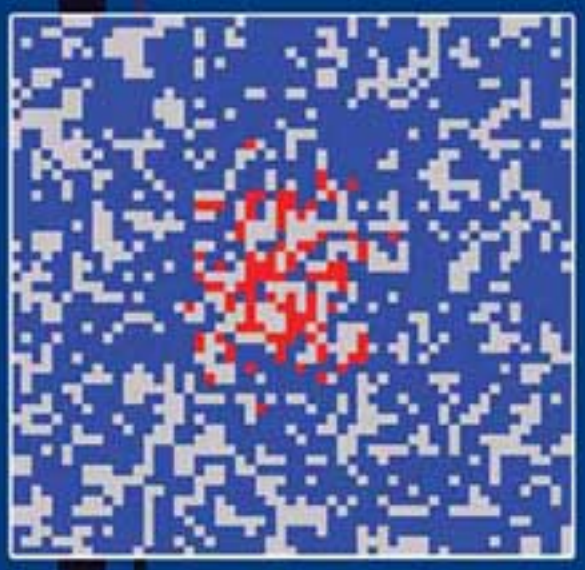
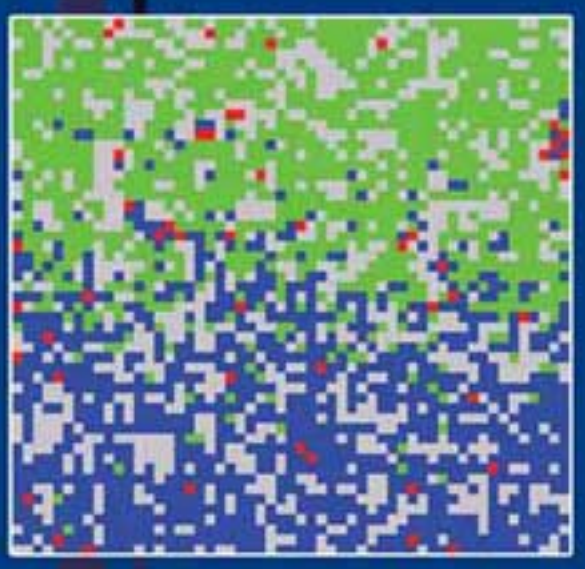
Modeling Chemical Systems using Cellular Automata





Lemont B. Kier
Paul G. Seybold
Chao-Kun Cheng

Springer

Cellular Automata Modeling of Chemical Systems

Cellular Automata Modeling of Chemical Systems

A textbook and laboratory manual

Lemont B. Kier, PhD

Professor of Medicinal Chemistry
Senior Fellow, CSBC
Virginia Commonwealth University
USA

Paul G. Seybold, PhD

Professor of Chemistry
Wright State University
External Fellow, CSBC
Virginia Commonwealth University
USA

Chao-Kun Cheng, PhD

Associate Professor of Computer Science
Fellow, CSBC
Virginia Commonwealth University

A publication of the Center for the Study of Biological Complexity
Virginia Commonwealth University
Richmond Virginia
USA

Springer

A C.I.P. Catalogue record for this book is available from the Library of Congress.

ISBN-10 1-4020-3657-4 (HB)
ISBN-13 978-1-4020-3657-6 (HB)
ISBN-10 1-4020-3690-6 (e-book)
ISBN-13 978-1-4020-3690-3 (e-book)

Published by Springer,
P.O. Box 17, 3300 AA Dordrecht, The Netherlands.

www.springeronline.com

Printed on acid-free paper

Printed in the Netherlands.

Table of Contents

Preface

Over the past two decades there has been a significant growth in the use of computer-generated models to study dynamic phenomena in the nature. These studies have ranged over many of the fields of human endeavor. For example, insect behavior is a target for dynamic models; automobile traffic is another. The sociologists have picked up on the possibilities afforded by computer models to study dynamic systems. In the physical and biological sciences, dynamic computer models have been used to study a variety of phenomena. Some studies in chemistry have appeared in the literature, but the field is so vast that only a small area has been considered for computer modeling. In our view chemistry is ripe for studies utilizing this paradigm. The study of chemistry is usually focused on changes; we establish a structure, a form, but it is of real interest when we consider how and to what it is transformed. Laboratory studies in schools introduce the student to simple processes that always work. More complex transformations are difficult to set up as experiments; they often do not "work" and so the didactic value of such experiences is marginal.

It is our purpose in this book to explore and reveal how some computer models might enrich the practical experiences, traditionally carried out in "wet" labs. We pursue this goal using one of the modeling schemes that was developed a half century ago: cellular automata. The record of cellular automata as a modeling paradigm is revealed in the literature. We have used cellular automata in our research for a decade, modeling solution and kinetic phenomena of chemical systems. We feel that this approach can bring new meaning to experimental chemistry in the form of *in silico* experiments. This book is dedicated to that objective.

The book is organized into three sections. In the first section we introduce the student to some of the concepts that are fundamental to an understanding

of chemical phenomena. These include a look at the subject of complexity. Imbedded in these concepts are general chemical phenomena such as self-organization, emergent properties, and local interactions. This section sets the stage for a look at some of the modeling techniques used to explore complex systems.

In the second section we present a brief overview of some currently used dynamic modeling methods before introducing cellular automata. After a brief history of this method we describe the ingredients that drive the dynamics exhibited by cellular automata. These include the platform on which cellular automata plays out its modeling, the state variables that define the ingredients, and the rules of movement that develop the dynamics. Each step in this section is accompanied by computer simulation programs carried on the CD in the back of the book.

With this background the student is then equipped to witness what has been done in chemistry using cellular automata models. These studies are accompanied by unfinished studies and challenges, "what if" ideas for the student. The laboratory in a general chemistry course is an ideal place to use this approach since it brings to the student views of many phenomena, previously difficult to visualize. As an adjunct to experimental work in the lab, it opens up a new level of understanding. It may even pique interest in pursuing new theoretical investigations in chemistry.

At a near final stage of writing this book, we had a golden opportunity to test the modeling exercises. Seven students in the Integrated Life Sciences graduate program at the Virginia Commonwealth University were asked to read the text and to perform many of the examples and studies. Their experiences were of immense value to us in finalizing the manuscript. We want to acknowledge them and thank them for their efforts. They are Xiangrong Kong, Julie Naumann, Jean Nelson, Antoine Nicolas, Elizabeth Prom, Alexander Tulchinsky, and Carl Zimmerman. We also want to thank Yingjin Cui for her help in creating some of the figures. The authors thank Marco Tomassini for early, helpful reviews of the manuscript. We thank Enguang Zhao for his help in preparing the Java version of the CA program. Finally we acknowledge the scholarly climate and encouragement given to us at the Center for the Study of Biological Complexity at the Virginia Commonwealth University.

