
Texts in Computer Science

Series editors

David Gries, Dept of Computer Science, Cornell University, Ithaca, New York, USA

Orit Hazzan, Faculty of Education in Technology and Science, Technion–Israel Institute of Technology, Haifa, Israel

Fred B. Schneider, Cornell University, Ithaca, New York, USA

More information about this series at <http://www.springer.com/series/3191>

Claudio Cioffi-Revilla

Introduction to Computational Social Science

Principles and Applications

Second Edition

Claudio Cioffi-Revilla
George Mason University
Fairfax, VA
USA

ISSN 1868-0941 ISSN 1868-095X (electronic)
Texts in Computer Science
ISBN 978-3-319-50130-7 ISBN 978-3-319-50131-4 (eBook)
DOI 10.1007/978-3-319-50131-4

Library of Congress Control Number: 2016959534

© Springer International Publishing AG 2014, 2017

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.

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

*To my Lady Jean, L.G.C.H.S.,
on our XLIV anniversary*

Preface to the Second Edition

Numerous developments have taken place in Computational Social Science (CSS) in the short time since the first edition of this textbook appeared in 2014. They include new university and college programs and curricula, in addition to many exciting research directions offered by big data analytics, advances in social complexity, and innovations in computational modeling tools. Reviews and comments by readers of the first edition have been encouraging, so this second edition provides a number of useful enhancements and corrections to the first.

This edition contains sets of questions, problems, and exercises in each chapter. Their purpose is multifaceted: to test what has been learned; to develop deeper understanding through problem-solving; to exercise critical thinking in support of scientific learning; to test or write code to implement ideas learned or in need of further exploration; or to apply principles in diverse social domains, in different situational contexts, or in particular disciplines.

If you are inclined, send me your responses to exercises and problems. I am happy to acknowledge and select the best for mention in the next edition.

Questions and problems are queries with exact answers, whereas exercises are more open-ended scientific inquiries for exploring and discussing various facets of the material covered in each chapter. Both are intended to solidify and extend knowledge, and to test understanding concerning some of the most important ideas presented in the main content of each chapter. Another function of problems and exercises is to delve deeper into the foundations of CSS, through special topics that could seem to branch off from or interrupt the main flow of the chapter. The answers to most questions and problems are provided in a separate section at the end of the book.

In each chapter, problems and exercises are presented in approximately the same order as the subject matter in the chapter, with very few exceptions. These include cases where knowledge is tested cumulatively, based on a combination of material drawn from two or more sections.

There are many more questions, problems, and exercises than can be assigned in a single semester-long course, or perhaps even in a year-long course. The purpose for this is to allow each instructor some flexibility in selecting the items, and students the opportunity to investigate additional ideas. A number of the exercises also provide ideas for more advanced exams, research papers, or theses. Quite a

number of them can also be used for group assignments to practice collaboration among students and assistance in coordination or mentoring by the instructor. Many exercises also lend themselves to creating interesting posters, which can then adorn a CSS learning environment by integrating research and teaching.

The first draft of these problems and exercises was written during the 2015 Lipari Summer School in Computational Social Science, in the last week of July, and completed during a sabbatical leave in the spring and summer of 2016. I am grateful to colleagues, students, and several readers of the first edition, especially Rob Axtell, Andrew Crooks, Harsh Gupta, Chenyi Hu, František Kalvas, Bill Kennedy, and Dan Rogers for their comments and suggestions.

Alexandria, VA, USA

Claudio Cioffi-Revilla

Preface to the First Edition

This textbook provides an introduction to Computational Social Science (CSS), an emerging field at the intersection of traditional social science disciplines, computer science, environmental science, and engineering sciences. CSS is inspired by 20th century pioneers such as Herbert A. Simon, who saw essentially a new way of doing social science enabled by computational science and technology. Scientist and visionary Peter J. Denning once said that “the science of the 21st century will be computational,” so this book is proof of that idea in social science domains.

As a textbook, this is intended as a systematic introductory survey to familiarize the reader with the overall landscape of CSS, including its main concepts, principles, applications, and areas of research. CSS investigates social complexity at all levels of analysis—cognitive, individual, group, societal, and global—through the medium of computation, as we will examine in greater detail in Chap. 1. This book is not intended as an advanced, specialized monograph to develop deep expertise.

The need for this book arose from the lack of unified treatment of the various areas of theory and research in CSS. As a consequence, those of us involved in teaching this new subject have been constrained to use a disparate library of readings without a single, unified framework. This book aims to be both comprehensive (include all major areas of CSS) and scientifically integrated by an overarching framework inspired by the paradigm of complex adaptive systems, as developed by Simon and his contemporaries in what may now be called the Founders’s Generation (described in Chap. 1).

This project originated from the course on Introduction to CSS that has been taught at George Mason University for the past ten years. It is the core course in CSS, required of all students entering our graduate program in the Department of Computational Social Science. Initially, I taught the course, then other colleagues joined. Approximately ten students have taken the course each year, mostly from the CSS program, but also from other departments across the social sciences, computer science, environmental science, and engineering sciences.

This book is intended for two types of readers, which reflect the diverse student communities who have taken this course over the years. Some students will use it as a one-time, comprehensive exposure to the field of CSS. Other students might use it as foundation for further study through more advanced, specialized work in one or more of the areas surveyed here. This book should also be helpful to students

preparing for their doctoral examination in CSS, as a review of basic ideas and a way to integrate knowledge.

The background assumed of the reader consists of some familiarity with one or more of the social sciences at a level equivalent to undergraduate study, basic knowledge of programming in any language (nowadays Python has become quite popular and is an excellent language for learning about computation), and some ability to follow mathematical modeling using logic, elementary probability, and basic calculus. Higher mathematics are unnecessary for introducing CSS.

The plan of the book is as follows: Chapter 1 provides an introduction, focusing primarily on the meaning of complex adaptive systems in social domains, including the significance of Herbert A. Simon's seminal theory and the paradigm it provides for CSS. This initial chapter also explains the main areas of CSS covered in this textbook, which are taken up in Chaps. 3 to 10. Chapter 2 provides a review of basic ideas in computing from a social science perspective, or computation as a paradigm for developing social science; it is *not* intended as a substitute for formal instruction on computation and programming for social scientists.

