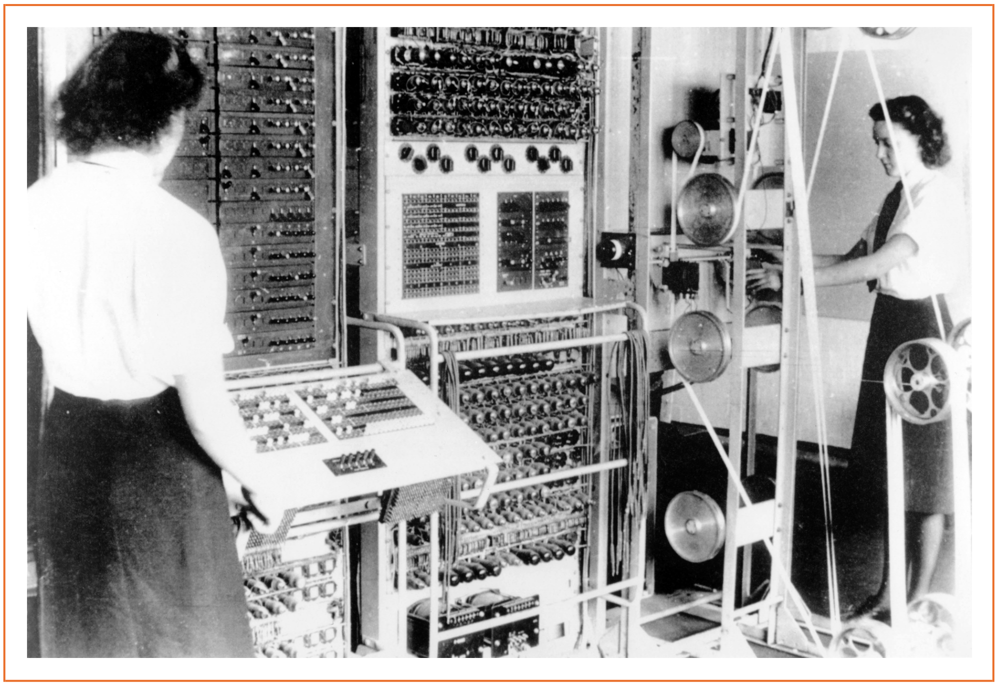
INTRODUCTION TO COMPUTATIONAL PHYSICAL CHEMISTRY



The Colossus was the first programmable electronic digital computer, developed by British code breakers during the Second World War. The first prototype was built in December 1943 and deciphered its first message in February 1944. By the end of the war, ten machines had been built.

The slanted control panel on the left sets the configuration of the decryption algorithm. The message to be decoded was input using the paper tapes shown to the right; these tapes moved at 27 miles per hour, for an input rate of 5000 characters per second. Calculations were performed using 2400 vacuum tubes, and output shown on a panel of indicator lamps. The total power usage was approximately 8.5 kW. The machines were operated by the Women's Royal Naval Service (nicknamed "Wrens"), who worked around the clock repairing the machines and processing the results.

After the war, all the Colossus machines were dismantled, all the documentation was destroyed, and the project was kept a secret until the 1970s. A functioning replica was reconstructed in 2007, based on information from surviving laboratory notebooks, and is on display at the National Museum of Computing at Bletchley Park, England.

Introduction to Computational Physical Chemistry

JOSHUA SCHRIER

Haverford College

UNIVERSITY SCIENCE BOOKS
MILL VALLEY, CALIFORNIA

University Science Books
Mill Valley, California
www.uscibooks.com

EDITOR Jane Ellis
PRODUCTION MANAGEMENT Paul C. Anagnostopoulos, Windfall Software
MANUSCRIPT EDITING & PROOFREADING Jennifer McClain
INTERIOR & COVER DESIGN Yvonne Tsang, Wilsted & Taylor
COMPOSITION Windfall Software, using ZzTEX
ILLUSTRATIONS Laurel Muller, Cohographics
PRINTING & BINDING