Lemont B. Kier
Paul G. Seybold
Chao-Kun Cheng

Chapter 1

MODELING NATURE

The chess-board is the world; the pieces are the phenomena of the universe; the rules of the game are what we call the laws of Nature. The player on the other side is hidden from us. But we know that his play is always fair, just, and patient. But also we know to our cost, that he never overlooks a mistake, or makes the smallest allowance for ignorance.

—Thomas Huxley

It is the role, and the privilege, of a scientist to study Nature and to seek to unlock her secrets. To unlock these secrets, a certain process is customarily taken. Normally, the scientific process starts with observations; the scientist observes some part of the natural world and attempts to find patterns in the behaviors observed. These patterns, when they are uncovered out of what may otherwise be a quite complicated set of events, are then called the "laws" of behavior for the particular part of nature that has been scrutinized. But the process does not stop there. Scientists are not content merely to observe nature and catalog her patterns—they seek explanations for the patterns. The possible explanations that scientists propose take the form of hypotheses and theories—"models"—about how things work behind the scenes of outside appearance. This book is about one such type of model and how it can be used to understand the patterns of chemistry.

But what do we mean by a "model"? A model is a substitute, usually greatly simplified, for what the early quantum physicists called the *Ding an sich*—the "thing itself," the real thing. The mathematician Jacob Bronowski spoke of models as metaphors, likenesses that we snatch from the larger world of eye and ear and touch. [1] A model should simulate or imitate the real system and display in some revealing way its most important or interesting features; where possible it should capture the essence of the system without being overly cumbersome or complicated. Science writer George Johnson [2] has described the nature of a successful simulation:

The mark of a good simulation is that it separates the essential from the incidental, cutting through what is deemed irrelevant detail to get at the heart of the problem.

Many models in the physical sciences take the form of mathematical relationships, equations connecting some property with other parameters of the system. Some of these relationships are quite simple, e.g., Newton's second law of motion, which says that force = mass × acceleration: $F = ma$. Newton's gravitational law for the attractive force F between two masses m_1 and m_2 also takes a rather simple form

$$F = Gm_1m_2/r^2$$

where r^2 is the square of the distance separating the masses and G is a constant that rationalizes the units. But many mathematical relationships are much more complicated and rely on the techniques of calculus to describe the rates of change of the quantities involved. An example is the basic equation of quantum theory, the Schrödinger equation, which takes the more formidable form (it is not necessary here to dwell on the meanings of the symbols):

$$\frac{-\hbar^2}{2m}\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}+\frac{\partial^2}{\partial z^2}\right)\psi + V\psi = E\psi$$

In chemical kinetics, one finds linked sets of differential equations expressing the rates of change of the interacting species. Overall, mathematical models have been exceedingly successful in depicting the broad outlines of an enormously diverse variety of phenomena in nature. Some scientists have even commented in surprise at how well mathematics works in describing nature. So successful have these mathematical models been that their use has spread from the hard sciences to areas as diverse as economics and the analysis of athletic performance [3].

In other cases, models take a more pictorial form. In the early atomic models, an atom was first pictured by J. J. Thomson as a "plum pudding," with negative electrons (the "plums") embedded in a spread-out positive charge (the pudding), and then later by Ernest Rutherford and Niels Bohr as a planetary system with a tiny positive core surrounded by circling electrons, a model called the "nuclear atom." Today, within quantum theory, the nuclear atom picture has been further transformed into one with a positive nucleus surrounded by a cloud of electron probabilities. In biology, the double helix model of the structure of the genetic material DNA proposed by James Watson and Francis Crick in 1953 led to an explosion of studies in the field of molecular genetics. Charles Darwin's model of evolution by means of natural selection pictures species composed of a collection of individuals with a variety of different traits interacting with their environments. Individuals with some traits are better suited to survive and

reproduce, thereby passing on these traits to their offspring. Over time new traits are introduced through mutations, environments gradually (or sometimes rapidly) change, and new forms develop from the old ones. The modern model of the human brain envisions regions devoted to different functions such as sight, motor movements, and higher thought processes. In geology, the tectonic plate model of the Earth pictures expansive continental plates moving gradually over the planet's surface generating earthquakes as they meet and slide over one another. And in psychology, the Freudian approach pictures human behavior as resulting from the actions of invisible components of the mind termed the id, the ego, and the superego.

The key feature of successful models is that they produce results consistent with the experimental observations. Successful models capture the essential features of the systems of interest, and they customarily go beyond this simple reproduction to predict new features of the systems that may have previously escaped notice. In this latter case, the predictions provide an important means for testing the validity of the models.

At this point, it is helpful to dissect models into their most significant parts so that we can start from a common basis.

1.1. The system

Studies in chemistry or any realm of science commonly consist of a series of directed examinations of parts of nature's realm called systems. A system is an identifiable fragment of the world that is recognizable and that has attributes that one can identify in terms of form and/or function. We can give examples at any level of size and complexity and in essentially any context. Indeed, a dog is a system at a pet show; whereas the human heart is a system to the cardiologist; a tumor cell is a system to the cancer specialist; a star or planet or galaxy is a system to an astronomer; a molecule or a collection of molecules is a system to a chemist; and an atom or group of atoms is a system to a physicist. A system is, then, whatever we focus our attention upon for study and examination.

1.2. States of the system

A system is composed of parts that can be recognized and identified. As time goes by, a system under study may acquire different attributes as a result of changes among its parts, and over time its appearance or function may change. Each of the different stages through which the system passes in its evolution is called a state of the system. A dog grows old over time, passing through stages recognized in general terms as puppy, dog, old dog, and, finally, dead dog. A heart may change its pattern of contractions, going from normal to tachycardia

to ventricular fibrillation, each of which we categorize as a different state of functioning. A solution of ethyl acetate in water may slowly decompose to mixtures of ethyl acetate, acetic acid, and ethanol, through a sequence of states characterized by their different compositions. Water may start as a solid (ice), become a cool liquid, then a warmer liquid, and finally appear as a vapor at higher temperature, passing through these different stages as it is melted, heated, and vaporized. We refer to each set of conditions for the present purposes as a "state." It is the various states of the system that we focus attention upon when we study any system.