The following chapters cover major areas of CSS, corresponding to four distinct methodological approaches, as summarized in Sect. 1.6:

- Automated information extraction (Chap. 3)
- Social networks (Chap. 4)
- Social complexity:
 - Origins and measurement (Chap. 5)
 - Laws (Chap. 6)
 - Theories (Chap. 7)
- Social simulation:
 - Methodology (Chap. 8)
 - Variable-based models (Chap. 9)
 - Object-based (Chap. 10)

Each chapter contains a brief opening section introducing and motivating the chapter. This is followed by a section summarizing some of the history of CSS in the chapter's area, based on significant milestones. The purpose of these historical chronologies associated with each chapter's theme is to make the reader aware of significant scientific roots of the field of CSS, including its braided development with related disciplines; it does not provide a systematic history. Each chapter also includes a list of Recommended Readings, primarily intended as a guide for deepening understanding of each chapter, not as exhaustive bibliographies.

The style of the textbook attempts to strike a balance between an informal, reader-friendly, narrative tone, and a more formal tone that is necessary for highlighting rigorous concepts and results. Concept formation is a major emphasis, as is the statement of laws and principles from theory and research in quantitative social science, especially formal theory and empirically validated models. Along these lines, an effort is made, beginning in Chap. 2, to provide CSS with systematic,

scientific, graphic notation that has been so sadly lacking in traditional social science. This is done by adopting the Unified Modeling Language (UML) as a viable system for describing social complexity through graphic models that have powerful analytical meaning, as well as having direct correspondence with computation and code. Mathematical notation used in this book is standard and aims at maintaining consistency across chapters.

Finally, in terms of possible uses of this textbook, instructors may consider the following options. The ten chapters of this textbook are normally more than sufficient for a one-semester course, because some chapters will require more than one week to work through. Chapter 1 is best covered in a single session. Chapter 2 can easily be covered in two sessions, by dedicating the second session to UML. Chapters 4, 5, 6, 7, 9, and 10 can also each be covered in two sessions, by dividing the material into the main sections composing each chapter. Hence, another option is to use this textbook for a two-semester sequence, as is done in many other fields. This extended format would also permit more use of Recommended Readings, supplemented by additional bibliography, and spending more time analyzing examples to deepen understanding of concepts and principles. Readers are strongly encouraged to use the list of Recommended Readings to study the classic works, which are highlighted in the historical section at the beginning of each chapter.

This book has benefited from significant feedback from students, so I welcome future suggestions for corrections and improvements. I hope you, the reader, enjoy learning from this book at least as much as I have enjoyed writing it.

Washington, DC
September 2013

Claudio Cioffi-Revilla

Acknowledgements

During the past four decades I have benefited from scientific discussions with mentors, colleagues, and students who have influenced my research and teaching in Computational Social Science. Much of my original interest in the field came from discussions with Herbert (“Herb”) A. Simon and members of the Triple-I Seminar on Complex Systems during the 1980s, including Elinor (“Lin”) and Vince Ostrom and Harvey Starr from Indiana University, Dina A. Zinnes, Dick Merritt, Bob Muncaster, Jim Kuklinsky, and Mike Krassa from the Merriam Lab at the University of Illinois at Urbana-Champaign, and Bob Boynton from the University of Iowa. Discussions with Karl Deutsch, Ed Azar, Andy Scott, Harold Guetzkow, Bruce Russett, Hayward Alker, Raoul Narroll, Steve Wolfram, Larry Smarr, Benoît Mandelbrot, Ray Dacey, Martin Shubik, Dwain Meford, Jim Rosenau, Pierre Allan, Giorgio Natalicchi, Sam Kotz, and Kurt Johnson from the earlier phase of my academic career are still memorable. Craig Murphy, Doug Nelson, Chuck Taber, Kelly Kadera, Terry Clark, and Paul Pudiate were among my earliest students. When I moved to the University of Colorado at Boulder I learned a great deal from John Rundle and colleagues at the Colorado Center for Chaos and Complexity (C4), especially V.J. Gupta and Liz Bradley.

This textbook grows out of the interdisciplinary Program in Computational Social Science at George Mason University, which I founded in 2002 through joint teamwork with numerous students, faculty, staff, and administrators from the Mason campus. At the risk of unintentionally omitting someone, I wish to thank the many who have helped me in myriad ways: Giorgio Ascoli, Rob Axtell, Peter Balint, Jackie Barker, Ernie Barreto, Andrea Bartoli, Jeff Bassett, Sheryl Beach, Jim Beall, Pete Becker, Tony Bigbee, Christina Bishop, Kim and Sharon Bloomquist, Gary Bogle, Annetta Burger, Joey Carls, Randy Casstevens, Gabriel Catalin Balan, Debbie Boehm-Davis, Dan Carr, Jack Censer, Guido Cervone, Kai-Kong Chan, Barbara Cohen, Marc Coletti, Jim Conant, Tim Conlan, Chenna Cotla, Julie Christensen, Andrew Crooks, Paul Cummings, David Davis, Ken De Jong, Dan Druckman, Bob Dudley, Debbie V. Duong, Kim Eby, Allan Falconer, Win Farrell, Tatiana Filatova, Kim Ford, Jennifer Fortney, Aaron Frank, Brendon Fuhs, Jim Gentle, Aldona Gozikowski, Omar Guerrero, Cathy Gallagher, Jack Goldstone, Jon Gould, John Grefenstette, Beth Grohnke, Greg Guagnano, Renate Guilford, Tim Gulden, Ates Hailegiorgis, Joey Harrison, Melissa Hayes, Kingsley Haynes, Dee

Holisky, Bill Honeychurch, Dan Houser, Chris Hyungsik Shin, Bob Jonas, Chris Jones, Mark Katz, Bill Kennedy, Matt Koehler, Maction Komwa, Dorothy Kondal, Raj Kulkarni, Mike Laskofski, Maciej Latek, Randy Latimer, Kate Leonard, Alex Levis, Collette Lawson, Ann Ludwick, Cynthia Lum, José Manuel Magallanes, Julie Mahler, Michelle Marks, David Masad, Steve Mastrofski, Kevin McCabe, Mike McDonald, Hugh McFarlane, Danny Menascé, Alan Merten, Tish Moreno, Michael Naor, Johnny Nelson, Jim Olds, Leslie Painter, Liviu Panait, Dawn Parker, Ann Palkovich, Sean Paus, Nicolas Payette, Carolyn Payne, Kathleen Pérez-López, Bianica Pint, Margaret Polski, Paul Posner, Steve Prior, Denise Quinto, Pris Regan, Cindy Roberts, Suzanne Robbins, Pedro Romero, Tom Rosati, Dave Rossell, Mark Rouleau (our first Ph.D. in CSS), Cathy Rudder, John Sacco, Mickey Satija, Tim Sauer, Laurie Schintler, Paul Schopf, Linda Schwartztein, Jagadish Shukla, Steve Scott, James Snead, Paul So, Arun Sood, Peter Stearns, Roger Stough, Jennifer Sturgis, Keith Sullivan, Burak Tanyu, Rhonda Troutman, Max Tsvetovat, Karen Underwood, Dick Wagner, Nigel Waters, Shandra Watson, Jane Wendelin, Steve Wilcox, Debbie Williams, Sarah Wise, David Wong, Chun-Yi Yang, Carol Zeeve, and Matt Zingraff. Their cumulative efforts have enabled Mason's Graduate Program in Computational Social Science (over a dozen courses in CSS, a Certificate, a Masters in Interdisciplinary Studies/CSS, and a Ph.D. degree), the Department of Computational Social Science, and the generative unit for CSS at Mason: the Center for Social Complexity (CSC).

