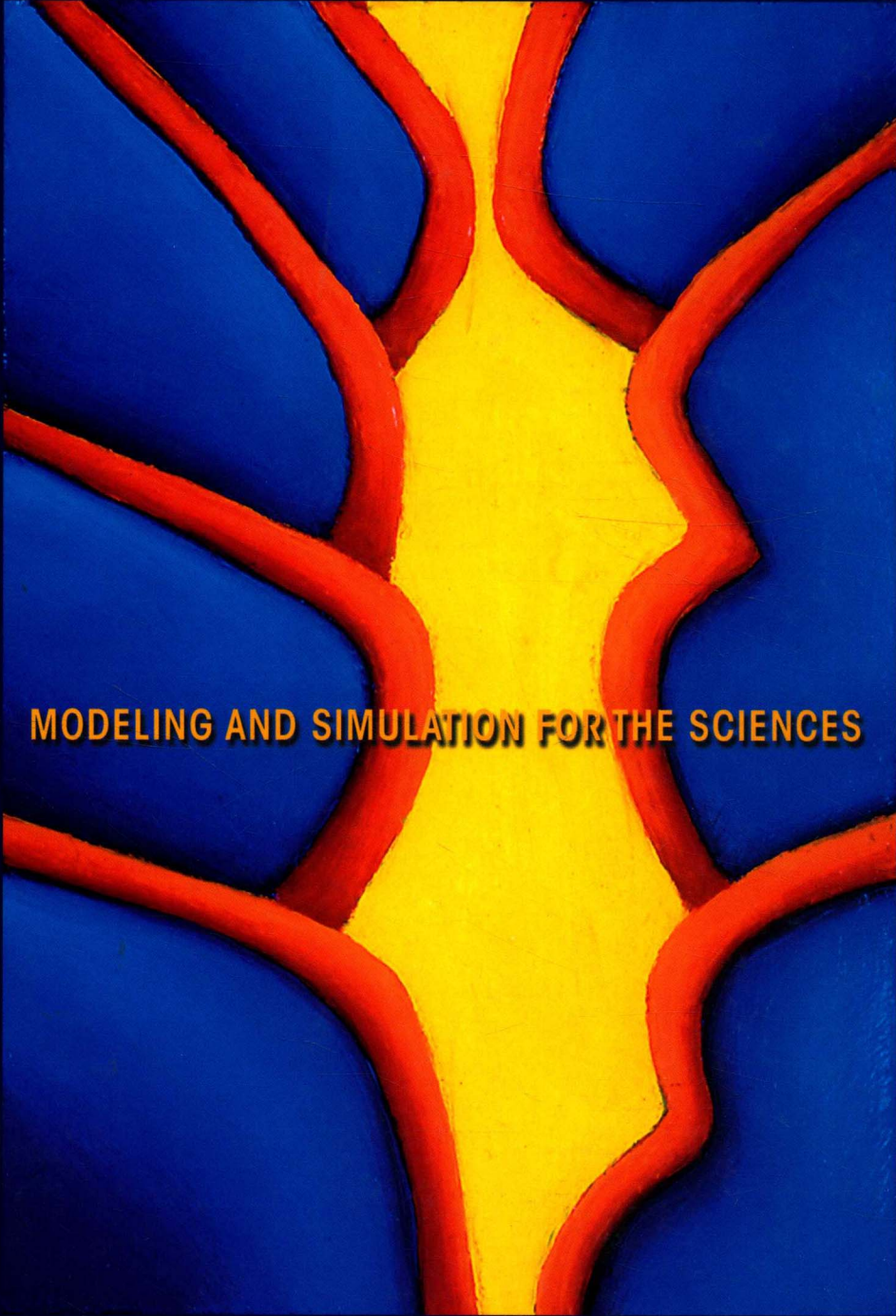


SECOND EDITION

INTRODUCTION TO COMPUTATIONAL SCIENCE



MODELING AND SIMULATION FOR THE SCIENCES

Angela B. Shiflet and George W. Shiflet

INTRODUCTION TO COMPUTATIONAL SCIENCE

MODELING AND SIMULATION
FOR THE SCIENCES

Second Edition

Angela B. Shiflet and George W. Shiflet

PRINCETON UNIVERSITY PRESS

PRINCETON AND OXFORD

Copyright © 2014 by Princeton University Press
Published by Princeton University Press, 41 William Street,
Princeton, New Jersey 08540
In the United Kingdom: Princeton University Press, 6 Oxford Street,
Woodstock, Oxfordshire OX20 1TW

press.princeton.edu

Cover art by Linda Stoudt, *Untitled 2005* (detail), 4 3/16 × 3, oil on paper

All Rights Reserved

Library of Congress Cataloging-in-Publication Data

Shiflet, Angela B.