1.3. Observables

Our studies require us to analyze and describe the changes that occur in the systems we are interested in that evolve with time. To accomplish this analysis properly, we need to record specific features that characterize what is occurring. The features assigned for this purpose are termed *observables*. For example, we distinguish the puppy from the old dog by the changes in its physical appearance and its behavior. The changes in a heart's rhythm are recorded on special charts monitoring electrical signals. The changes occurring in a solution of ethyl acetate in water can be characterized by changes in the solution's acidity, by spectroscopic readings, or by detection of the odor of acetic acid. To be as precise as possible in a scientific investigation, it is necessary to assign numerical values to the characteristics that distinguish one state of a system from another. The state of a system is studied through detection and recording of its observables.

1.4. Interactions

The parts of a system naturally interact with one another, and the fascinating and often complex evolutions of natural systems depend crucially on the nature of these interactions. The interactions supply the driving forces for the changes that we observe in the systems. In addition, we can change the behavior of a system by introducing new elements or ingredients. Intrusions of this kind produce new interactions, which in turn alter the system. By carefully choosing the added factors and interactions, we can develop new patterns of observables that may be revealing. Interaction with your dog might include exercising to increase his running stamina, which in turn will lead to a new, improved set of health-related (state) indicators. Electrical stimulation of a fibrillating heart can introduce interactions that lead to the conversion of the heart from the fibrillating state to a normal, healthy state of performance. Heating the ethyl acetate solution will eventually complete the hydrolysis reaction and distill away the resulting

ethanol leaving a solution of acetic acid. The interactions introduced and the accompanying changes in the systems' observables produce information about the nature of the systems and their behaviors under different conditions. With enough observables, we may be able to piece together a reasonable description, a model, for how the system operates.

1.5. Back to models

From a carefully selected list of experiments with a system, we can evoke certain conclusions. The mosaic of information leads us to piece together a description of the system, what is going on inside it, the relationships among its states, and how these states change under different circumstances. In the case of our dog, the exercise tests may lead us to theorize that the dog is in good or poor health. With the heart, the electrical impulses that we record can reveal a pattern of changes (observables) that we theorize belong to a healthy (or diseased) heart. By subjecting the solution of hydrolyzing chemicals to fractional distillation and chemical analysis, we may theorize that we originally had a system of water and ethyl acetate.

We can arrive at our theories in two main ways. In the first, as illustrated earlier, we subject a system to experimental perturbations, tests, and intrusions, thereby leading to patterns of observables from which we may concoct a theory of the system's structure and function. An alternative approach, made possible by the dramatic advances that have occurred in the area of computer hardware in recent times, is to construct a computer model of the system and then to carry out simulations of its behavior under different conditions. The computer "experiments" can lead to observables that may be interpreted as though they were derived from interactions.

1.6. Simulations

It is important to recognize the different concepts conveyed by the terms "model" and "simulation," even though these terms are sometimes used interchangeably. As noted above, a model is a general construct in which the parts of a system and the interactions between these parts are identified. The model is necessarily simpler than the original system, although it may itself take on a rather complicated form. It consists of ingredients and proposals for their interactions.

Simulations are active imitations of real things, and there are generally two different types of simulations with different aims. In one approach, a simulation is merely designed to match a certain behavior, often in a very limited context. Thus, a mechanical noisemaker may simulate a desired sound and does so

through a very different mechanism than the real sound maker. Such a simulation reveals little or nothing about the features of the original system, and is not intended to do so. Only the outcome, to some extent, matches reality. A hologram may look like a real object, but it is constructed from interfering light waves.

A second type of simulation is more ambitious. It attempts to mimic at least some of the key features of the system under study, with the intent of gaining insight into how the system operates. In the context of our modeling exercise, a simulation of this sort means letting our model "run." It refers to the act of letting the parts of our model interact and seeing what happens. The results are sometimes very surprising and informative.

1.7. Models in chemistry

Chemistry, like other sciences, progresses through the use of models. Models are the means by which we attempt to understand nature. In this book, we are primarily concerned with models of complex systems, those systems whose behaviors result from the many interactions of a large number of ingredients. In this context, two powerful approaches have been developed in recent years for chemical investigations: molecular dynamics and Monte Carlo calculations [4–7]. Both techniques have been made possible by the development of extremely powerful, modern, high-speed computers.

Both of the above approaches rely in most cases on classical ideas that picture the atoms and molecules in the system interacting via ordinary electrical and steric forces. These interactions between the species are expressed in terms of "force fields," i.e., sets of mathematical equations that describe the attractions and repulsions between the atomic charges, the forces needed to stretch or compress the chemical bonds, repulsions between the atoms due to their excluded volumes, etc. A variety of different force fields have been developed by different workers to represent the forces present in chemical systems, and although these differ in their details, they generally tend to include the same aspects of the molecular interactions. Some are directed more specifically at the forces important for, say, protein structure, while others focus more on features important in liquids. With time more and more sophisticated force fields are continually being introduced to include additional aspects of the interatomic interactions, e.g., polarizations of the atomic charge clouds and more subtle effects associated with quantum chemical effects. Naturally, inclusion of these additional features requires greater computational effort, so that a compromise between sophistication and practicality is required.

The molecular dynamics approach has been called a "brute-force solution of Newton's equations of motion." [8] One normally starts a simulation using some assumed configuration of the system components, for example, an X-ray

diffraction structure obtained for a protein in crystalline form or some arrangement of liquid molecules enclosed in a box. In the protein case, one might next introduce solvent molecules to surround the protein. One then allows the system, protein-in-solvent or liquid sample, to evolve in time as governed by the interactions of the force field. As this happens, one observes different configurations of the species that appear and disappear. "Periodic" boundary conditions are usually applied such that molecules leaving the box on the right side appear on the left; those leaving at the top appear at the bottom, and so forth. The system's evolution occurs via time steps (iterations) that are normally taken to be very short, e.g., 0.5–2.0 fs (10^{-15} s), so that Newton's second law of motion

$$F = ma = m(dv/dt)$$

can be assumed to hold a nearly linear form. The evolution is followed over a very great number of time steps, often more than a million, and averages for the features of interest of the system are determined over this time frame. Because the calculation of the large number of interactions present in such a system is very computationally demanding, the simulations take far longer than the actual time scale of the molecular events. Indeed, at present most research level simulations of this type cover at best only a few tens of nanoseconds of "real time." (Note that 10^6 steps of 1 fs duration equal one nanosecond, 10^{-9} s.) Whereas such a timeframe is sufficient to examine many phenomena of chemical and biochemical interest; other phenomena, which occur over longer time scales, are not as conveniently studied using this approach.