I have learned much from co-authoring publications, developing new courses and grant proposals, discussing theory and research, and organizing events with a large and stimulating community of colleagues from around the world, including: Tef Abate, Petra Ahrweiler, Guillermo Algaze, Luís Antunes, Aruna Apte, George Atkinson, Fulvio Attinà, Scott Atran, Brent Auble, Tom Baerwald, Bill Bainbridge, Steve Bankes, Mike Batty, Ana Lucia Bazzan, Russ Bernard, Brian Berry, Ravi Bahvnani, Dmitri Bondarenko, Nathan Bos, Peter Brecke, Stuart Bremer, Cathy Cameron, Kathleen Carley, Cristiano Castelfranchi, John Casti, Lars-Erik Cederman, Fahmida Chowdhury, Alfred Cioffi, Wayne Clough, Helder Coelho, Louise Comfort, Rosaria Conte, Chet Cooper, Linda Cordell, Angela Corolla, Nuno David, Guillaume Deffaunt, Hiroshi Deguchi, Christophe Deissenberg, Jerry Dobson, David Dornish, Jim Doran, Massimo Drei, Julie Dugdale, Bruce Edmonds, Giorgio Einaudi, Carol and Mel Ember, Josh Epstein, Mike Fischer, Bill Fitzhugh, Bruno Frohlich, José Manuel Galán, Jianbo Gao, Michele Gelfand, Nigel Gilbert, Gary Goertz, Rebecca Goolsby, Nick Gotts, Ariel Greenberg, Steve Guerin, Alessandro Guidi, George Gumerman, Myron Gutmann, David Hales, Dirk Helbig, Matt Hoffmann, Barry Hughes, Luís Izquierdo, Wander Jager, Eric Jones, Steve Kaisler, Anna Kerttula, Dennis King, Alan Kirman, Jürgen Klüver, Tim Kohler, Nick Kradin, Arie Kruglanski, Larry Kuznar, Steve Lansing, Efraim Laor, Randy Latimer, Steve Lekson, Nicola Lettieri, Mark Lichbach, David Lightfoot, Fred Liljeros, Corey Lofdahl, Urs Luterbacher, Thomas Lux, Patty Mabry, Charles Macal, Ed MacKerrow, Michael Macy, Greg Madey, Artemy Malkov, Joyce Marcus, Jack Meszaros, Manny Midlarsky, Jeff Millstein, Byong Won Min, Harold Morowitz, ScottMoss, Akira Namatame, Dana Nau, Martin Neumann, Michael North, Andrzej

Nowak, Sean O'Brien, Paul Ormerod, John Padgett, Mario Paolucci, Domenico Parisi, Peter Peregrine, Peter Perla, Gary Polhill, Brian Pollins, Denise Pumain, Rodolfo Ragonieri, Bill Rand, Dwight Read, Colin Renfrew, Bob Reynolds, Fred Roberts, J. Daniel Rogers, Juliette Rouchier, Dieter Ruloff, Jerry and Paula Sabloff, John Salerno, David Sallach, Lena Sanders, Todd Sandler, Antonio Sanfilippo, Dez Saunders-Newton, Vittorio Scarano, Steve Schlosser, Phil Schrodt, Lee Schwartz, Frank Schweitzer, Payson Sheets, Andrew Sherratt, Carl Simon, Ian Skoggard, Ricard Solé, Jim Spohrer, Detlef Sprinz, Flaminio Squazzoni, Gene Stanley, John Sterman, Christina Stoica, Rick Stoll, Gary Strong, Lee Schwartz, David Sylvan, Rein Taagepera, Keiki Takadama, John Tangney, Takao Terano, Pietro Terna, Rita Teutonico, Jim Thomas, Qing Tian, Klaus Troitzsch, Peter Turchin, Alex Vespignani, Mitch Waldrop, David Warburton, Paul Werbos, Jon Wilkenfeld, and Peyton Young.

A graceful invitation from my colleague and friend, Shu-Heng Chen, to deliver the 2011 Herbert A. Simon Lecture Series in Computational Social Science at National Chengchi University in Taipei, Taiwan, provided a unique opportunity to organize my ideas for this textbook. A preview of this textbook was provided in March 2012 at the invitation of the Center for the Study of Complex Systems at the University of Michigan. I am grateful to Robert Axelrod, John Holland, Scott Page, Rick Riolo, and their students for sharing their insights and suggestions.

Some of the examples or modeling applications discussed in this book originated or came into focus through discussions with members of the government policy and analytical community. None of them bears any responsibility for my interpretations or inferences.

Students from the Fall 2012 session of CSS 600—Introduction to CSS—helped as I finalized the outline of this textbook. I am especially grateful to Gary Bogle, Tom Briggs, Annetta Burger, Paul Cummings, José Manuel Magallanes, and Dan Pryce. Several chapters of this textbook were also used while lecturing at the Lipari International Summer School in Computational Social Science, now in its 5th year. I am grateful to students and invited faculty, including David Beaver, Kathleen Carley, Alfredo Ferro, Giovanni Giuffrida, Carlo Pennisi, Alessandro Pluchino, Kalev Leetaru, Roy Lindelauf, Huan Liu, Roel Popping, Raghu Ramakrishnan, Marc Smith, Philip Schrodt, V.S. Subrahmanian, Alberto Trobia, and Calogero Zarba.

I am especially grateful for input on various chapters received from Dan Rogers, Linda Cordell, Sean Luke, Nazli Choucri, Bill Kennedy, Siggy Scott, and Joey Harrison. I also wish to thank Jean N. Cioffi, Dorothy Kondal, and Nancy Turgeon for careful editing and assistance with proofreading.

Support received from the US National Science Foundation and the Office of Naval Research, as well as from the Center for Social Complexity and the Provost's Office at George Mason University, especially intellectual support received from Provost Peter Stearns, is gratefully acknowledged.