Introduction to computational science : modeling and simulation for the sciences /
Angela B. Shiflet and George W. Shiflet. — Second edition.

pages cm

Includes bibliographical references and index.

ISBN 978-0-691-16071-9 (hardcover : alk. paper) 1. Computer sci-
ence. 2. Computer simulation. 3. Mathematical models. 4. Computational
complexity. I. Shiflet, George W. II. Title.

QA76.6.S54143 2014

004—dc23

2013024877

The epigraph in module 7.15 is © Woody Allen

British Library Cataloging-in-Publication Data is available

This book has been composed in Times Roman

Printed on acid-free paper. ∞

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

Dedicated to

Robert K. Cralle, Theodore H. Einwohner, and George A. Michael
Lawrence Livermore Laboratory
and
Robert M. Panoff
The Shodor Education Foundation,
whose friendship and guidance we have treasured

PREFACE

Although only recently thought of as an emerging field of study, computational science has now become a fundamental and integrated aspect of almost every type of scientific research. Without question, the power of our computers and computational tools has advanced rapidly, but the complexity of the problems we face and the questions we ask may seem to be more than a match. The astounding volume of data generated in our pursuit to understand ourselves and our surroundings demands all the computational power we can muster. Without scientists who can appreciate and manage this power, we will not be able to apply our computational capabilities to their best advantage. We need scientists who are able to see things from a variety of perspectives and who can talk with those from other disciplines. In fact, boundaries between disciplines are less defined than traditionally thought, and many of the most important scientific questions are at crossroads of various disciplines. For the foreseeable future, it is essential to have scientists who can work at those intersections.

At our institution, we have used this text for our course, Modeling and Simulation for the Sciences, which is one of the required courses in an Emphasis in Computational Science program. Established at Wofford in 1998, this program requires four courses besides Modeling and Simulation—Calculus I, Programming and Problem-Solving, Data Structures, and one of the following: Data and Visualization, High-Performance Computing, or Bioinformatics. Additionally, to qualify for the Emphasis, students must major in a science, mathematics, or computer science with a bachelor of science and complete an internship that employs computational methods. We designed the program and the text to be a launching pad for computationally literate scientists who will fearlessly work in interdisciplinary efforts to solve scientific problems.

As with the first edition, *Introduction to Computational Science: Modeling and Simulation for the Sciences* (2nd ed.) is designed to help the student to comprehend and exploit essential concepts of computational science, the modeling process, computer simulations, and scientific applications. Modeling and simulation are now crucial to the exploration of complex systems. These techniques enable scientists to conduct large numbers of experiments more quickly and cheaply than at the bench and to select promising research paths for their laboratories. Insights and discoveries made through modeling augment our understanding and can promote novel experimental approaches.

The **first edition** considered **three major approaches to computational science problems: system dynamics models, empirical modeling, and cellular automaton simulations**. The **second edition** includes **two additional computational approaches: agent-based simulations and modeling with matrices**.

System dynamics models afford global views of major systems that change with time. For example, such a model could include changes over time in the numbers of predator (e.g., wolf spider) and prey species (e.g., cricket). For system dynamics modeling, as with all approaches, the **text employs a nonspecific tool, or generic, approach**—to model such dynamic systems, students using the text can employ any

one of several tools, such as *STELLA*®, *Vensim*®, Personal Learning Edition (PLE) (free for personal and educational use), *Berkeley Madonna*®, *NetLogo* (free), *Python* (free), and *R* (free). With many of these tools, the student can create visual representations of models, develop relationships, run simulations, and generate graphical results.

Unlike system dynamics, **cellular automaton simulations** present local views of individuals affecting individuals. Such a world is represented as a rectangular grid of cells, with each cell having a state that can change with time according to set rules. For example, the state of one cell could represent the presence of a cricket and the state of an adjacent cell could correspond to a wolf spider. One rule could be that, when adjacent, a spider captures a cricket with a probability of 25%. Thus, on the average at the next time step, a 25% chance exists that a particular cricket next to a spider will not chirp again. The text employs a generic approach for cellular automaton simulations and scientific visualizations of the results, so that students can employ any one of a variety of computational tools, such as *Maple*®, *Mathematica*®, *MATLAB*®, *Python*, and *R*.