The Monte Carlo method for molecular simulations takes a rather different approach from that of the molecular dynamics method [9]. Rather than watching the system evolve under the influence of the force field, as done in molecular dynamics, a very large number of possible configurations of the system are sampled by moving the ingredients by random amounts in each step. New configurations are evaluated according to their energies, so that those lowering the system's energy are accepted, whereas those raising the energy are conditionally "weighted" or proportionately accepted, according to their potential energies. The weighting is normally taken to have the form of the Boltzmann distribution, i.e., to be proportional to $e^{-\Delta V/kT}$, where ΔV is the potential energy change, k is Boltzmann's constant, and T is the absolute temperature. From statistical analysis of a large, weighted sample ("ensemble") of such configurations, one can ascertain many of the important thermodynamic and structural features of the system. A typical sample size employed for this purpose might encompass between one and ten million configurations.

Both the molecular dynamics and the Monte Carlo approaches have great strengths and often lead to quite similar results for the properties of the systems investigated [10]. However, these methods depend on rather elaborate models for the molecular interactions. As a result, as noted above, both methods are

very computationally demanding and research level calculations are normally run on supercomputers, clusters, or other large systems. In Chapter 2, we shall introduce an alternative approach that greatly simplifies the view of the molecular system, and that, in turn, significantly reduces the computational demand, so that common personal computers suffice for calculations and elongated time frames can be investigated. The elaborate force fields are replaced by simple, heuristic rules. This simplified approach employs cellular automata.

I have avoided words such as "true and false", "correct and incorrect", and "valid and invalid". Such descriptives have no place in a discussion of chemical models which are, above all, fictitious. Models—one must never forget—are to be used, not believed.

—Fredric Menger [11]

References

1. J. Bronowski, *The Ascent of Man*, Little, Brown & Co., Boston, 1973.
2. G. Johnson, *Fire in the Mind: Science, Faith, and the Search for Order*, Knopf, New York, 1995.
3. M. Lewis, *Moneyball: The Art of Winning an Unfair Game*, W. W. Norton, New York, 2003, p. 77.
4. J. M. Haile, *Molecular Dynamics Simulation: Elementary Methods*, Wiley, New York, 1992.
5. A. R. Leach, *Molecular Modelling: Principles and Applications*, Longman, Harlow, England, 1996.
6. M. P. Allen and D. J. Tildesley, *Computer Simulation in Chemical Physics*, Kluwer Academic Publishers, Boston, 1998.
7. D. J. Tildesley, *The Molecular Dynamics Method*, Kluwer Academic Publishers, Boston, 1998, 23–47.
8. M. P. Allen, Introduction to Monte Carlo simulations. In *Observation, Prediction and Simulation of Phase Transitions in Complex Fluids*, M. Baus, L. F. Rull, and J.- P. Ryckaert, Eds., Kluwer Academic Publishers, Boston, 1995, 339–356.
9. W. L. Jorgensen, Monte Carlo simulations for liquids. In *Encyclopedia of Computational Chemistry*, P. V. Ragué Schleyer, Ed., Wiley, New York, 1998, 1754–1763.
10. W. L. Jorgensen and J. Tirado-Rives, Monte Carlo vs molecular dynamics for conformational sampling. J. Phys. Chem., 100, 1996, 14508–14513.
11. F. M. Menger, In *Advances in Molecular Modeling*, D. Liotta, Ed., JAI Press, Greenwich, CN, 1988, 189–213.

Chapter 2

CELLULAR AUTOMATA

To discover and analyze the mathematical basis for the generation of complexity, one must identify simple mathematical systems that capture the essence of the process. Cellular automata are a candidate class of such systems. . . . Cellular automata promise to provide mathematical models for a wide variety of complex phenomena, from turbulence in fluids to patterns in biological growth.

–Stephen Wolfram [1]

In the first chapter several traditional types of physical models were discussed. These models rely on the physical concepts of energies and forces to guide the actions of molecules or other species, and are customarily expressed mathematically in terms of coupled sets of ordinary or partial differential equations. Most traditional models are deterministic in nature—that is, the results of simulations based on these models are completely determined by the force fields employed and the initial conditions of the simulations. In this chapter a very different approach is introduced, one in which the behaviors of the species under investigation are governed not by forces and energies, but by rules. The rules, as we shall see, can be either deterministic or probabilistic, the latter leading to important new insights and possibilities. This new approach relies on the use of cellular automata.

Cellular automata (CA) were first proposed by the mathematical physicist John von Neumann and the mathematician Stanislaw Ulam more than a half century ago [2–4] and similar ideas were suggested at about the same time, in the 1940s, by the German engineer Konrad Zuse [5–7]. Von Neumann's interest was in the construction of "self-reproducing automata." [8] His original idea was to construct a series of mechanical devices or "automata" that would gather and assemble the parts necessary to reproduce themselves. A suggestion by Ulam led him to consider more abstract systems consisting of grids with moving ingredients, operating under sets of rules. The first such system proposed by von

Neumann consisted of a two-dimensional grid of square cells, each having a set of possible states, along with a set of rules. The system he developed eventually employed as many as 29 different possible states for the cells, and was, at the least, clumsy to work with. With the development of modern digital computers, however, it became increasingly clear to a small number of scientists that these very abstract ideas could in fact be usefully applied to the examination of real physical and biological systems, with interesting and informative results [9,10].

A number of research groups have subsequently developed different realizations of the CA paradigm for the study and simulation of a broad range of physical, biological, chemical, and even sociological, phenomena. These models have contributed important new insights regarding the deeper, often hidden, factors underlying a host of complex phenomena. These diverse CA studies have been especially important in treating the often-surprising behaviors of systems where large numbers of complicated interactions between the system ingredients serve to hide the general patterns involved and, in addition, render the conventional, differential-equation-based methods difficult to implement or ineffective, i.e., complex systems. In this book we shall describe the details of particular CA models developed by our own research groups for the study and simulation of complex physical, chemical, and biochemical systems.

The present chapter will focus on the practical, "nuts and bolts" aspects of this particular CA approach to modeling. In later chapters we will describe a variety of applications of these CA models to chemical systems, emphasizing applications involving solution phenomena, phase transitions, and chemical kinetics. In order to prepare readers for the use of CA models in teaching and research, we have attempted to present a user-friendly description. This description is accompanied by examples and "hands-on" calculations, available on the compact disk that comes with this book. The reader is encouraged to use this means to assimilate the basic aspects of the CA approach described in this chapter. More details on the operation of the CA programs, when needed, can be found in Chapter 10 of this book.