This book is printed on acid-free paper. ♾

Print ISBN: 978-1-938787-90-4
eBook ISBN: 978-1-938787-91-1

Library of Congress Cataloging-in-Publication Data

Names: Schrier, Joshua, 1979–
Title: Introduction to computational physical chemistry / Joshua Schrier, Haverford College.
Description: Mill Valley, California : University Science Books, [2017] | Includes index.
Identifiers: LCCN 2017010818 (print) | LCCN 2017010486 (ebook) | ISBN 9781938787904 (alk. paper) | ISBN 9781938787911 ()
Subjects: LCSH: Chemistry—Mathematics—Textbooks. | Chemistry, Physical and theoretical—Textbooks. | Chemometrics—Textbooks.
Classification: LCC QD39.3.M3 S327 2017 (ebook) | LCC QD39.3.M3 (print) | DDC 541.01/51—dc23
LC record available at https://lccn.loc.gov/2017010818

Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

Behold, only this have I found out:
God made mankind straight, but men
have had recourse to many calculations.
—Ecclesiastes 7:29

Science can transform this earth into heaven.
But it can do so only in combination with
non-violence. If science is yoked to violence,
the world will be shattered to bits."
—Vinoba Bhave

BRIEF CONTENTS

PART I. QUANTUM MECHANICS

PART II. THERMODYNAMICS AND KINETICS

PART III. RESOURCES

CONTENTS

PART I. QUANTUM MECHANICS

4. Variational Method 67

5. Hartree-Fock Self-Consistent Field 97

6. Hückel Molecular Orbital Theory 139

7. Quantum Theory of Solids 179

8. Density Functional Theory 203

9. Quantum Monte Carlo 221

PART II. THERMODYNAMICS AND KINETICS

10. Classical Gas Laws 255

11. The Metropolis Monte Carlo Method 271

PART III. RESOURCES

PREFACE

WHY STUDY COMPUTATIONAL CHEMISTRY?

I think of *chemistry* as any field of study where one cares about the atomistic details, so this broadly includes molecular biology and nanoscience, (in the spirit of G. N. Lewis's definition that "physical chemistry is anything which is interesting"). There are essentially two-and-a-half big ideas that explain all of chemistry: quantum mechanics, thermodynamics, and kinetics (which is only "half" an idea, since it can in principle be derived from the first two). Computational chemistry is a branch of physical chemistry that uses numerical calculations to implement quantum mechanical, thermodynamic, and kinetic theories in order to understand and predict chemical phenomena. The computational approach is *necessary* because these theories become too complicated to solve exactly for experimentally relevant systems. The computational approach is *useful* because it allows us to perform "virtual experiments" where we can track every atom in the system (try that in the laboratory!), study molecules that have not yet been synthesized, or turn off interactions to study their effects. Due to the rapid advances in computer technology it is now possible to calculate the properties of molecular systems accurately using these methods, and computational chemistry has established itself as an essential part of modern research.

WHY THIS BOOK?

The purpose of this book is to provide a bridge between the traditional "solve a bunch of equations exactly for a very simplified model approach" and the practical computational methods that can be used to solve research problems. While the more recent textbooks include exercises using packaged Hartree-Fock/DFT calculations, this is too much of a black box for my tastes. I want my students to be able to participate meaningfully in my research, and I am a firm believer that one only understands what one can build. The expedient (though sometimes unpolished) do-it-yourself (DIY) approach taken in this book is aimed at providing a sort of "MAKE:" magazine approach to computational chemistry. Consequently, this book does not aim to replace any of the (many) textbooks on the chemical applications of thermodynamics and quantum mechanics to chemical problems. On the contrary, it is assumed that the student has already read the relevant portions of a more comprehensive traditional textbook, so that we can proceed without rewriting the basic proofs or redefining standard terminology.

The reader of this book should come away with the ability to apply and adapt these techniques to her own research problems, and to have a greater ability to critically evaluate computational results in the literature that use these methods. The "Looking Forward" sections in each chapter connect the topics and simulations discussed here to broader scientific problems and available (mostly free/open-source) software packages that expand upon these techniques to perform even more sophisticated simulations.

WHO IS THIS BOOK FOR?