Agent-based simulations are similar to cellular automaton simulations. Agents, such as spiders and crickets, often reside on a grid. Each autonomous, decision-making agent has a state represented by a set of state variables and behaviors that control its actions. Moreover, at each time step, the simulation sweeps through all the agents, each updating its state based on its location and what is nearby. By contrast, a cellular automaton simulation updates the states of all the cells in a grid at each time step. With both cellular automaton and agent-based simulations, individual actions and local interactions can help us to access their effects on the whole system. Both simulation techniques can be effective in modeling dynamic, spatially complex situations. With a generic approach in the text, tutorials and files associated with the modules are available on the text's website for various agent-based tools, such as *AgentSheets* and *NetLogo*.

Simulating with agents or a cellular automaton or modeling with system dynamics, we seek to develop a computational representation that explains a situation. On the other hand, with an **empirical model**, data, such as the prevalence of crickets in an area over several years, are our only source of information about the system. We find a function that captures the trend of the data and use this mathematical model to make predictions. The same computational tools employed in cellular automaton simulations can be used for empirical models and modeling with matrices.

Matrix models incorporate certain probabilities and averages, such as the probability that a cricket exhibits a certain behavior at the next time step or the probability that it matures to the next life cycle stage and its average number of offspring. Using matrix models, we can make long-term predictions about system behaviors and populations. Moreover, we can use matrices to represent contact networks that track the simulated behavior of individuals in a variety of situations from the spread of a disease to the strength of social connections.

Modules presented in Chapter 13 on modeling with matrices include sections that discuss the utility of **high-performance computing** (HPC) in computational science. Moreover, Chapter 12 provides an introduction to some of the hardware and algorithms for HPC, which is becoming increasingly important with the advent of "big data" and massive computational problems. The text's website includes additional related material and programs in C and MPI.

Tutorials, package-specific Quick Review Questions with answers, and files to accompany the text material are **available from the text's website in various system dynamics tools** (including *STELLA*®, *Vensim*®, *Berkeley Madonna*®, *NetLogo*, *Python*, and *R*), **in several computational tools** (including *Maple*®, *Mathematica*®, *MATLAB*®, *Python*, and *R*) and **in assorted agent-based tools** (including *Agent-Sheets* and *NetLogo*). Typically, an instructor selects one system dynamics tool, one computational tool, and one agent-based tool, depending on the types of models covered, for class use during the term. The variety of tools and examples of their use employed in this text provides an instructor with many options. Note that **at least one tool from each type is available as open source**.

Text Prerequisites

Prerequisites for *Introduction to Computational Science* are minimal. The text **does not require computer-programming experience**. Although the concept of rate of change, or derivative, from a first course in calculus is used throughout the text, the necessary background is contained in Module 2.2, "Unconstrained Growth and Decay." Otherwise, students do not need to know how to take derivatives in order to understand the material or develop the models. For those who would like additional discussions of the material, two modules on fundamental calculus concepts, "Rate of Change" and "Fundamental Concepts of Integral Calculus," are available on the text's website.

Learning Features

While the interdisciplinary nature of computational science is distinctive, it is also challenging, particularly for students with limited experience in computer science, mathematics, and various areas of the sciences. To mitigate this challenge, the text provides the background that is necessary for the student to understand the material, work the problems/projects, and confidently succeed in the course. Each module involving a scientific application **covers the prerequisite science without overwhelming the reader** with excessive detail. Furthermore, the text provides **a wide variety of application areas** for examples, exercises, and projects, including astronomy, biology, chemistry, economics, engineering, environmental science, finance, geology, medicine, physics, psychology, and the social sciences.

Introduction to Computational Science has chapters that consist of several modules each. The text's website contains **two tutorials on system dynamics tools, seven tutorials on computational tools, and two tutorials on agent-based tools**. The text presents the **tutorials in a just-in-time fashion**, covering the requisite information for the subsequent material.