I also wish to thank Wayne Wheeler and Simon Rees, my editors at Springer, for their encouragement and patience. They are among the most professional, persevering, and pleasant editors I have worked with.

Images used in this textbook were produced by me or obtained from NASA, [PublicDomainPictures.net](#), Wikipedia, Malteser International, and faculty webpages without copyrights.

Contents

Preface to the Second Edition	vii
Preface to the First Edition	ix
Acknowledgements	xiii
Acronyms	xxiii
List of Figures	xxvii
List of Tables	xxxv
1 Introduction	1
1.1 What Is Computational Social Science?	1
1.2 A Computational Paradigm of Society	2
1.3 CSS as an Instrument-Enabled Science	3
1.4 Examples of CSS Investigations: Pure Scientific Research Versus Applied Policy Analysis	4
1.5 Society as a Complex Adaptive System	7
1.5.1 What Is a CAS in CSS?	7
1.5.2 Tripartite Ontology of Natural, Human, and Artificial Systems	9
1.5.3 Simon's Theory of Artifacts: Explaining Basic Social Complexity	10
1.5.4 Civilization, Complexity, and Quality of Life: Role of Artificial Systems	11
1.6 Main Areas of CSS: An Overview	12
1.6.1 Automated Social Information Extraction	13
1.6.2 Social Networks	14
1.6.3 Social Complexity	14
1.6.4 Social Simulation Modeling	15
1.7 A Brief History of CSS	18
1.8 Main Learning Objectives	20
Problems	21
Exercises	26
Recommended Readings	32

2	Computation and Social Science	35
2.1	Introduction and Motivation	35
2.2	History and First Pioneers	36
2.3	Computers and Programs	37
2.3.1	Structure and Functioning of a Computer	37
2.3.2	Compilers and Interpreters	40
2.4	Computer Languages	40
2.5	Operators, Statements, and Control Flow	45
2.6	Coding Style	47
2.7	Abstraction, Representation, and Notation	48
2.8	Objects, Classes, and Dynamics in Unified Modeling Language (UML)	53
2.8.1	Ontology	53
2.8.2	The Unified Modeling Language (UML)	57
2.8.3	Attributes	68
2.8.4	Operations	71
2.9	Data Structures	74
2.10	Modules and Modularization	76
2.11	Computability and Complexity	77
2.12	Algorithms	78
	Problems	80
	Exercises	93
	Recommended Readings	101
3	Automated Information Extraction	103
3.1	Introduction and Motivation	103
3.2	History and First Pioneers	104
3.3	Linguistics and Principles of Content Analysis: Semantics and Syntax	107
3.4	Semantic Dimensions of Meaning: From Osgood to Heise	109
3.4.1	EPA-Space and the Structure of Human Information Processing and Meaning	109
3.4.2	Cross-Cultural Universality of Meaning	111
3.5	Data Mining: Overview	112
3.6	Data Mining: Methodological Process	114
3.6.1	Research Questions	115
3.6.2	Source Data: Selection and Procurement	116
3.6.3	Preprocessing Preparations	116
3.6.4	Analysis	117
3.6.5	Communication	124
	Problems	124
	Exercises	133
	Recommended Readings	139

4	Social Networks	141
4.1	Introduction and Motivation	141
4.2	History and First Pioneers	142
4.3	Definition of a Network	147
4.3.1	A Social Network as a Class Object	148
4.3.2	Relational Types of Social Networks	149
4.3.3	Level of Analysis	150
4.3.4	Dynamic Networks	151
4.4	Elementary Social Network Structures	152
4.5	The Network Matrix	155
4.6	Quantitative Measures of a Social Network	155
4.6.1	Nodal Measures: Micro Level	156
4.6.2	Network Measures: Macro-Level	157
4.7	Dynamic (Actually, Kinetic) Networks as Ternary Associations	158
4.8	Applications	159
4.8.1	Human Cognition and Belief Systems	159
4.8.2	Decision-Making Models	163
4.8.3	Organizations and Meta-Models	163
4.8.4	Supply Chains	165
4.8.5	The Social Structure of Small Worlds	167
4.8.6	International Relations	167
4.9	Software for SNA	168
	Problems	170
	Exercises	182
	Recommended Readings	191
5	Social Complexity I: Origins and Measurement	193
5.1	Introduction and Motivation	193
5.2	History and First Pioneers	193
5.3	Origins and Evolution of Social Complexity	196
5.3.1	Sociogenesis: The “Big Four” Primary Polity Networks	197
5.3.2	Social Complexity Elsewhere: Secondary Polity Networks	201
5.3.3	Contemporary Social Complexity: Globalization	202
5.3.4	Future Social Complexity	203
5.4	Conceptual Foundations	205
5.4.1	What Is Social Complexity?	205
5.4.2	Defining Features of Social Complexity	206
5.5	Measurement of Social Complexity	209
5.5.1	Qualitative Indicators: Lines of Evidence	210
5.5.2	Quantitative Indicators	212

Problems	219
Exercises	234
Recommended Readings	245
6 Social Complexity II: Laws	247
6.1 Introduction and Motivation	247
6.2 History and First Pioneers	247
6.3 Laws of Social Complexity: Descriptions	249
6.3.1 Structural Laws: Serial, Parallel, and Hybrid Complexity	249
6.3.2 Distributional Laws: Scaling and Nonequilibrium Complexity	255
6.4 Power Law Analysis	264
6.4.1 Empirical Analysis: Estimation and Assessing Goodness of Fit	264
6.4.2 Theoretical Analysis: Deriving Implications	267
6.5 Universality in Laws of Social Complexity	272
Problems	272
Exercises	280
Recommended Readings	288
7 Social Complexity III: Theories	291
7.1 Introduction and Motivation	291
7.2 History and First Pioneers	291
7.3 Theories of Social Complexity: Elements of Explanation	294
7.3.1 Sequentiality: Modeling Processes. Forward Logic	295
7.3.2 Conditionality: Modeling Causes. Backward Logic	299
7.3.3 Hybrid Bimodal Social Complexity: Several-Among-Some Causes	303
7.4 Explaining Initial Social Complexity	304
7.4.1 Emergence of Chiefdoms	310
7.4.2 Emergence of States	319
7.5 General Theories of Social Complexity	328
7.5.1 Theory of Collective Action	328
7.5.2 Simon's Theory of Adaptation via Artifacts	331
7.5.3 Canonical Theory as a Unified Framework	335
Problems	341
Exercises	360
Recommended Readings	371
8 Simulations I: Methodology	375
8.1 Introduction and Motivation	375
8.2 History and First Pioneers	376