This book has several possible audiences: undergraduate physical chemistry students, upper-level undergraduate or lower-level graduate computational chemistry students, and "independent" researchers at all levels. I have integrated the majority of this material into the standard two-semester junior-level undergraduate physical chemistry curriculum using the McQuarrie and Simon series published by University Science Books. The "To the Instructor" notes provide details about how to do this. Admittedly, this makes for a rather intensive course that places heavy demands on the students. One could also use this book in a stand-alone semester-long upper-level undergraduate or lower-level graduate course in computational chemistry methods. It is also accessible for self-study by, for example, a new research student who is beginning a computational project, a "professional" scientist who wants to learn about what is actually going on in the theoretical calculations her collaborators produce, or an "amateur" scientist who wants to try this at home. I use isolated chapters as tutorials for new undergraduate research students, and my colleagues teaching Computational Physics and Computer Science courses have used isolated chapters as the foundation for end-of-semester projects.

WHY MATHEMATICA?

My first attempts at teaching this material were in Fortran; the student response was not encouraging. I have settled on Mathematica for the pragmatic reasons that my students have previous exposure to it in their calculus classes, and that it is simple to produce nice-looking figures. Moreover, there are many useful data sources in Mathematica that simplify carrying out nontrivial calculations. However, it has been said that one can write Fortran in any language, and the discerning reader will detect this in my stylistic choices throughout the text. All of the numerical calculations may be readily translated into Fortran or C (provided one has an eigenvalue package such as EISPACK or ARPACK), or Perl/Python (provided one has SciPy), or any other mathematics package, such as MATLAB or Octave. The only nongeneralizable functionality is the use of `ChemicalData[]` in the section on gas laws (to obtain various gas constants) and Hückel theory (to obtain molecular structures and connectivity matrices). While these features are program dependent, it seemed a shame not to use this functionality. The reader, however, will be able to substitute for this by doing things the "old-fashioned way," i.e., looking up this data in reference tables and constructing the Hückel matrices by hand.

My general philosophy is to illustrate the general principles and the essential tricks, while limiting complicated optimizations. In situations where "easy to understand" and "efficient" are mutually exclusive, I have chosen the former, since many students enter the course with no programming background. There are a few topics where simple algorithmic changes can speed up the calculations dramatically, but I have tried to encapsulate these discussions into their own sections of the chapter, so that they can be safely ignored. (The `Timings[]` results were all performed on a 2012-vintage MacBook Air, so you will find everything faster.)

ACKNOWLEDGMENTS

The premise of this book was born during afternoon cappuccinos at Caffe Strada with Michele Ceotto (Universitá degli Studi di Milano) and Albert Pan (D. E. Shaw Research), while we were all graduate students at the University of California, Berkeley. The topics and models chosen here are an outgrowth of those conversations, and remembering those warm California afternoons has been an encouragement during the cold winter mornings that I spent writing this book. The actual substance of this book developed out of several iterations of the year-long undergraduate junior-level physical chemistry course that I teach at Haverford College. Without the students, I would not have had the motivation, nor the feedback, necessary to produce the material you have before you. I thank all of these students for their good-natured patience with the early iterations of the book, and the ongoing demands that this course puts upon them. I thank Nicholas Jackson (Argonne National Laboratory) and the first round of anonymous reviewers of University Science Books for their careful reading and thoughtful comments of an early version of the manuscript, and Casey Londergan (Haverford), Elizabeth Young (Amherst), and Alán Aspuru-Guzik (Harvard) for testing portions of the manuscript in their courses and for providing valuable feedback about problems. I also thank Robert Berger (Western Washington), Eric Bittner (Univ. of Houston), Robert Hanson (St. Olaf's), Patrick Huang (Cal. State–East Bay), Mark Marshall (Amherst), and George Shields (Furman) for their thorough reviews of the penultimate manuscript. Mervin Hanson (Humboldt State) deserves a double portion of thanks for providing line-by-line critical reviews of two versions of the manuscript.