Module 1.2 introduces the **modeling process**, and the text consistently uses the process to guide the user through numerous scientific examples. For instance, after covering the prerequisite scientific background, Module 4.4 develops a model of malaria by following the modeling process in a step-by-step fashion. Also, Module

9.2 introduces the **development of computer simulations** along with their utilities in the sciences. Thus, the text is structured to help students to learn how modelers model and to develop their own modeling skills.

The text presents material in a clear manner with generous use of examples and figures. Most sections of a module end with **Quick Review Questions** that provide an assessment of the student's comprehension of the material, and the text includes more than 230 such questions, averaging about three parts each. **Answers**, often with explanations, at the end of the module give immediate feedback and reinforcement to the student. In the case of system dynamics, computational, or agent-based tool-dependent questions, the questions and answers are on the text's website in PDF files for a variety of tools, as appropriate.

To further aid in understanding the material, most modules include a number of **exercises** (about 275 in the text) that correlate directly to the material and that the student is to complete for the most part with pencil and paper and occasionally with a computer. Answers to selected problems, whose exercise numbers are in color, appear in an appendix.

A subsequent **"Projects"** section provides numerous project assignments for students to develop individually or in teams. The text contains approximately 650 projects. While a module, such as "Modeling Malaria," might develop one model for an application area, the projects section suggests many other refinements, approaches, and applications. The ability to work well with an interdisciplinary team is important for a computational scientist. **Chapters 7 and 14 provide modules of additional, substantial projects** from a variety of scientific areas that are particularly appropriate for teams of students. These modules indicate prerequisite text material, and the projects sections of earlier modules refer to appropriate projects from Chapters 7 and 14 that the instructor can assign at that point in the term.

A **"References"** section occurs at the end of most modules. It provides a list of hyperlinks, books, and articles for further study.

A **glossary** of scientific, modeling, and simulation terms is provided for quick reference. The text's website provides links to **downloadable tutorials, models, PDF files, and datasets** for various tool-dependent quick review questions and answers, examples, and projects.

Using the Material

Because the area is emerging, a variety of departments offer introductory computational science courses, and instructors approach the material in diverse ways. The first edition has been used at various levels (undergraduate, graduate, and high school) and in different types of courses. The second edition provides two additional modeling approaches, agent-based and matrix modeling, and more than enough material so that an instructor can select one or more of the methodologies. Thus, *Introduction to Computational Science* provides **several pathways through the material**. For example, one could choose to start a course in any of a variety of places—with an overview of the modeling process (Chapter 1) and system dynamics modeling (Chapters 2–4); Chapter 8, "Data-Driven Models"; Chapter 9, "Monte Carlo Simulations"; Chapter 11, "Agent-Based Modeling"; or Chapter 13, "Matrix

Models.” Moreover, Chapter 5, “Computational Error,” can be covered or not covered any time during the course. The same options are true for Chapter 6, “Simulation Techniques,” after consideration of Module 2.2. For those who wish to cover Chapter 13 before (or instead of) cellular automaton simulations, a tutorial for various computational tools, primarily covering matrix operations, is available on the text’s website. One possible course outline is described shortly, and the website gives alternative syllabi and a prerequisite structure for the modules. Moreover, the text provides an abundance of discipline-specific applications, so that the text is suitable either for an **introductory course generally in computational science or**, with appropriate selection of applications, **specifically in computational applications for biology**.

The text begins with an introduction to computational science and the modeling process. Chapter 2 commences with a discussion of system dynamics and models where the rate of change of a quantity is proportional to that quantity. Two tutorials available in a choice of several tools lead the student step by step through the process of implementing a model with the software. “Unconstrained Growth and Decay” discusses models that exhibit exponential growth or decay and introduces concepts of time-driven simulations. The module also develops the analytical solution to unconstrained growth and decay problems for students who have had integral calculus and for those who have not. The module “Constrained Growth” considers situations in which the quantity under change, such as a population, has a maximum value, or carrying capacity. In this context, we introduce the concepts of equilibrium and stability. The module “Drug Dosage,” which includes geometric series, provides other examples where rate is proportional to amount.

For those interested in physics models, Chapter 3, “Force and Motion,” provides modules on falling and skydiving, bungee jumping (springs), pendulum clocks, and rocket motion. However, the instructor can choose to skip this chapter or to assign its modules and projects for work outside of class.

Numerous system dynamics models involve interactions, such as with population dynamics or chemical reactions. Chapter 4 considers such models with discussions of competition, predator-prey models, spread of SARS, modeling malaria, and enzyme kinetics.