8.3	Purpose of Simulation: Investigating Social Complexity Via Virtual Worlds	377
8.4	Basic Simulation Terminology	379
8.5	Fidelity of Representation and Implications	382
8.6	Types of Social Simulation: From System Dynamics to Agent-Based Models	383
8.7	Development Methodology of Social Simulations	384
8.7.1	Motivation: What Are the Research Questions Addressed by a Given Model?	384
8.7.2	Conceptual Design: What Does the Abstraction Look Like?	386
8.7.3	Implementation: How Is the Abstracted Model Written in Code?	387
8.7.4	Verification: Does the Simulation Perform as Intended?	388
8.7.5	Validation: Can We Trust the Results?	389
8.7.6	Virtual Experiments and Scenario Analyses: What New Information Does the Simulation Generate?	390
8.8	Assessing the Quality of a Social Simulation	391
8.8.1	General Principles for Social Modeling Assessment	391
8.8.2	Dimensions of Quality in Social Simulation Models	393
8.9	Methodology of Complex Social Simulations	396
8.10	Comparing Simulations: How Are Computational Models Compared?	398
	Problems	400
	Exercises	408
	Recommended Readings	413
9	Simulations II: Variable-Oriented Models	415
9.1	Introduction and Motivation	415
9.2	History and First Pioneers	415
9.3	System Dynamics Models	417
9.3.1	Motivation: Research Questions	419
9.3.2	Design: Abstracting Conceptual and Formal Models	419
9.3.3	Implementation: System Dynamics Software	425
9.3.4	Verification	426
9.3.5	Validation	426
9.3.6	Analysis	427
9.4	Queueing Models	429
9.4.1	Motivation: Research Questions	429
9.4.2	Design: Abstracting Conceptual and Formal Models	432

9.4.3	Implementation: Queuing Systems Software	435
9.4.4	Verification	435
9.4.5	Validation	436
9.4.6	Analysis.	436
	Problems.	437
	Exercises.	445
	Recommended Readings	453
10	Simulations III: Object-Oriented Models.	455
10.1	Introduction and Motivation	455
10.2	History and First Pioneers	455
10.3	Cellular Automata Models	459
10.3.1	Motivation: Research Questions	463
10.3.2	Design: Abstracting Conceptual and Formal Models	463
10.3.3	Implementation: Cellular Automata Software	466
10.3.4	Verification	468
10.3.5	Validation	468
10.3.6	Analysis.	469
10.4	Agent-Based Models.	470
10.4.1	Motivation: Research Questions	473
10.4.2	Design: Abstracting Conceptual and Formal Models	476
10.4.3	Implementation: Agent-Based Simulation Systems	479
10.4.4	Verification	482
10.4.5	Validation	482
10.4.6	Analysis.	483
	Problems.	484
	Exercises.	496
	Recommended Readings	508
	Answers to Problems	513
	Glossary	543
	References	585
	Author Index.	593
	Subject Index.	601

Acronyms

ABM	Agent-based model
ACE	Agent-based computational economics
ACM	Association for Computing Machinery
AI	Artificial intelligence
AND	Boolean conjunctive operator
BDI	Beliefs, desires, intentions
CA	Cellular automaton or automata
CAMEO	Conflict and Mediation Event Observations
CAS	Complex adaptive system
CASOS	Center for Computational Analysis of Social and Organizational Systems, Carnegie Mellon University
CCDF	Complementary cumulative density function (also c.c.d.f.)
CDF	Cumulative density function (also c.d.f.)
CIDCM	Center for International Development and Conflict Management, University of Maryland
CIKM	Conference on Information and Knowledge Management of the ACM
CMU	Carnegie Mellon University
COA	Course of action
COPDAB	Conflict and Peace Data Bank
CPU	Central processing unit
CSC	Center for Social Complexity, George Mason University
CSS	Computational Social Science
CSSN	Computer-supported social networks
CSSSA	Computational Social Science Society of the Americas
DARPA	Defense Advanced Research Projects Agency
DDR3 SDRAM	Double-data-rate three synchronous dynamic random access memory
DYNAMO	DYNAmic MOdels
EC	Evolutionary computation
ECPR	European Consortium for Political Research
ECML-PKDD	European Conference on Machine Learning and Principles and Practices of Knowledge Discovery in Databases

EOS	Evolution of Organized Society project, University of Essex
EPA	Evaluation, potency, activity. Dimensions of Osgood's semantic space
ERG	Exponential random graph
EU	European Union
FEARLUS	Framework for the Evaluation and Assessment of Regional Land Use Scenarios
FIFO	First-in-first-out
FILO	First-in-last-out
FORTTRAN	FORmula TRANslation
GB	Gigabyte
GCM	General Circulation Model
GDELT	Global Data on Events, Location, and Tone
GeoMASON	Geospatial MASON
GHz	Gigahertz
GIS	Geographic Information System
GPU	Graphic processing unit
GUI	Graphic user interface
HMM	Hidden Markov model
HPC	High-performance computing
HRAF	Human Relations Area Files, Yale University
I/O	Input-output
ICPSR	Interuniversity Consortium for Political and Social Research
ICR	Institute for Communications Research, University of Illinois at Urbana-Champaign
IEEE	Institute of Electrical and Electronic Engineers
INSNA	International Network for Social Network Analysis
IPCC	Intergovernmental Panel on Climate Change
ISIMADE	International Symposium on Intelligent Multimedia and Distance Education
ISS	International Space Station
JVM	Java virtual machine
KWIC	Keywords in context
KWOC	Keywords out of context
kya	Thousands of years ago
LEO	Low Earth orbit
LIFO	Last-in-first-out
LILO	Last-in-last-out
LISP	LISt Processing
LOC	Lines of code
LRD	Long-range dependence
LUCC	Land-Use and Cover Change
M2M	Model-to-model
MAS	Multi-agent system or systems
MASON	Multi-Agent Simulator of Networks or Neighborhoods