Turning a manuscript into a book is a craft. I thank Jakub Kuczmarski (Univ. of Warsaw) for developing the open-source mmacells (http://github.com/jkuczm/mmacells) and MathematicaCellsToTeX (http://github.com/jkuczm/MathematicaCellsToTeX) packages that allowed me to typeset the code in a pleasing way, Paul Anagnostopoulos (Windfall Software) for his patient assistance in turning this code into print, the team at Wilsted and Taylor Publishing Services for their thoughtful attention to the design and presentation of this work, and Bruce Armbruster and Jane Ellis (University Science Books) for their encouragement during this process.

ERRORS

As Tai T'ung wrote in his thirteenth-century history of Chinese writing, *Liu Shu Ku (The History of the Six Writings)*, “Were I to await perfection, my book would never be finished.” I welcome comments, corrections, and criticisms that will lead to improvements. Please send them to jschrier@haverford.edu. A list of errata is available at http://www.uscibooks.com/schrier_errata.htm.

TO THE INSTRUCTOR

The general arrangement of the material (and in particular the statistical approach to thermodynamics) is now common in undergraduate physical chemistry textbooks (e.g., McQuarrie and Simon, Atkins and de Paula, Engel and Reid). In my own teaching, I use the *Quantum Chemistry* and *Molecular Thermodynamics* textbooks by McQuarrie and Simon published by University Science Books, and for various institutional reasons, most students at Haverford take thermodynamics first and quantum chemistry second. This is just the opposite of the order of presentation in McQuarrie and Simon's *Physical Chemistry: A Molecular Approach*. To accommdate either possible ordering, the two parts of this book are written as self-contained entities, and readers can begin with either portion. This necessitates some gentle repetition in the discussion of numerical versus symbolic operations, list processing, etc., in the text, which most students find helpful.

USER'S GUIDE WHEN TEACHING QUANTUM MECHANICS

I use McQuarrie's *Quantum Chemistry*, 2nd ed., from University Science Books, but the discussion below can be generalized to any introductory quantum chemistry textbook. The class meets twice a week for 90 minutes during a 14-week semester.

McQuarrie Chapter 1 treats the "Dawn of Quantum Theory," and the problems at the end of this chapter are an appropriate place to introduce this book's appendix on data analysis (Appendix B). I then spend a class on reviewing mathematical topics (using McQuarrie's MathChapters A, B, C, and G) and to introduce Mathematica (using this book's Appendix A on computer programming).

I find it helpful to introduce the particle-in-a-box (PIB) problem alongside the computational approach (Chapter 1). One class introduces the formal aspects of the 1D-PIB problem (McQuarrie 3.1–3.8) alongside the basic computational methods (here Sections 1.1–1.2). The next class generalizes this to the 2D and 3D PIB (McQuarrie 3.9, here Sections 1.3–1.4). Section 1.5 here uses the language of measurement; unlike McQuarrie, I like to introduce this earlier in my course, and so I include it during the class on the 2D and 3D PIB. However, this topic could also be deferred until McQuarrie Chapter 4, where it might leaven an otherwise dense formal treatment. Finally, the time-dependent Schrödinger equation (McQuarrie 4.8–4.10) can also be presented alongside the corresponding computational work (Section 1.6).

Next, I spend one class on the finite difference method (Chapter 2), which is introduced in connection with the particle-in-a-box and finite well problem. I have found that this reinforces the idea of testing new algorithms on problems with known solutions—or as I like to call them, "sanity checks" that the code is doing what one thinks it is. I then have the students solve the harmonic oscillator problem (including calculations of the dependence of energy on the mass and spring constant) using this numerical approach (Problem 2-4). In this way, the students start building an intuition about this problem numerically before they encounter its analytical solution in McQuarrie Chapter 5. The numerical exercise demystifies the simple fact that the harmonic oscillator is just another potential—solvable numerically or analytically—which is not always clear to students when first deriving the analytical solutions. That being said, it is possible to skip this chapter without detracting from the rest of the text.