2.1. What are cellular automata?

But just what are cellular automata? Mathematician Stephen Wolfram has defined cellular automata as follows [1]:

Cellular automata are simple mathematical idealizations of natural systems. They consist of a lattice of discrete identical sites, each site taking on a finite set of, say, integer values. The values of the sites evolve in discrete time steps according to deterministic rules that specify the value of each site in terms of the values of neighboring sites. Cellular automata may thus be considered as

discrete idealizations of the partial differential equations often used to describe natural systems.

Wolfram has elaborated on this description elsewhere [11, 12]. As we shall see, the restriction to deterministic rules is unnecessary, and we shall in fact make extensive use of probabilistic rules in our studies of real physical and chemical systems.

Cellular automata, then, are models, in the same sense that the Monte Carlo and molecular dynamics approaches are models, which can be employed for the purpose of simulating real systems. We shall use the term cellular automaton (singular) to refer to a model consisting of the following components:*

- A grid composed of cells.
- A set of ingredients.
- A set of local rules governing the behaviors of the ingredients.
- Specified initial conditions.

Once the above components of the model are defined, a simulation can be carried out. In the simulation the system evolves *via* a series of discrete time-steps, or iterations, in which the model rules are applied to all the ingredients of the system, and the configuration of the system is accordingly updated.

A striking feature of the cellular automata (CA) models is that they treat not only the ingredients, or agents, of the model as discrete entities, as do the traditional models of physics and chemistry, but now time (iterations) and space (the cells) are also regarded as discrete, in stark contrast to the continuous forms for these parameters assumed in the traditional, equation-based models. In practice, as we shall see, this distinction makes little or no difference, for the traditional continuous results appear, quite naturally, as limiting cases of the discrete CA analyses. Nonetheless, this quantization of time and space does raise some interesting theoretical and philosophical questions, which we shall, however, ignore at this time.†

A new, important feature is sometimes observed in studies of the evolutions of these computational systems: the development of unanticipated patterns of

* Historically, there has been some looseness of terminology in this field. A few authors have used the term "cellular automaton" to refer to a cell. We shall not use the term in this sense.

† Some recent theories of modern physics, such as string theory and quantum loop gravity theory, raise the interesting possibility that at some ultimate level—perhaps at very short times approaching the so-called *Planck time*, about 10^{-43} s, and at distances approaching the *Planck distance*, about 10^{-35} m—the discrete natures of time and space might reveal themselves. Time, for example, might proceed in jumps at very short times, in the same way that quantum theory shows that energy comes in jumps called quanta, when events at the submicroscopic level are examined. Physicists refer to such hypothetical time units as "chronons" in analogy to the "photons" of light energy. At the present time, of course, such extremely short times and distances lie well beyond experimental detection.

ordered dynamical behavior. These patterns have come to be called emergent properties. As Stuart Kauffman has expressed it [13]

Studies of large, randomly assembled cellular automata . . . have now demonstrated that such systems can spontaneously crystallize enormously ordered dynamical behavior. This crystallization hints that hitherto unexpected principles of order may be found [and] that the order may have significant explanatory import in [biology] and . . . physics.

Experience has shown that this enticing assessment is, in fact, too cautious: cellular automata carry great potential for revealing "hitherto unexpected principles" not only in biology and physics but also in chemistry and a host of other fields as well.

As noted earlier, a variety of different models can be developed within the general CA framework pictured above. In this book we describe a particular realization of this concept that the present authors have found especially well suited for the examination of physicochemical and biochemical systems. We shall now examine the components of this CA model in more detail.

2.2. The grid and the cells

2.2.1. The grid

The grid in a CA model may contain a single cell, or more commonly, a larger collection, with possibly as many as 100,000 or more cells. In principle, the grid itself might be one-, two-, or three-dimensional in form, although most studies have used two-dimensional grids. These two-dimensional grids will be the principal focus in this book. A moving ingredient may encounter an edge or boundary during its movements. Three general types of two-dimensional grids will be considered relating to the boundaries: (1) a box, (2) a cylinder, and (3) a torus. In the box grid, moving ingredients encounter boundaries on all four sides; in the cylinder they encounter only top and bottom boundaries; and in the torus no boundaries are present to restrict the ingredient movements. An illustration of a $7 \times 7 = 49$ cell grid of square cells occupied by two types of ingredients, A and B, is shown in Figure 2.1.

The nature of the grid type employed will normally depend on the boundary characteristics of the system of interest. For some systems, e.g., when the ingredients themselves are either stationary or confined, a box grid is perfectly suitable. In other cases, one may need only a constraining top and bottom (or right and left sides), and a cylindrical grid will be most appropriate. An example here would be the condensation of a gas to a liquid under the influence of gravity (see Chapter 9). For this, a bottom is required to restrain the liquid below and a top holds in the gas, but the ingredients remain free to move unobstructed to the

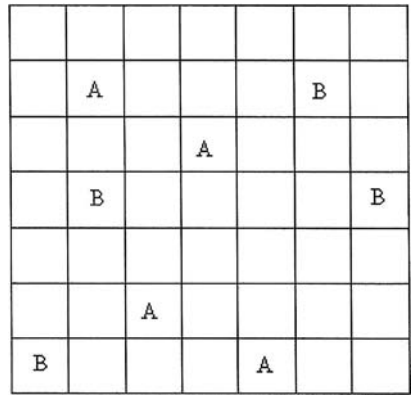


Figure 2.1. A two-dimensional cellular automata grid. Shown are two sets of occupied cells of different states, A and B. The unoccupied cells are blank

right or left, such that those moving off the edge to the right will appear at the left, and those moving off the grid to the left will appear on the right. In all these cases the ingredients can only move to unoccupied cells. The torus effectively simulates a small segment of a larger, unrestricted system by allowing cells also to move off the top edge and appear at the bottom, or move off the bottom edge and appear at the top. These features are shown in Figures 2.2 and 2.3.

