attractor, 201
 - equations, 199, 494
- loss in the fiber, 412
- Lotka-Volterra model, 99, 164, 212, 304
- low-gain saturation function, 523
- low pass filter, 484
- lowest common multiple, 256
- Lyapunov
 - quantity, 247
 - stability, 531

- Lyapunov domain of stability, 155
- Lyapunov exponent, 197, 366, 612
 - Lorenz system, 603
- Lyapunov function, 151, 154, 194, 247, 284, 550, 607
 - Hopfield network, 532
- Lyapunov quantities, 290
- Lyapunov stability theorem, 152
- lynx and snowshoe hares, 99

- Mac OS, 2
- Mackey-Glass model, 306
- magnetic field vector, 405
- magnetic flux, 405
- magnetostrictive ribbon, 494
- Mandelbrot, 443
- Mandelbrot set, 389, 391, 434
- manifold, 72
- Maple, 26
- math module, 4
- Mathematica, 26
- MATLAB, 26
- MATLAB code to Python, 603
- matplotlib, 18
- maximal interval of existence, 54, 62, 118
- Maxwell's equations, 404
- Maxwell-Bloch equations, 406
- Maxwell-Debye equations, 406
- McCulloch-Pitts neuron, 522
- MEA, 577
- mean, 545
 - infectivity period, 212
 - latency period, 212
- mechanical oscillator, 91
 - DDE, 321
- mechanical system, 117, 176
- Melnikov
 - function, 259
 - integral, 258
- memory devices, 407
- memristance, 52
- memristor, 51, 572, 573
 - mathematical model, 574
- meshgrid, 157
- meteorology, 199
- method of multiple scales, 134
- method of steepest descent, 527
- method of steps, 299
- mgrid, 425
- micro-parasite—zooplankton—
 - fish system, 104
- minimal chaotic neuromodule, 542
- minimal Gröbner basis, 257
- mixed fundamental memories, 537
- mixing, 353
- modulo, 254
- monomial, 253
 - ordering, 253
- mortgage assessment, 526
- motif, 434
- mplot3d, 207
- multi-electrode array, 577
- multidegree, 254
- multifractal, 447, 472, 602
 - formalism, 448
 - Hénon map, 459
 - Sierpiński triangle, 459
 - spectra, 448
- multistability, 174
- multistable, 151, 175, 206, 241, 543
- murder, 60
- muscle model, 60
- mutual exclusion, 96
- myimages, 292
- mylein sheath, 558

- national income, 332
- negative
 - limit set, 118
 - semiorbit, 118
- negatively, invariant, 118
- net reproduction rate, 339

- network architecture, 521
- neural network, 375, 505, 521
 - DDE, 315
- neuristor, 572
- neurodynamics, 541
- neurological assay, 577
- neuromodule, 541
- neuron
 - module, 375
- neuron(s), 114, 521, 551, 558
- neuronal model, 521
- neurotransmitters, 558
- Newton fractal, 395, 602
- Newton's law of cooling, 60
- Newton's law of motion, 146
- Newton's method, 395, 528
- noise, 497
- NOLM, 404
 - with feedback, 410
- nonautonomous system, 116, 227
- nonconvex closed curve, 124
- nondegenerate
 - critical point, 148, 246
- nondeterministic chaos, 195, 492
- nondeterministic system, 520
- nonexistence of limit cycles, 123
- nonhyperbolic
 - critical point, 79, 151, 607
 - fixed point, 371
- nonlinear
 - center, 247
 - optics, 374
 - phase shift, 412
 - refractive index coefficient, 413
- nonlinearity, 175, 407
- nonperiodic behavior, 196
- nonsimple canonical system, 67
- normal form, 164, 170
- normalized eigenvector, 340
- not robust, 101
- notebook, 14
- np.mgrid, 88
- nullclines, 67
- numerical solutions, 42
- numpy, 18
- occasional proportional feedback (OPF), 494
- ODE, 34
- odeint, 42, 57
- OGY method, 495
- ohm, 49
- Ohm's law, 48
- optical
 - bistability, 407
 - computer, 407
 - fiber, 410
 - fiber double ring, 410
 - memories, 407
 - oscillators, 572
 - resonator, 176
 - sensor, 408
- optimal sustainable, 340
- optogenetics, 578
- orbit, 66, 118
- ordinary differential equation, 34
- oscillation of a violin string, 114
- oscillatory threshold logic, 563
- output vector, 521
- ozone production, 211
- parasitic infection, 111
- Parkinson's disease, 577
- partial differential equations, 34
- partition function, 448
- Pascal's triangle, 464
- passive circuit, 51
- Peixoto's theorem in the plane, 164
- pendulum, 147, 159, 241
 - double, 593
- perceptron, 522
- perihelion, 593
- period, 259
 - bubblings, 368
 - limit cycle, 103, 119, 205