MC	Monte Carlo
MDIVVA	Motivate-design-implement-verify-validate-analyze
MDS	Multi-dimensional scaling
MINUIT	Numerical minimization computer program
MIT	Massachusetts Institute of Technology
MLE	Maximum likelihood estimate, estimator, or estimation
NAACSOS	North American Association for Computational Social and Organizational Sciences
NASA	National Aeronautics and Space Administration
NATO	North Atlantic Treaty Organization
NER	Named entity recognition
NIST	National Institute of Standards and Technology
NRR	Normal relations range
NSF	National Science Foundation
NVAC	National Visualization Analytics Center, PNNL
OCR	Optical character recognition
OMG	Object Management Group
ONR	Office of Naval Research
OO	Object-oriented
OOM	Object-oriented model or modeling
OOP	Object-oriented program or programming
OR	Boolean disjunctive operator
ORA	Entity extraction algorithm by CASOS
PDF	Probability density function (also p.d.f.)
PNAS	Proceedings of the National Academy of Sciences of the USA
PNNL	Pacific Northwest National Laboratory, Department of Energy
PPNB	Pre-Pottery Neolithic B period
PRNG	Pseudo-random number generator
RAM	Random access memory
RNG	Random number generator
SAS	Statistical Analysis System
SD	System dynamics
SDC	Size, development, and capability
SEQAND	Boolean sequential conjunctive operator
SES	Socioeconomic status
SIAM	Society for Industrial and Applied Mathematics
SIGKDD	Special Interest Group on Knowledge Discovery and Data Mining of the ACM
SIMPEST	Simulation of Political, Economic, Social, and Technological Systems
SIMPLE	Simulation of Industrial Management Problems with Lots of Equations
SIMPOP	SIMulation of POPulation project, University of Paris-Sorbonne
SNA	Social network analysis

SOCPAC	A FORTRAN IV program for structural analysis of sociometric data
SPSS	Statistical Package for the Social Sciences
SSRC	Social Science Research Council
SSRN	Social Science Research Network
STELLA	System dynamics simulation system
TABARI	Textual Analysis by Augmented Replacement Instructions
TBJ	Truth, beauty, and justice
TRIAL	Technique for Retrieval of Information and Abstracts of Literature
UAV	Unmanned autonomous vehicle
UCINET	University of California-Irvine social network analysis software
UCLA	University of California-Los Angeles
UML	Unified Modeling Language
UN	United Nations
URL	Uniform resource locator
US	United States
USSR	Union of Soviet Socialist Republics
VENSIM	System dynamics simulation system
WWW	World-Wide Web
XOR	Boolean exclusive disjunctive operator

List of Figures

Figure 2.1	A computer with its functional components (the five boxes) based on a bus architecture (fast-speed data connections)	38
Figure 2.2	A “social world” consists of a social system situated in its environment. This ontology is foundational for many social theories examined through formal and empirical analysis, including Simon’s Theory of Artifacts, the Canonical Theory, and others based on the Complex Adaptive Systems Paradigm. Unfortunately, this graphic representation is useless although common throughout social science. Later in this section we introduce UML as a helpful graphic notation system for representing social worlds	54
Figure 2.3	Ontology across scales of human and social systems complexity: The family is the smallest kin-based social system (<i>upper left</i>). Teams of people provide assistance in humanitarian crises and disasters (<i>upper right</i>). Politics are complex social aggregates capable of producing historical milestones (<i>lower left</i>). Humans in space constitute complex, coupled, socio-technical systems operating in extreme environments (<i>lower right</i>)	54
Figure 2.4	UML class diagram of a basic world ontology consisting of a social system and its environment. Note that this graph is intended to represent the same as Fig. 2.2, but it conveys much more information	59
Figure 2.5	Associations among classes or objects are drawn in UML using arrows with different arrowheads that denote different types of relations (e.g., social relations, socio-environmental interactions, or others). Unlike the informal and widespread use of arrows in many social science illustrations, the notation for social relations modeled with a class diagram is formal and strictly	

	defined, making meanings inter-subjective and reliable from a conceptual and terminological perspective. Examples of each type of social relation are provided in the main text.	61
Figure 2.6	UML class diagram of the standard model of a polity in political science. The diagram consists of four entities and three types of associations that denote different kinds of social relations, as explained in the main text. Diagrams such as these, and subsequent versions with more details, are valuable for communicating between social science modelers and computer programmers in charge of code implementation. Adapted from Cioffi-Revilla (2008)	65
Figure 2.7	UML sequence diagram of basic dynamic processes in a simple polity.	65
Figure 2.8	UML state (or “state machine”) diagram of macro system-level dynamics in a simple polity consisting of a Society stressed by issues and a Government that formulates policies to address public issues and lower or eliminate stress. A state diagram provides a more dynamic model of a polity than a class diagram, but entities (classes, objects) are not represented. <i>Source:</i> This and other UML diagrams of a polity are adapted from Cioffi-Revilla (2008)	67
Figure 2.9	UML class and object diagrams with various specifications of attributes and operations: (a) Class and object associated by inheritance, without specific attributes or operations, as in earlier class diagrams. (b) Class and object notation containing encapsulated attributes and operations shown by convention in the second and third compartments, respectively. (c) Example of class and object with some specific attributes. (d) Visibility of attributes denoted by public (<i>plus sign</i>) and private (<i>minus</i>) attribute notation. (e) Complete specification of a class with encapsulated attributes, operations, and visibilities	70
Figure 2.10	UML class diagram of the standard polity model, with specified attributes (variables). Note that each attribute is denoted by a uniquely designated name and corresponding data type.	71
Figure 2.11	UML class diagrams of a polity with class attributes and operations. The model on the <i>left</i> shows operations in the third vertical compartment of each class. The model on the <i>right</i> makes explicit the “manages” association between Government and PublicIssues, elevating the association to the higher status of a class by itself, named Policy	73

Figure 3.1	Example of a manual coding form used to record an event based on a newspaper source. Forms such as these were used in the early days of computational content analysis to record news into machine-readable format and enable statistical analysis of large amounts of data	104
Figure 3.2	Major pioneers of content analysis: Max Weber, sociologist, proposed the first large-scale content analysis in 1910 (<i>upper left</i>). Andrey Markov, mathematician, pioneered computational linguistics (<i>upper right</i>). Harold Lasswell pioneered computational content analysis (<i>lower left</i>). Charles E. Osgood discovered and quantified semantic space (<i>lower right</i>)	105
Figure 3.3	Word frequencies automatically extracted from Herbert A. Simon's autobiography using the Wordle™ algorithm. <i>Source</i> Simon (1992)	108
Figure 3.4	<i>Osgood's 3D Semantic Differential EPA-space</i> . The cognitive dimensions of evaluation <i>E</i> (ranging from good to bad), potency <i>P</i> (strong to weak), and activity <i>A</i> (fast to slow) span a three-dimensional semantic space. In Osgood-space each term or word <i>w</i> is located by a triplet (<i>e</i> , <i>p</i> , <i>a</i>) or vector $\mathbf{w} = e\mathbf{i} + p\mathbf{j} + a\mathbf{k}$ with norm given by $ \mathbf{w} = \sqrt{e^2 + p^2 + a^2}$	110
Figure 3.5	<i>General Data Mining Methodological Process</i> . Data mining for automated information extraction involves several stages, the most important being the six highlighted here and discussed below. The core is Analysis for answering research questions, but the other five stages are just as critical for overall quality of the scientific investigation. Each of the six stages involves a variety of procedures, most of them dependent on the research questions being addressed.	114
Figure 3.6	<i>Spatial analysis using event data</i> . This Google map of the world shows the top 2,000 political events on October 7, 2013, based on the GDELT data set (Leetaru and Schrodtt 2013). Color-coded events indicate degrees of conflict (<i>red</i> and <i>yellow</i>) or cooperation (<i>green</i> and <i>blue</i>). <i>Source</i> GDELT website, downloaded October 8, 2013	120
Figure 4.1	A social network consisting of nodes and links. In this network $g = 4$ nodes and $L = 4$ links	146
Figure 4.2	UML class diagram of a social network as an object composed of node objects associated to the network by composition	148