In most cases, and in all cases found in this book, the cells of the boundaries are themselves inert, and have no interactions with the grid ingredients other than to constrain their movements in certain directions. However, more generally the boundary cells can be constructed to have active properties just like any other ingredient, following rules (see below) that permit joining, breaking,

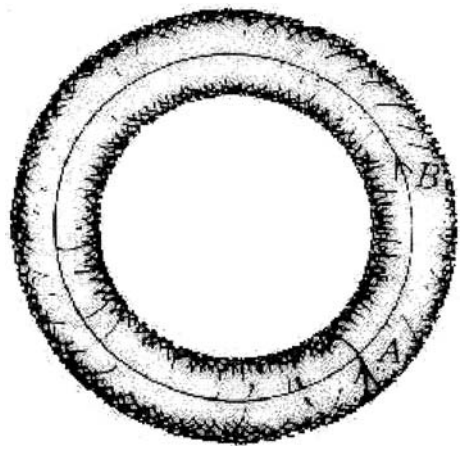


Figure 2.2. A torus grid eliminates boundaries

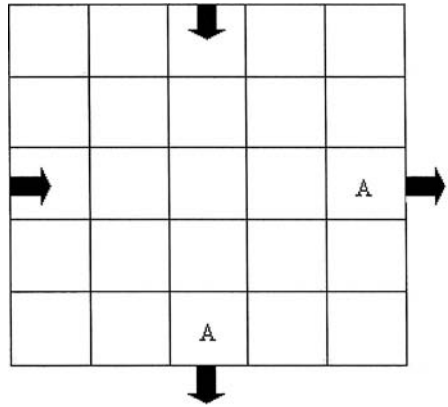


Figure 2.3. Movement of cell ingredients at the boundaries of a grid on the surface of a torus. One ingredient A might move off the grid to the right and reappear at the left edge of the grid. Another ingredient A might move off the bottom of the grid and reappear at the top of the grid

and reacting with the grid ingredients. Thus in principle, the boundaries can be either inert or active in some way.

2.2.2. The cells

The cells themselves can take a variety of shapes; they can be triangles, squares, hexagons, or other shapes on the two-dimensional grid, with square cells being most common. Each cell in the grid can normally exist in a number of distinct "states" which define the occupancy of the cell. The cell can be empty or contain a specific ingredient, where the ingredient, if present, might represent a particle, a type of molecule or isomer, a particular molecular electronic state, an organism, an automobile, or some other entity pertinent to the study in question.

The choice of the cell shape is based on the objective of the study. In the case of studies of water-related phenomena, for example, square cells are especially advantageous, since water molecules, H_2Os, are quadravalent with respect to their participation in intermolecular hydrogen bonding. An individual water molecule can employ two hydrogen atoms and two lone pairs of electrons to form hydrogen bonds with its neighbors. This leads to the tetrahedral configuration found in ice, a structure that is retained to some extent in the liquid state. The four faces of a square cell thus correspond nicely to the four hydrogen-bonding opportunities of a water molecule.

The interactions of an ingredient with other ingredients take place at the cell edges. Originally, cellular automata models routinely assumed that all of the edges of a given ingredient should obey the same rules. More recently, the idea of a variegated cell, in which each edge can have its own independent rules

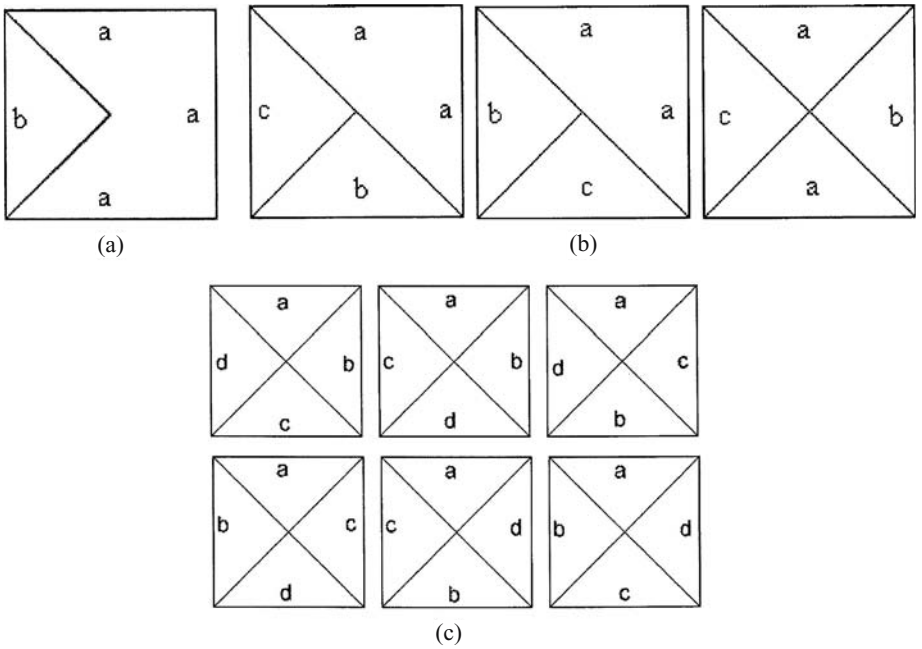


Figure 2.4. Examples of variegated cells: (a) Two different types of edges with different rules, (b) Three different types of edges with different rules, and (c) Four different types of edges with different rules

for interaction with other ingredients, has been introduced and shown to have considerable value in modeling [14]. Examples of some types of variegated cells are shown in Figure 2.4.

2.2.3. Cell neighborhoods

As we shall soon see, the movements and other actions of an ingredient on the grid are governed by rules, and these rules depend only on the nature of the cells in close proximity to the ingredient. This proximate environment of a cell is called its neighborhood. The most common neighborhood used in two-dimensional cellular automata studies is called the von Neumann neighborhood, after the original pioneer of the CA method. This neighborhood for a cell, A, refers to the four, B, cells adjoining its four faces (Figure 2.5a). Another common neighborhood is the Moore neighborhood, pictured in Figure 2.5b, referring to the eight B cells completely surrounding cell A, including those cells on the diagonals. Another useful neighborhood is the extended von Neumann neighborhood, shown in Figure 2.5c, where the four C cells lying just beyond the four B cells of the von Neumann neighborhood are included.