- undoublings, 368
- period-doubling, 203
- period-doubling bifurcations to
 - chaos, 363
- period-n cycle, 195
- period-one behavior, 350
- period-two, 196
 - behavior, 350
- period-three behavior, 351
- periodic
 - behavior, 115
 - orbit, 259
 - windows, 363
- periodicity, 348, 353
- permittivity of free space, 405
- perturbation methods, 127
- phase portrait, 66
- phase shift, 412
- physiology, 505
- piecewise, 300, 315
- piecewise linear function, 523
- pinched hysteresis, 52, 574
- pitchfork bifurcation, 167
- pixels, 451
- planar manifold, 187
- plastics, 453
- plt.axes, 292
- Poincaré
 - compactification, 275
 - map, 119, 199, 216, 259, 371, 495
 - section, 216, 611
- Poincaré-Bendixson theorem, 120, 227, 281, 287
- Poisson brackets, 223
- polar coordinates, 69, 276
- pole placement technique, 496
- pollution, 106
- polymer, 453
- population, 92
 - of rabbits, 86
- population model, 614
- positive
 - limit set, 118
 - semiorbit, 118
- positively, invariant, 118
- potato man, 394
- potential difference, 48
- potential energy, 146, 151
- potential function, 151
- pow(x,y), 315
- power, 412
 - law, 443
 - spectra, 199, 203
 - of a waterwheel, 92
- power-splitting ratio, 412
- pprint, 57
- Prandtl number, 200
- preallocate, 20
- predation, 104
 - rate, 101
- predator-prey, 117
 - DDE, 304
 - models, 99
 - system, 109
- probe vector, 536
- propagation, 412
- psychological profiling, 526
- PyDDE, 298
- pydelay, 298, 311
- Pyragas's method, 493
- Python
 - based exam, 607, 613
 - files download, 2
- qth moment, 448
- qualitative behavior, 66
- qualitatively equivalent, 74
- quasi-periodicity, 221
- quasi-polynomials, 302
- quasiperiodic, 544, 552
 - route to chaos, 203
- quasiperiodic forcing, 228
- quiver, 88

- Rössler
 - attractor, 194
 - system, 194
- radioactive decay, 610
- randint(a,b), 459
- random behavior, 195
- Raspberry Pi, 2
- rate constant, 45
- rate-determining step, 45
- Rational(1,2), 376
- rationaly independent, 221
- ravel, 157
- Rayleigh number, 200
- Rayleigh system, 115
- re, 396
- reaction rate equation, 45
- real distinct eigenvalues, 68
- recurrence relation, 328
- recurrent neural network, 524, 531
- red blood cells, 374
- red and grey squirrels, 96
- reduced, 262
- reduced Gröbner basis, 257
- reflected, 407
- refractive index, 407
- refractive nonlinearity, 407
- refuge, 106
- regionprops, 477
- regulator poles, 496
- relative permeabilities, 406
- relative permittivities, 406
- repeated real eigenvalues, 70
- repolarization, 559
- resistance, 49
- resonance terms, 173
- resonant, 173
- response system, 506, 508
- restoring coefficient, 282
- restoring force, 116
- restrictions in programming, 291
- return map, 247, 607
- reverse rate constant, 46
- reversed fundamental memories, 537
- RGB image, 474
- rgb2gray, 477
- ring, 252
- ringing, 241
- RLC circuit, 51, 117
- roach:fish population, 338
- robust, 103
- Rotating Wave Approximation (RWA), 312
- rsolve, 342
- rubbers, 453
- S-polynomial, 256
- saddle point, 69, 148
- saddle-node bifurcation, 165
- saddle-node on an invariant cycle bifurcation, 176
- safe bifurcation, 174
- save image, 472
- savefig, 22
- scaling, 443, 448
- scatter, 425
- sea lions and penguins, 96
- seasonal effects, 106
- seasonality, 212
- second iterative method, 420, 422, 552
- second order linear difference equation, 330
- second part of Hilbert's sixteenth problem, 272
- second-order differential equation, 50
- secular term, 131
- sedimentary rocks, 453
- self-similar, 448
- self-similar fractal, 441
- self-similarity, 434
- semistable
 - critical point, 56
 - limit cycle, 118, 220, 286