Figure 4.3	Types of social networks according to their social relations $\mathbb{L}_{\{\ell_{1,2,\dots,L}\}}$. <i>Upper left</i> : a directed graph or digraph \mathcal{D} . <i>Upper right</i> : a signed graph \mathcal{S} with valences. <i>Lower left</i> : a weighted network \mathcal{W} . <i>Lower right</i> : a multiplex \mathcal{M} with various kinds of social relations possible between nodes. . . .	149
Figure 4.4	Structural types of social networks according to their architecture. <i>Upper left</i> : chain or line network. <i>Upper right</i> : star network. <i>Middle left</i> : Y network. <i>Middle right</i> : circle network. <i>Lower left</i> : complete network. <i>Lower right</i> : cellular network. Each structural type is represented by its associated graph, adjacency matrix \mathbf{A} and geodesic matrix \mathbf{G}	153
Figure 4.5	Long-form UML class diagram of a social network modeled as an object composed of node objects associated to the network by composition. This model highlights the nodal composition of networks while placing network links in the background	156
Figure 4.6	UML class diagram of a dynamic social network represented as a ternary association class with multiplicities. Each link in the association corresponds to a membership in one or more (up to q) concurrent networks over a period of n time units.	158
Figure 4.7	Some simple beliefs modeled as valued networks	160
Figure 4.8	Cognitive balancing by Abelson's differentiation mechanism. <i>Left</i> : Having positive relations with a country that is disliked results in an imbalanced cognition. This belief is balanced by differentiating between evil rulers and good people, and reassigning valuations to each of the new relations.	162
Figure 4.9	Network structure of the Rational Choice Model. <i>Left</i> A decision \mathbb{D} consists of choosing an alternative $A^* \in \{A_i\}$ that has the maximum expected utility over the entire set of n alternatives.	164
Figure 4.10	Meta-network model of a social event involving actors, locations, resources, and other entities denoted by nodes and links of various shapes and colors. Produced by the ORA software at the Center for Computational Analysis of Social and Organizational Systems (CASOS), Carnegie Mellon University. A complex humanitarian crisis can be represented by a meta-network linking victims affected by the disaster, relief workers, supplies and equipment, locations, and responder activities. Similar examples include financial crises and conflicts of various kinds, all of them consisting of data n -tuples that can be extracted from raw sources.	165

Figure 5.1	Global geo-chronology of origins of social complexity in the four “cradles of civilization.” <i>Source</i> Adapted from Cioffi-Revilla (2006)	198
Figure 5.2	Long-range dependence (LRD) or memory structure in time series measured by the Hurst parameter H . <i>Source</i> Adapted from Gao et al. (2013: 16)	218
Figure 6.1	<i>Structural patterns of social complexity by causal necessity and sufficiency. a</i> Serial complexity by causal conjunction; <i>b</i> parallel complexity by causal disjunction; and <i>c</i> a case of hybrid serial–parallel complexity with some parallelized disjunctive components within an overall serialized 3-conjunctive structure.	250
Figure 6.2	<i>Structural patterns of social complexity by logic conjunction and disjunction. a</i> Serial complexity by causal conjunction; <i>b</i> parallel complexity by causal disjunction; and <i>c</i> a case of hybrid serial–parallel complexity with some parallelized disjunctive components within an overall serialized 3-conjunctive structure.	251
Figure 6.3	The power law in (<i>a</i>) untransformed hyperbolic form and (<i>b</i>) linearized or log-linear form in log–log space.	255
Figure 6.4	The power law and other distribution models	257
Figure 6.5	Taxonomy of power law models according to types of dependent variables	258
Figure 6.6	“Bending” is frequently observed in visual assessment of empirical power law distributions.	264
Figure 7.1	An Abelson-balanced belief system relating multiple aspects of communal worship	311
Figure 7.2	Forward sequential causal logic tree for initial sociopolitical complexity, denoted as the contingent process $\mathcal{P}_3(\Omega)$ of politogenesis with three antecedents	314
Figure 7.3	Forward sequential causal logic tree for initial politogenesis \mathbb{C} grafted with a first-order backward conditional causal tree for complexity potential \mathbb{P} (Conditions 1–9; Sect. 7.4.1.2)	317
Figure 7.4	Forward sequential causal logic tree for Simon’s theory of adaptation and emergence of social complexity	333
Figure 7.5	Forward sequential causal logic tree for the Canonical Theory of emergence and development of social complexity. <i>The main upper part of the graph illustrates the fast process. Change in the probability of social complexity is shown across the bottom. Node notation decisions are denoted by triangle-nodes, lotteries by square-nodes, and hybrids by diamond-nodes</i>	336