With computational estimates, the modeler should always be aware of sources of computational error. Thus, after a beginning tutorial on a tool we can use for computation, empirical models, and cellular automaton simulations (tutorial versions on the text’s website), Chapter 5, “Computational Error,” contains a module “Errors.” However, an instructor can delay coverage of the first two computational tool tutorials until considering material from Chapters 8–10.

After a second computational tool tutorial from the text’s website, Chapter 6 covers three simulation techniques: Euler’s, Runge-Kutta 2 (Euler’s Predictor-Corrector), and Runge-Kutta 4. One or more of these techniques can be covered at any time after Chapter 2’s module “Unconstrained Growth and Decay.” For example, the instructor may choose to discuss Euler’s Method immediately after that module and delay consideration of the other two techniques until later in the term.

Chapter 7 provides opportunities for students to learn system dynamics modeling by completing additional extensive projects. Unlike earlier chapters, the modules of this chapter do not include examples. Instead, each module contains sufficient background in a scientific application area for students to develop their own system dy-

namics models, as suggested by project descriptions. Each module lists the prerequisite material so that students can do Chapter 7's projects at any time after covering the earlier material. These projects and some of the more extensive projects in previous chapters provide excellent opportunities for teamwork. The chapter includes the following topics: radioactive chains, blood cell populations, scuba diving, carbon cycle, global warming, growth of a garden, cardiovascular system, electrical circuits, transmission of nerve impulses, antibiotic resistance, carbohydrate metabolism, mercury pollution, economics of commercial fishing, biochemical pathways, and colon cancer.

Chapter 8 shifts away from system dynamics modeling. After a third tutorial on a computational tool, an optional tutorial covers functions that often appear in modeling. With this background, empirical models, which are based only on data and are used to predict and not to explain a system, are considered.

Monte Carlo simulations of Chapter 9 form the basis for modules in this and the next chapter. After an appropriate tutorial, the chapter considers simulation development and area estimation using this technique. An instructor interested in doing so can also cover how to generate random numbers in other probability distributions for computer simulations and details of the multiplicative linear congruential method to generate uniformly distributed random numbers. After an additional computational tutorial, the chapter concludes with the random walk method that occurs in numerous computer simulations.

Chapter 10 considers many applications of cellular automata. A computational tool tutorial leads into a module involving applications, such as spreading heat through an iron bar, as well as fundamental concepts, such as periodic boundary conditions. An instructor can choose to cover one or more additional in-depth cellular automaton simulations on spreading of fire, movement of ants, or growth of biofilms.

On the text's website, two tutorials in various tools are associated with Chapter 11 on agent-based modeling. After the first tutorial, the student can consider the spread of disease in beef cattle as the animals reside on and travel to various locations. Movement of an invasive species, the cane toad, from one artificial watering point to another and attempts to contain their spread can be modeled in a subsequent module after completing a second tutorial.

Some modeling and simulation projects require massive computational power beyond the capabilities of present-day sequential computers. Thus, Chapter 12 provides an introduction to high-performance computing (HPC). The chapter covers the basic concepts and hardware configurations of HPC as well as some parallel-processing algorithms. With this background, the student can gain an appreciation of some of HPC's potential and challenges.

The utility of HPC is discussed in several sections of the following chapter, "Matrix Models." After a computational tool tutorial involving matrix operations, Module 13.2 covers various vector and matrix operations in the context of population applications. An instructor can then cover one or several applications in any order. When we can divide the life of an organism into ages/stages, Module 13.3 shows how we can often employ age-structured/stage-structured models to determine intrinsic growth rates (eigenvalues), stable distributions, threats of extinction, and how sensitive the long-term population growth rates and predicted times of extinction are to small changes in parameters. Covering the necessary background in probability

theory, Module 13.4 develops a Markov chain model, which employs the probability of passing from one state to another and can solve a large variety of problems from predicting the behavior of animals to examining forest succession to locating genes in DNA. The last module covers individual-based (or network-based) epidemiology simulations that track the simulated behavior of individuals. Such a model involves a contact network, and the module covers the basics of graph theory, implementation of graphs with matrices, and characteristics of social networks.