- sensitivity to initial conditions,
 - 196, 348, 353, 492
- separable differential equation, 35
- separation of variables, 35
- separatrix, 151
 - cycle, 261
- series solutions, 42
- SFR, 403
 - resonator, 409, 412
- sharks and fish, 99
- Sierpiński triangle, 439
- sigmoid function, 523
- signal processing, 453, 482
- simple canonical system, 68
- simple nonlinear pendulum, 146
- simply connected domain, 123
- singlet, 212
- singular node, 70
- Smale horseshoe map, 234, 373
- Smale-Birkhoff theorem, 235
- small perturbation, 56, 417
- small-amplitude limit cycle, 246
- soft bifurcation, 174
- solar system, 492
- solution curves, 36
- solve, 88, 258
- soma, 521, 558
- spatial vector, 405
- spectrum of Lyapunov exponents,
 - 197
- speed of light, 406
- spike train, 559
- spin-glass states, 537
- spirals, 355
- spurious steady state, 537
- SR flip-flop, 569
- stability, 151, 190
 - diagram, 420
- stable
 - critical point, 55, 152
 - fixed point, 361, 388
 - focus, 69
 - limit cycle, 103, 118
 - manifold, 72, 79, 186, 191,
 - 495
 - node, 69
- staircases, 356
- stationary point, 55
- std, 545
- steady state, 51, 103
- stem cell, 578
- stiff system, 47, 212
- stiffness, 116
- stochastic methods, 520
- stock market analysis, 453
- stoichiometric equations, 45
- Stokes's theorem, 405
- strange attractor, 197
- stretching and folding, 348
- strictly dominant, 336
- structurally
 - stable, 103, 164
 - unstable, 101, 164
- subcritical Hopf bifurcation, 174,
 - 200
- subharmonic oscillations, 229
- subplots, 22, 107
- summing junction, 521
- superconductor, 573
- supercritical Hopf bifurcation,
 - 174
- supervised learning, 524
- surface plot, 24
- susceptible population, 105
- susceptibles, 85
- sustainable, 338
- switches, 407
- symbols, 57
- sympy, 15
- synaptic
 - cleft, 558
 - gap, 558
 - vesicles, 558
 - weights, 521
- synchronization, 494, 505, 541

- synchronization of chaos, 505
- synchronous updating, 537
- target vector, 524, 527
- targeting, 497
- $\tau(q)$, 449
- Taylor series expansion, 56, 78, 371, 417
- tent map, 348, 608
- 3D plot, 23
- three-dimensional system, 186
- threshold, 559
 - logic, 564
 - value, 85
- time series, 105, 364
 - chaos detection, 198
 - plot, 198
- Tinkerbell map, 601
- Toda Hamiltonian, 240
- topological dimension, 448
- topologically equivalent, 74
- torus, 228
- total degree, 253
- totally
 - connected, 388
 - disconnected, 388
- training, 524
- trajectory, 66, 118
- transcritical bifurcation, 167
- transfer function, 521, 551
- transient, 51
- transmitted, 407
- transversal, 247
- transversely, 216
- travelling salesman problem, 531
- triangular pulse, 421
- trigsimp, 130
- trivial fixed point, 356
- turbulence, 453, 492
- 2D plot, 21
- two-neuron module, 533
- uint8, 476
- unconstrained optimization
 - problem, 527
- uncoupled, 187
- uniform asymptotic expansion, 127
- uniform harvesting, 341
- unipolar activation function, 522
- uniqueness theorem, 53
- universality, 365
- Unix, 2
- unstable
 - critical point, 55, 153
 - fixed point, 361, 388
 - focus, 69
 - limit cycle, 118
 - manifold, 72, 79, 186, 191
 - node, 68
- unsupervised learning, 524
- vacuum, 405
- vacuum tube oscillator, 571
- value of homes in Boston, 529
- van der Pol equation, 130
- van der Pol system, 114, 259
- vector field, 66
 - plot, 533
- velocity of light, 413
- Verhulst's equation, 38, 96
- virus, mobile phone, 592
- viscosity, 200
- viscous fingering, 434
- volt, 48
- voltage drop, 48
- wave equations, 404
- wave vector, 406
- wavelength, 406
 - light, 413
- W_C , 191
- while loop, 6, 29

Windows, 2
wing rock, 175
WinPython, 14
 W_S , 79, 191
 W_U , 79, 191
X-ray spectroscopy, 453

XOR gate, 522
You Tube, 386
youngest class harvesting, 339
 Z_q , 448