Tunneling is only introduced as an end-of-chapter problem in McQuarrie Chapter 4, or in very qualitative terms in Engel's textbook. I find this topic interesting, and dedicate an entire class to tunneling, first working through the analytical setup of the square barrier problem (using the problems in McQuarrie Chapter 4), and then through the numerical approach presented here in Chapter 3. If tunneling is not part of your course, you may skip this chapter, as the subsequent text does not depend on it.

The next few weeks of the course are dedicated to the canonical single-particle model problems: harmonic oscillator (McQuarrie Chapter 5), rigid rotator (McQuarrie Chapter 6), and hydrogen atom (McQuarrie Chapter 7). These classes give the students another chance to utilize the finite difference method (Chapter 2) to find the eigenstates of these potentials. For example, McQuarrie discusses anharmonic potentials (e.g., the Morse potential) in Chapter 5, and in conjunction with this I have students solve the Morse potential numerically using finite differences (Problem 2-5). Moreover, during the discussion of selection rules (McQuarrie Section 5.12), I find it useful to have students calculate the dipole coupling matrix elements numerically using their finite difference solutions to the harmonic oscillator and Morse potentials—again, with the purpose of revealing the underlying calculation in a more transparent way, and showing how this can be done for more complicated wavefunctions than the ones derived in the textbook.

Following the hydrogen atom, it is typical for textbooks to treat perturbation theory and the variational method (McQuarrie Chapter 8), which is a natural place to use Chapter 4. I typically teach this as one course period for the variational method using McQuarrie Sections 8.1–8.3 and Chapter 4 in this book, and a second class period introducing perturbation theory (McQuarrie Sections 8.4–8.6) and continuing with further variational problems from Chapter 4 in this book.

Most textbooks—including McQuarrie—present Hartree-Fock theory in "atomic" and "molecular" forms. To this end, Chapter 5 is broken up into three sections: preliminary theoretical background (chapter introduction), helium atom (Section 5.1), and hydrogen molecule (Section 5.2). By the end of Chapter 5 students will write a molecular Hartree-Fock code from scratch, and this takes some time. I find it helpful to distribute this over multiple class periods. For example, after a class on many-electron atoms (McQuarrie Chapter 9), we spend a class working on the Hartree-Fock theory of the helium atom (Chapter 5 introduction and Section 5.1). Then, after finishing McQuarrie's discussion of the hydrogen molecule (McQuarrie Chapter 10), we spend two classes on the molecular Hartree-Fock calculations. The first class introduces Gaussian basis functions (McQuarrie 12.1–12.2 and Sections 5.2.1–5.2.3 here). In my experience, it is better not to rush this, so that students can become comfortable with the concept of fitting multiple sets of Gaussian functions to reproduce Slater orbitals, and the more general idea of having basis sets that are not the minimal atomic orbitals they learned in their earlier chemistry courses. The next class is spent building, running, and modifying the molecular Hartree-Fock calculation (Section 5.2.4 to the end of the chapter). I typically recommend that students reread McQuarrie 12.1 and review McQuarrie Chapter 10 for this second session.

After having built their own Hartree-Fock code, running some packaged calculations (e.g., using Gaussian/GAMESS/Q-Chem via WebMO) is rather straightforward. McQuarrie Chapter 12 provides sufficient theoretical grounds for this. In keeping with the very brief description of density functional theory in that chapter, the corresponding portion of this book (Chapter 8) is relatively brief and focuses only on atomic problems, so as to avoid having to introduce sophsiticated numerical quadrature methods.

Finally, we end the semester with a discussion of Hückel theory (Chapter 6). The computational approach simplifies the treatment of aromatic hydrocarbons (it's just matrix diagonalization!), which makes time available to study aspects of molecular properties and reactivity. For that reason, I divide Hückel theory into two class periods. The first class introduces the theory (McQuarrie 11.6–11.8) and the basic computational approach (Sections 6.1–6.3). The second class period is spent discussing the treatment of heteroaromatic molecules, molecular properties, and frontier molecular orbital theory of reactivity (Sections 6.4–6.7). The Hückel theory chapter only relies on having completed the variational methods chapter (Chapter 4), and only refers to the Hartree-Fock chapter in passing but not in a substantial way.