As with Chapter 7, Chapter 14 provides opportunities for students, perhaps in teams, to enhance their computational science problem-solving abilities through completion of additional extensive projects that they can do at any time after covering prerequisite material. The modules do not have examples but do have sufficient scientific background for the projects. The applications of computational science empirical models, random walks, cellular automaton simulations, agent-based simulations, and modeling with matrices in Chapter 14 include the following: polymers, solidification, foraging, pit vipers and heat diffusion, mushroom fairy rings, spread of disease, HIV in the body, predator-prey relationships, clouds, fish schooling, invasive plants, numerous cellular automaton simulations originally considered with agent-based models, and vice versa, bioinformatics, and social science networks.

A Possible Course Outline

As with most courses, an instructor is likely to vary material coverage from term to term. Alternative approaches are on the text's website. With project assignments encompassing one to two weeks, the following is a possible schedule covering system dynamics and empirical modeling and cellular automaton simulations:

- Week 1: Chapter 1, 2.1 (System Dynamics Tutorial 1) in a lab if possible, start 2.2 (project assignments from module)
- Week 2: 2.2, 2.3, assign 2.4 (System Dynamics Tutorial 2)
- Week 3: 2.5; project assignments from 2.5 or Chapter 3 and 7.1–7.6; 4.1, 4.2
- Week 4: 4.3, project work in lab if possible, project assignments from 7.7–7.15
- Week 5: 5.2 (exercises assigned to be graded), 6.2, 6.3
- Week 6: In lab if possible—project work, 5.1 (Computational Tool Tutorial 1), and 6.2 (Computational Tool Tutorial 2)
- Week 7: Project presentations, midterm
- Week 8: Assign 8.1 (Computational Tool Tutorial 3), 8.3 (project assignments from module), review assignment on Computational Tool Tutorials 1–3
- Week 9: Assign 9.1 and 9.5 (Computational Tool Tutorials 4 and 5), 9.2, 9.6, project assignments from 14.1 and 14.2
- Week 10: Work on projects, assign 10.1 (Computational Tool Tutorial 6)
- Week 11: 10.2, 10.3, project assignments from 14.3–14.12
- Week 12: 10.3, work on projects
- Week 13: 12.1, 12.2
- Week 14: Project work and presentations
- Week 15: Final exam

Students have found it helpful to have short quizzes taken directly from the Quick Review Questions. Other graded assignments include tutorials, selected exercises, projects, a midterm, and a final exam.

Supplementary Materials

Instructors and students can link to the text's website through the Princeton University Press' website (<http://pup.princeton.edu/>) or Wofford College's Computational Science website (<http://www.wofford.edu/ecs/>). The following **resources** are available on the text's site:

- Two system dynamics tool tutorials in several tools, such as *STELLA*, *Vensim PLE*, *Berkeley Madonna*, *Python*, and *R*
- Seven computational toolbox tutorials in several tools, such as *Maple*®, *Mathematica*®, *MATLAB*®, *Python*, and *R*
- For a variety of tools, PDF files that contain system-dependent Quick Review Questions and answers
- In a variety of tools, files that contain models, as indicated in the "Download" sections of modules
- Datasets
- References with links to other websites

The text's website also has an **online *Instructor's Manual***, which contains the following material:

- Solutions to all exercises in the text
- Solutions to the two system dynamics tool tutorials using several tools, such as *STELLA*®, *Vensim PLE*®, *Berkeley Madonna*®, *NetLogo*, *Python*, and *R*
- Solutions to the seven computational toolbox tutorials and a function tutorial using several tools, such as *Maple*®, *Mathematica*®, *MATLAB*®, *Python*, and *R*
- Solutions to the two agent-based tool tutorials using several tools, such as *NetLogo* and *AgentSheets*
- Test problems with answers
- Model solutions and suggested approaches for selected projects
- *PowerPoint* files of key figures and algorithms
- Suggested pathways through the material
- Prerequisite structure of the modules

Instructors who adopt the text may obtain a password from Princeton University Press to access the online *Instructor's Manual*.

Acknowledgments

The first edition of this text was published in 2006, and much in the world and our lives has changed. We are still eternally grateful for the support and encouragement

of our colleagues and students, but we have also had enormously important input from many external users of the book. The development of the second edition has been no less a labor of love than the first, but we have written it with significantly more experience professionally. Colleagues at Wofford College have been especially accommodating in their willingness to class-test many of the new modules and to provide us with extraordinarily constructive suggestions for improvement. In particular, we would like to thank Drs. Anne J. Catllá, Joseph D. Sloan, and David A. Sykes. Drs. Catllá and Sloan used the first edition and new materials in their modeling and simulation classes, and Dr. Sykes meticulously reviewed most of the new modules and projects. Drs. Ted Monroe and Joseph Spivey also gave very helpful feedback on several of the new modules they utilized in their upper-level mathematics courses.