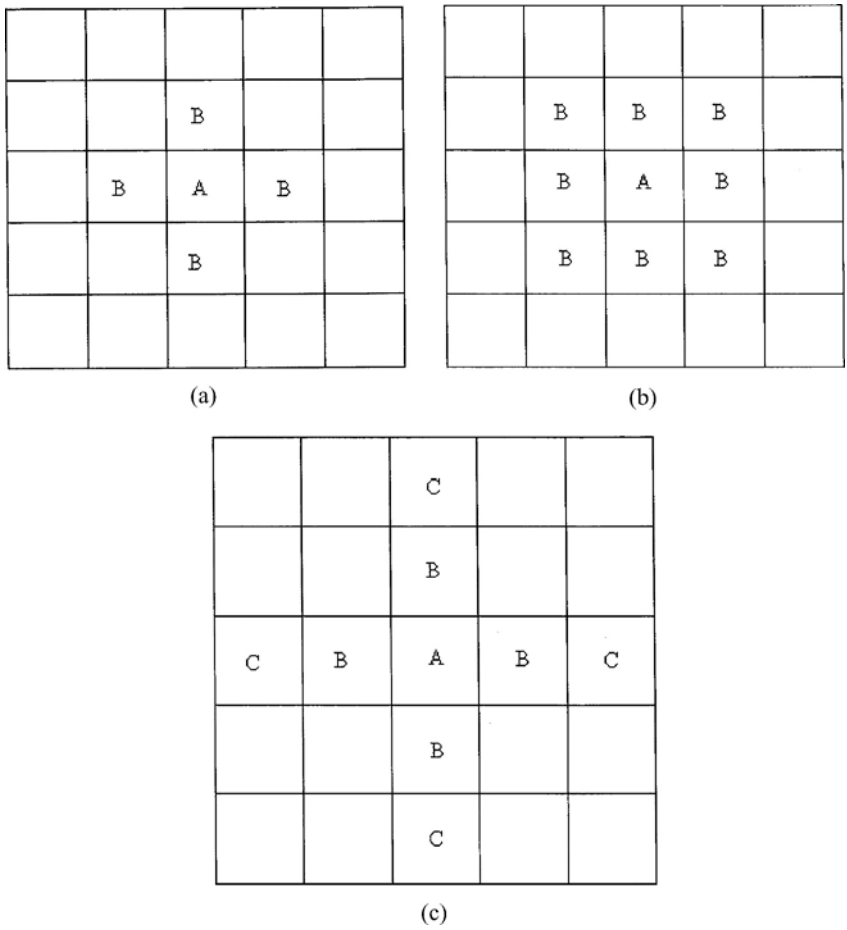


Figure 2.5. Cell neighborhoods: (a) the von Neumann neighborhood, (b) the Moore neighborhood, and (c) the extended von Neumann neighborhood of cell A

2.3. The rules

2.3.1. Types of rules

Several different types of rules govern the behaviors of the ingredients on the grid, and thereby the subsequent evolutions of the CA systems. Movement rules govern the movements of ingredients about the grid. These rules take several forms. The breaking probability, P_B(AB), determines the degree to which two adjacent ingredients A and B tend to stay bonded, or "stick," to each other. The joining parameter, J(AB), establishes the propensity of an A ingredient to move toward or away from a B ingredient, when these two ingredients are separated by an empty cell. The free-moving probability P_m(A)

of an ingredient A defines the ingredient's tendency to move more rapidly or more slowly on the grid. A gravitational parameter G, if present, denotes a greater-than-random tendency for an ingredient to move downward on the grid, thereby distinguishing motion in this direction from motion in the other directions on the grid. Transition rules govern the likelihood that during an iteration of the system, an ingredient will transform to some other species. The simple first-order transition probability P_T (AB) defines the probability that an ingredient of species A will change to species B during an iteration. The reaction probability P_R(AB) defines the probability that ingredients A and B will transform to species C and D, respectively, when they "encounter" each other (come into contact) during their movements about the grid. Occasionally other types of rules may be added. The key features of all these rules are that they are local, involving only an ingredient itself and possibly those ingredients in its immediate neighborhood, and that they are uniformly applied throughout the CA simulation.

2.3.2. Transition rules

Transitions occur constantly in nature; molecules change from one tautomeric form to another, radioactive nuclei decay to form other nuclei, acids dissociate, proteins alter their shapes, molecules undergo transitions between electronic states, chemicals react to form new species, and so forth. Transition rules allow the simulation of these changes.

As indicated, transition rules govern the probability that during each iteration of the simulation, an ingredient will transform to a different type of ingredient. If $P_T(AB) = 1.0$ the transition A $\rightarrow$ B is certain to occur; if $P_T(AB) = 0.0$, it will never occur. But if, for example, $P_T(AB) = 0.5$, then during each iteration, there will be a 50% chance that the transition A $\rightarrow$ B will occur. The first two cases can be considered deterministic, since they do not allow for different outcomes. The third case is stochastic, however, since it allows different outcomes, the ingredient might remain unchanged or it might transform to a different state.[‡] The transition probabilities may, in some cases, depend on the conditions prevailing in neighboring cells. For example, the transformation probability $P_T(AB)$ might depend on the occupancies of neighboring cells. In reaction simulations two ingredients A and B that come in contact on the grid will have a probability $P_R(AB)$ of "reacting," or transforming, to

[‡] The probabilities are enforced using a random-number generator in the CA program. Suppose that the random-number generator generates numbers between 1 and 1000. For a 20% probability of movement, one might assign a choice of numbers between 1 and 200 inclusively as a "move" decision, while a choice in the remaining number set, 201–1000, is a "no-move" decision. Each ingredient is then assigned a random number, and correspondingly behaves according to that value.

other species, say, C and D, during such an "encounter." In this case the reaction probability $P_R(AB)$ defines the probability that the reaction $A + B \rightarrow C + D$ will occur when A and B encounter one another in the course of their motions. If $P_R(AB) = 1$, the reaction will take place on every encounter, but if $P_R(AB) = 0.1$, for example, only 10% of such encounters will lead to reaction.

2.3.3. Movement rules

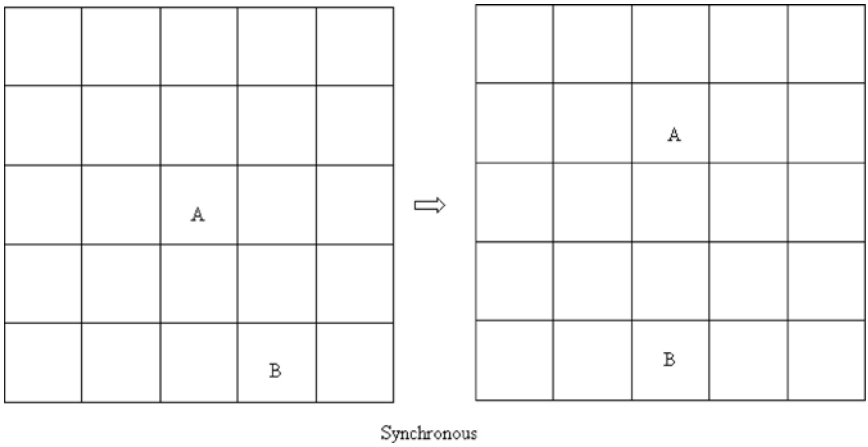
Much of the dynamic character of cellular automata models is developed through the movements of the ingredients about the grid. During each time-step interval, or iteration, in the CA simulation, an ingredient on the grid has the possibility of moving vertically or horizontally to an adjacent, unoccupied cell. In the absence of further restrictions, a free ingredient would therefore, over time, perform a random walk about the grid. Normally, however, there are other ingredients on the grid, and the presence of these ingredients will influence the motion of the first ingredient. During each iteration the movement of every ingredient on the grid is computed based on rules (described below) that involve the status of its neighboring cells, i.e., whether these cells are empty or occupied, and, if occupied, by what types of ingredients. Deterministic cellular automata use a fixed set of rules, the values of which are immutable and uniformly applied to the ingredients. In probabilistic, or stochastic, cellular automata, the movements of the ingredient are based on probabilistic rules, embodied as probabilities of moving or not moving during an iteration. We shall now consider the several types of movement rules individually.