Figure 7.6	Forward sequential causal logic tree for explaining risky hazards and societal disasters according to Canonical Theory	340
Figure 8.1	<i>Basic terminology and general methodology of social simulation.</i> Social simulation methodology is an iterative process that begins with a referent system (<i>explanandum</i>) in the real world. Abstraction, formalization, programming, and appropriate data are used to develop a viable simulation model (<i>explanans</i>). This general process is independent of the specific kind of simulation model	380
Figure 8.2	UML class diagram illustrating the hierarchy of scientific models (<i>left</i>), social science models (<i>center</i>), and social simulations (<i>right</i>), each having increasingly specific standards for judging quality (moving from <i>left</i> to <i>right</i>). <i>Source</i> Cioffi-Revilla (2013)	392
Figure 9.1	Major pioneers of system dynamics models: Jay Forrester, founder of SD modeling (<i>upper left</i>); Dennis Meadows, director of the Club of Rome Project on the Predicament of Mankind, <i>The Limits to Growth</i> (<i>upper right</i>); Linda Cordell, pioneer in dynamical systems models in archeology, elected to the National Academy of Sciences in 2005 (<i>lower left</i>); Nazli Choucri, MIT pioneer SD modeler of energy, conflict, and state stability dynamics (<i>lower right</i>)	418
Figure 9.2	Causal loop diagram for a system dynamics model of norm adoption	421
Figure 9.3	Causal loop diagram for a system dynamics model of inter-group rivalry	422
Figure 9.4	SD stock and flow diagram for representing variables (stocks represented as <i>rectangles</i>) and rates of change (flow represented as <i>valves</i>)	423
Figure 9.5	Stock and flow diagram for a system dynamics model of a two-group rivalry interaction	424
Figure 9.6	Screenshot while implementing an SD social simulation using the Vensim system	425
Figure 9.7	Major pioneers of queueing systems models: Agner Krarup Erlang, founding pioneer of queueing models (<i>upper left</i>); David George Kendall, inventor of the standard notation for queueing system models (<i>upper right</i>); Thomas L. Saaty (<i>lower left</i>), author of classic works in applied discrete mathematics, including queueing theory, and inventor of the Analytic Hierarchy Process; John Dutton Conant Little, discoverer of the law of mean arrival times and a founder of modern marketing research (<i>lower right</i>)	430

Figure 9.8	The Weibull distribution. Probability density functions (<i>left</i>) and associated event intensity functions (<i>right</i>) shown for different values of the shape parameter. The Weibull distribution reduces to the exponential distribution when the shape parameter is 1.0 and approximates the normal distribution when the shape parameter is around 5	433
Figure 10.1	Major pioneers of cellular automata models: John von Neumann, inventor of cellular automata (<i>upper left</i>); John Horton Conway, inventor of the CA-based Game of Life (<i>upper right</i>); Stuart A. Bremer, pioneer computational political scientist in the use of CA models of international conflict (<i>lower left</i>); Nobel prize winner Thomas C. Schelling, famous for his model of racial segregation (<i>lower right</i>)	460
Figure 10.2	Examples of cellular automata models: The Schelling model with square cells and Moore neighborhood is initialized with ethnically mixed population (<i>upper left</i>). Racial segregation emerges as neighbors become cognizant of their surroundings and decide to move away from where they started (<i>upper right</i>). The Interhex model with hexagonal cells representing small, simple polities begins with uniformly distributed capabilities (<i>lower left</i>). As neighboring polities interact through normal balance of power dynamics, mild stochasticity is sufficient to grow a system of countries. Both models shown in this figure were implemented in MASON, discussed in Sect. 10.3.3.	462
Figure 10.3	Screenshot of a two-dimensional cellular automata model of growth with varying number of neighbors running in NetLogo	467
Figure 10.4	Pioneers of agent-based models. Joshua Epstein, creator of Sugarscape (with R. Axtell) (<i>upper left</i>); Robert Axelrod, author of <i>The Complexity of Cooperation</i> and other CSS classics (<i>upper right</i>); Nigel Gilbert, editor of <i>Journal of Artificial Societies and Social Simulation</i> (<i>lower left</i>); Hiroshi Deguchi, president of the Pacific-Asian Association for Agent-based Social Science (<i>lower right</i>)	471

- Figure 10.5 The Sugarscape agent-based model: agent behavior. The Sugarscape model consists of a society of agents (*red dots*) situated on a landscape consisting of a grid of square sites where agents with von Neumann neighborhood-vision feed on sugar (*yellow dots*). *Left* At initialization agents are assigned a uniform distribution of wealth and they reside in the southwestern region. *Right* After a number of time steps, most agents have migrated away from their original homeland as they move around feeding on the landscape. This MASON implementation by Tony Bigbee also replicates the “wave” phenomenon generated by the original (and now lost) implementation in Ascape, observed here by the northwest-southeast formations of diagonally grouped agents in the northeast region 474
- Figure 10.6 The Sugarscape agent-based model: emergence of inequality. Lorenz curves (*top*) and histograms (*bottom*) portray the distribution of agents’ wealth. *Left* Agents are assigned some wealth at initialization $t = 0$, following an approximately uniform distribution, as shown by the nearly straight Lorenz curve and wealth histogram. *Right* After some time, inequality emerges as a social pattern, as shown by the more pronounced Lorenz curve and much more skewed histogram, similar to Pareto’s Law and diagnostic of social complexity 475
- Figure 10.7 Pioneers of ABM toolkits. Swarm’s Chris Langton (*upper left*); NetLogo’s Uri Wilensky (*upper right*); Repast’s David Sallach (*lower left*); MASON’s Sean Luke (*lower right*). All of them collaborated with others in creating today’s leading simulation systems for building social ABMs. 480
- Figure 10.8 Screenshot of a Sugarscape model implemented in NetLogo. 481

List of Tables

Table 2.1	Comparison of computer programming languages. Paradigm types are explained in the text. <i>Source</i> Wikipedia, “Comparison of programming languages: General comparison”	41
Table 2.2	Main data types in Python	51
Table 2.3	Human entities and selected associations in socio-technical systems. Environments are named, not detailed.	56
Table 2.4	Social, artifactual, and natural components of coupled systems	57
Table 2.5	Multiplicity values in UML class diagrams	60
Table 3.1	Measures of association depending on levels of measurement	118
Table 4.1	Origin and evolution of the earliest social networks between 100,000 and 5,000 years ago (100–5 kya) according to system-of-systems network order $O(N)$	147
Table 5.1	Social complexity according to the polity-level Human Development Index D_H (2012) in the top fifteen countries. <i>Source</i> United Nations Development Programme, 2013 <i>Human Development Report</i>	215
Table 6.1	The Type IV power law model of social complexity compared to other common social processes and distributions	262
Table 6.2	Goodness of fit statistics used for assessment of an empirical power law	265
Table 8.1	Quality criteria for evaluating models in domains of science	392
Table 10.1	Examples of agent-based models in CSS by empirical calibration	472
Table A.1	The Richardson magnitude μ of five examples of revolutions in recent centuries	514

Table A.2	Probabilities at micro- and macro-levels of a Shannon channel, given $n = 4$ information processing stages. Overall probability P is an emergence property.	515
Table A.3	Effect of adding noise to a Shannon channel, with $n = 5$ information processing stages. Note the nonlinear effect on the overall probability of communication P and how even a high value of p results in a slightly better than even-odds value of P	515