We are privileged to teach in an institution that attracts a remarkably talented group of science, computer science, and mathematics students, several of whom have contributed directly and indirectly to the development of various modules and associated files. We would like to recognize particularly Mayfield Reynolds, Shay Ellison, Jesse Hanley, and Whitney Sanders for their dedication, creativity, and diligence.

Additionally, we would like to acknowledge the generous comments and contributions, including R files, from Dr. Stephen Davies, computer scientist at Mary Washington University. Dr. Davies has not only shared excellent feedback for the first edition, but he has contributed directly to this edition by allowing us to incorporate an interesting module he developed (Module 7.6, “Plotting the Future—How Will Your Garden Grow”) and a clever Project 10 on paratroopers in Module 3.1, “Modeling Falling and Skydiving.”

This revised text has been significantly improved by experience we have gained from teaching with the material in various courses and through the sabbatical leave we were granted by Wofford College in 2010–11. Dr. David Wood, Senior Vice-President and Academic Dean at Wofford, has been very understanding of our work and the need for time to do research and writing. He and the college Committee on Non-Curricular Faculty Concerns were most munificent in accommodating this leave. The experiences we gained during the year changed our personal and professional lives immeasurably, and they continue to inspire us in our teaching, research, and writing.

We are unable to stress adequately the value of our time working abroad to the development of this work. For five months we worked with some extremely talented scholars of the Computational Biology Group, the University of Oxford, and we would like to acknowledge the graciousness of Professor David Gavaghan, who agreed to host us. Our focus at Oxford was the modeling of colon cancer, as part of a subgroup concentrating on projects in soft tissue mechanics and cancer. We worked most closely with a then-D.Phil. student in mathematics, Ornella Cominetti, who delighted us with her brilliance, her enthusiasm, and her wonderful sense of humor. The time there broadened our experience in modeling and programming (with *MATLAB*[®] and *Chaste*[®]).

This yearlong adventure was really enabled by a computer scientist we met at a meeting of the International Conference on Computational Science in Krakow, Poland, in 2008. Professor David Abramson, then of Monash University in Melbourne and more recently of the University of Queensland, Australia, was undoubtedly surprised by our queries about sabbatical positions in Australia and England. Thank-

fully, he agreed to help us, even providing the essential contact at Oxford. David gave us the opportunity to work within the eResearch and Grid Engineering Laboratory at Monash, where we had five months to do research and write. He encouraged us to present two workshops for graduate students and faculty in computational science at Monash, which provided us with invaluable experiences. Subsequently, one of the participants in the workshop, Dr. Valerie Maxville, who leads the Education Program at iVEC, Western Australia's supercomputing and eResearch facility, invited us to give a talk and workshop in Perth. Hopefully, our efforts have encouraged more computational applications at Monash and other Australian institutions.

We are indebted to the following reviewers, who offered many valuable constructive criticisms as we prepared the first edition:

Rob Cole, Evergreen State University
 Richard Hull, Lenoir-Rhyne College
 James Noyes, Wittenberg College
 Bob Panoff, The Shodor Foundation
 Sylvia Pulliam, Western Kentucky University
 Joseph Sloan, Wofford College
 Chuck Swanson, University of Minnesota
 David Sykes, Wofford College
 Peter Turner, Clarkson University
 Ignatios Vakalis, Capital University

Second Edition Table of Contents and/or Module Reviewers:

Anne Catllá, Wofford College
 Melanie Correll, University of Florida at Gainesville
 Stephen Davies, Mary Washington University
 Charles Epstein, University of Pennsylvania
 David Joiner, Kean University
 Hong Liu, Embry-Riddle
 Richard Salter, Oberlin College
 Joseph Sloan, Wofford College
 David Sykes, Wofford College
 Carlo Tomasi, Duke University

Vickie Kearn, Senior Editor at Princeton University Press, unfailingly supported us in the preparation of the first edition, which was a novel project in an emerging discipline. She has always had a clear understanding of the project and provided excellent guidance. In the same spirit, Vickie has been enthusiastic in the revision of that project. Her vision and trust have been remarkable, and we reiterate our gratitude for all that she has done to facilitate our latest efforts.