The chapters on quantum theory of solids (Chapter 7) and quantum Monte Carlo methods (Chapter 9) are outside this first undergraduate course in quantum chemistry, but I have used these as additional readings for (optional) end-of-the-semester independent projects or as a starting point for advanced courses in solid-state chemistry. My colleagues in physics have also used these as a starting point for independent projects in computational physics courses. Chapter 7 benefits from some exposure to Hückel theory (Sections 6.1–6.2 are sufficient). Chapter 9 can be used independently from the rest of the book by readers who have sufficient programming and quantum mechanics backgrounds.

USER'S GUIDE WHEN TEACHING THERMODYNAMICS

I use McQuarrie and Simon's *Molecular Thermodynamics* published by University Science Books, but the discussion below can be generalized to any other undergraduate-level (statistical-mechanics-oriented) thermodynamics course. The class meets twice a week for 90 minutes during a 14-week semester.

As mentioned earlier, most of my students take this as their first physical chemistry class. To that end, I like to spend more time explicitly introducing the concepts of programming and working with data in Mathematica and of modeling physical problems in general. To that end, I allocate a full class period to introducing Mathematica and its features, using Appendixes A and B.

I then spend the next week discussing quantum mechanics in a nutshell, using McQuarrie and Simon Chapter 1, supplemented by Appendix C of this book. The first day is spent on electronic, vibrational, and rotational energy levels (McQuarrie and Simon Sections 1.1–1.8); relevant problems on this material are a natural application of the computational methods that were previously introduced. (I try to let students discover that for themselves rather than explicitly forcing them to do so; but if you would prefer to force them, you can introduce Appendix C at the same time.) The second day is spent on the quantized energies of polyatomic molecules and computational approaches (McQuarrie and Simon 1.9–1.11 and Appendix C here). This continues the gentle introduction to programming by defining and using some simple functions for the energy levels. During this class, I also demonstrate how to use WebMO, and have the students do some basic calculations on diatomic and polyatomic spectroscopic constants, as described in Appendix D. Appendixes C and D are intended primarily for students who have not taken a previous (computational) physical chemistry course, so they go into some depth on function definitions and use. If your students have already completed Part I of this book, then neither appendix is necessary.

The next week is spent introducing the classical gas equations of state, using McQuarrie and Simon Chapter 2, and Chapter 10 in this book. This shows students the power of a computational approach to these types of problems, and gives them ample opportunity to practice various computational tasks (e.g., plot functions, solve cubic equations, perform symbolic integration and differentiation).

The introduction and initial studies of molecular partition functions (McQuarrie and Simon Chapters 3–4) doesn't relate to any explicit content in this book. However, my students generally gravitate toward using Mathematica to perform symbolic manipulations (like those described in Chapter 10) without much coaxing.

After completing McQuarrie and Simon Chapter 4, I introduce Metropolis Monte Carlo (Chapter 11) to demystify the meaning of the partition function and provide a tool for treating interacting systems. I feel this is important, because it enables students to go beyond the gas-phase-centric approach in McQuarrie and Simon. Condensed phases and interacting systems are more compelling for students (particularly those with biological interests). One class period is devoted to having students work through the problems at the end of Chapter 11.

At this junction there are several paths through the material. One can (i) combine the 2D Ising model (Chapter 12) with the introduction to Monte Carlo methods; (ii) dedicate a class to it; or (iii) wait until after the treatment of phase equilibria and solubility (McQuarrie and Simon Chapters 9 and 10) and combine it with a microscopic application. I've taught it all ways and don't have a clear preference. For the first path, have students read both Chapter 12 and Chapter 11, and practice the material in a single class. For the second path, dedicate the next class to Chapter 12. With either of these two paths, you have another option about when to cover applications of the Ising model to phase transitions and solubility. You can either (a) introduce the material in Sections 13.2 and 13.3 alongside McQuarrie and Simon Chapters 9 and 10 (respectively); or (b) dedicate an entire class to Sections 13.2–13.3 after finishing McQuarrie and Simon Chapter 10. Option (a) takes less time, but I find it to be a bit rushed. Option (b) takes more class time, but places a stronger emphasis on building and using microscopic models. Finally, path (iii) implies option (b); i.e., the third possibility is to wait until after finishing McQuarrie and Simon Chapter 10, and then to dedicate a single class to Chapter 12 and Sections 13.2–13.3. I have found this also to be a bit rushed, and it has the hazard of making students confuse the Ising *model* (of interacting systems) with its *applications* (to specific phenomena).