We are so grateful to the extraordinarily capable team at Princeton University Press who helped craft the second edition of this text. Quinn Fusting was always our "go to" person for answers and logistics. Debbie Tegarden very ably guided production, encouraging everyone else in the process. We were delighted to work again with Dimitri Karetnikov, who continues to transform mere figures into art, and Lorraine Doneker, who fashioned another attractive design. We were so fortunate to have Bytheway Publishing Services, directed by Lori Holland, working so carefully and capably again on the typesetting and composition of this very technical and

complex project. Thanks also go to Linda Thompson, who skillfully copyedited this edition.

Linda Stoudt, George's cousin, has been so generous in allowing us to display some of her talent on the covers of both editions of our text. Everything that she does is done with loving care and powerful intellect.

There is no person more responsible for our writing this text than Dr. Robert M. Panoff of the Shodor Education Foundation/National Computational Science Institute. From the first Shodor workshop we attended to the present, Bob has been the source—ideas, encouragement and support. Bob has helped us in countless ways, including traveling to Australia to make the first workshop on *computational thinking* a resounding success. Moreover, the NSF-funded Blue Waters Undergraduate Petascale Education Program, for which he was a co-PI, funded development of six educational modules, which were preliminary versions of modules in this second edition, and internships for three students, who assisted us with HPC materials. He is passionate about computational science and computational science education, and he is enormously generous with his time and ideas. We are indebted to him for our professional transformation that has facilitated the development of the computational science program at Wofford and the creation of this book.

Our parents Isabell and Carroll Buzzett and Douglas and George Shiflet, Sr., worked so tirelessly to make our lives intense, joyful, and stimulating. We were blessed and inspired by their living efforts, and we miss them. Hopefully, they would be pleased by what we have been doing with our lives.

CONTENTS

Preface

xxiii

1 OVERVIEW

Module 1.1 Overview of Computational Science	3
Projects	6
References	6
Module 1.2 The Modeling Process	7
Introduction	7
Model Classifications	8
Steps of the Modeling Process	9
Exercises	11
References	12

2 SYSTEM DYNAMICS PROBLEMS WITH RATE PROPORTIONAL TO AMOUNT

Module 2.1 System Dynamics Tool—Tutorial 1	15
Download	15
Introduction	15
Module 2.2 Unconstrained Growth and Decay	17
Introduction	17
Rate of Change	17
Differential Equation	19
Difference Equation	20
Simulation Program	24
Analytical Solution: Introduction	25
Analytical Solution: Explanation with Indefinite Integrals (Optional)	26
Analytical Solution: Explanation with Derivatives (Optional)	26
Completion of the Analytical Solution	27
Further Refinement	28
Unconstrained Decay	28
Reports for System Dynamics Models	30
Exercises	31
Projects	32
Answers to Quick Review Questions	33
Reference	33
Module 2.3 Constrained Growth	34
Introduction	34
Carrying Capacity	35
Revised Model	36
Equilibrium and Stability	39

Exercises	39
Projects	41
Answers to Quick Review Questions	42
References	43
Module 2.4 System Dynamics Tool—Tutorial 2	44
Download	44
Introduction	44
Module 2.5 Drug Dosage	45
Downloads	45
Introduction	45
One-Compartment Model of Single Dose	46
One-Compartment Model of Repeated Doses	48
Mathematics of Repeated Doses	50
Sum of Finite Geometric Series	53
Two-Compartment Model	54
Exercises	54
Projects	55
Answers to Quick Review Questions	56
References	57
3 FORCE AND MOTION	
Module 3.1 Modeling Falling and Skydiving	61
Downloads	61
Introduction	61
Acceleration, Velocity, and Position	62
Physics Background	65
Friction during Fall	68
Modeling a Skydive	70
Assessment of the Skydive Model	72
Exercises	72
Projects	73
Answers to Quick Review Questions	76
References	77
Module 3.2 Modeling Bungee Jumping	78
Downloads	78
Introduction	78
Physics Background	79
Vertical Springs	81
Modeling a Bungee Jump	84
Exercises	86
Projects	87
Answers to Quick Review Questions	88
References	88
Module 3.3 Tick Tock—The Pendulum Clock	90
Download	90
Introduction	90