Molecular Thermodynamics does not include chapters on the kinetic theory of gases, chemical kinetics, or surface chemistry. For this purpose, I use chapters from McQuarrie and Simon's *Physical Chemistry: A Molecular Approach,* 1st ed. (University Science Books).

In principle Chapter 14 on molecular dynamics could be used when studying kinetic theory of gases (McQuarrie and Simon, *Physical Chemistry: A Molecular Approach,* Chapter 27), though I have not done this in my introductory physical chemistry course, simply for lack of time. However, I often use this chapter as a starting point for students doing molecular dynamics research projects.

The deterministic kinetics material in Section 15.1 can be presented alongside the more traditional treatment in McQuarrie and Simon's *Physical Chemistry: A Molecular Approach,* Chapters 28 and 29. Because students find kinetic Monte Carlo methods conceptually challenging, I allocate an entire class period on stochastic kinetics (Section 15.2); this also provides an opportunity to do several of the problems at the end of the chapter that review both deterministic and stochastic kinetics solutions. Admittedly, stochastic kinetics is not (yet) a core topic in undergraduate physical chemistry curricula, and this section could be neglected entirely.

The treatment of adsorption isotherms using the Ising model (Section 13.4) would be best paired with the discussion in McQuarrie and Simon, *Physical Chemistry: A Molecular Approach,* Sections 31.6–31.8). Alternatively, one could also introduce this material in relationship to the Gibbs energy discussion in *Molecular Thermodynamics* Chapter 12, although the connection is more abstract. In practice, I have tended to use Section 13.4 as a tutorial for students beginning research projects on gas-surface interactions, rather than incorporate it into my introductory physical chemistry class. It would also be appropriate for including in a first course in surface science.

TO THE STUDENT

If you want to learn this material, *reading* isn't going to be enough. You have to put what you read into action.

One way to do this is to type in all the examples, paying close attention to the underlying logic of what is going on. This should not be an exercise in typing skills, but instead an opportunity to slow down your reading and think about what is going on. Ideally, you should look at the example on the page, think about it for a few moments, and then type each line of code from your own memory (rather than looking directly at the page). This is a way to engage your working memory so as to improve long-term memory formation.

Second, you need to do the exercises—not all of them, but at least some of them—to gain experience in taking apart the code and making it do what you want.

Third, you need to "play" by applying the methods to real scientific research problems. If you are reading this book as part of an organized course or research project, your instructor can help you identify problems to tackle; if you are reading this on your own, you might try searching the literature. You don't have to win a Nobel Prize with this first research project; a small task or model (however well thought out or unrealistic) will suffice to give you practice.

SYNTAX COLORING CONVENTIONS

The Mathematica interpreter colors variables and functions to indicate the current state or the role of these symbols. This syntax coloring is added automatically as you type, and reveals Mathematica's underlying interpretation of your code. Paying attention to these variations can assist you in understanding and debugging your code. Right-clicking and selecting "Why the Coloring . . . " from the resulting menu opens a window describing the underlying variable or function types in the current evaluation cell.

The Mathematica interpreter displays these in color. To render the syntax coloring in print, we have chosen the following scheme:

	Mathematica	Rendered in
Type	interpreter default	this book as
Operators and defined global variables and functions	Black, bold, roman type	**+**, **Sin[]**, **x**
Undefined global variables and functions	Blue, bold, roman type	x
Loop iteration variables	Cyan, roman type	i
Local variables in modules	Green, roman type	x
Function argument variables	Green, italic type	*x*_
Strings	Gray, bold, roman type	**"string"**
Comments	Gray, bold, roman type	(* comment *)

INTRODUCTION TO COMPUTATIONAL PHYSICAL CHEMISTRY