2.3.4. The free-moving probability P_M

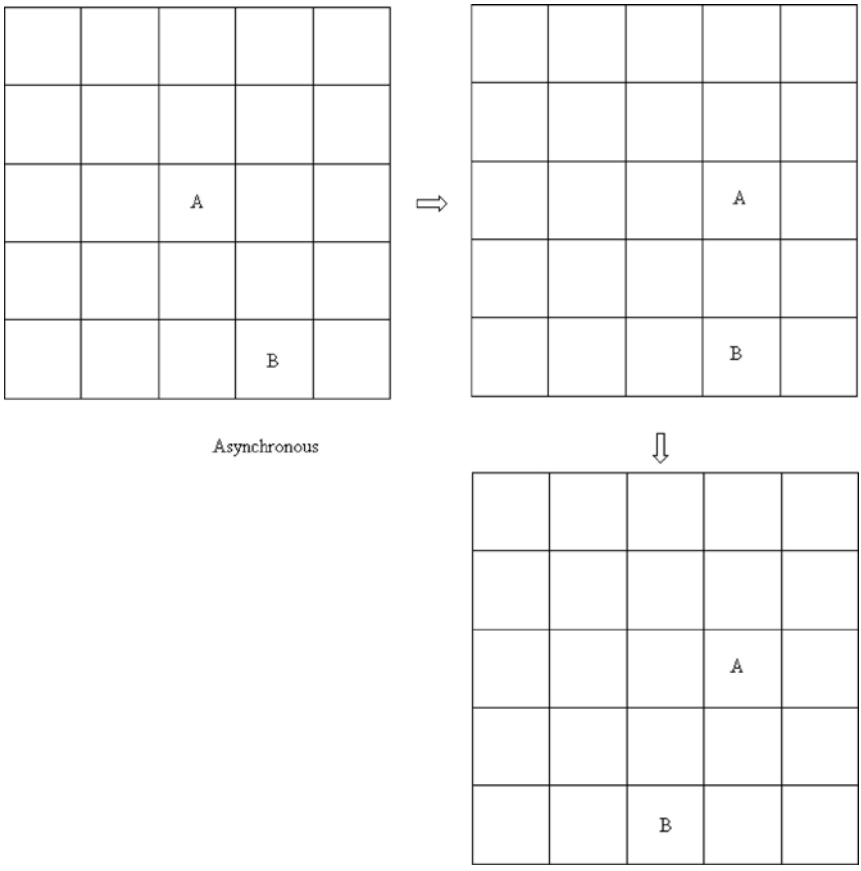
The free-moving probability $P_M(A)$ defines the probability that an ingredient A in a cell, i, will move to one of the four adjacent cells, j, in its von Neumann neighborhood, if that space is unoccupied. An example would be the ingredient A in cell, i, in Figure 2.6a, which might move to any of the unoccupied neighboring cells, j. Two ingredients might move simultaneously as in Figure 2.6a or in sequence as in Figure 2.6b. This will be discussed later. As a matter of course this probability is usually set at $P_M = 1.0$, which means that a movement in one of the allowed directions always happens (a rule). However, in some cases P_M can be set to lower values if certain species in the CA simulation are to be regarded as moving more slowly than others.

2.3.5. The joining parameter J

The first of the two trajectory or interaction rules is the joining trajectory parameter, $J(AB)$, which defines the propensity of movement of an ingredient A toward or away from a second ingredient B, when the two are separated by



(a)



(b)

Figure 2.6. Possible movement of cell *i* occupant A or B to unoccupied cells *j*

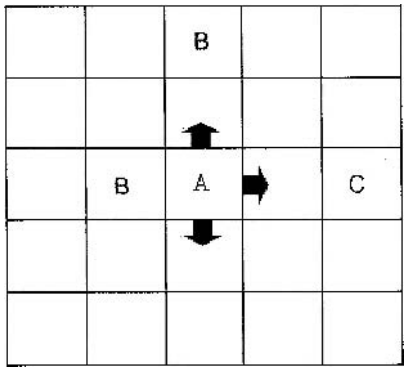


Figure 2.7. Effect of the joining parameter J on the movement of ingredient A in different directions

a vacant cell (Figure 2.7). It thus involves the extended von Neumann neighborhood of ingredient A, and has the effect of adding a short-range attraction or repulsion component to the interaction between ingredients A and B. J is a nonnegative real number. When $J = 1$, species A has the same probability of movement toward or away from B, as when the B cell is not present. When J is greater than 1, ingredient A has a greater probability of movement toward a B ingredient than when ingredient B is absent, simulating, in effect, a degree of short-range attraction. When J lies between 0 and 1, ingredient A has a lower probability of such movement, and this can be considered as a degree of mutual repulsion. When $J = 0$, ingredient A cannot move toward B at all.

2.3.6. The breaking probability P_B

The second trajectory or interaction rule is the breaking probability, P_B. This parameter, in effect, assigns a "stickiness" to the interaction between two ingredients that are in contact, i.e., adjacent to each other on the grid. The breaking rule assigns the probability $P_B(AB)$ that an ingredient A, adjacent to an ingredient B, will break apart from B, as shown in Figure 2.8a. The value for P_B necessarily lies within the range 0–1. Low values of P_B imply a strong cohesion between A and B, whereas high values indicate little cohesion. Thus if $P_B = 0$, the ingredients will not separate from each other, and if $P_B = 1$, they have no tendency to adhere to one another. If P_B lies between these values, there is an intermediate tendency to break apart. When molecule A is bordered by two ingredients, B and B, the simultaneous probability of A breaking away is given by the product $P_B(AB) \times P_B(AB)$, as shown in Figure 2.8b. If ingredient A has three adjacent ingredients (B, B, and B), the simultaneous breaking probability of ingredient A, the probability that it will move to the remaining adjacent empty cell, is $P_B(AB) \times P_B(AB) \times P_B(AB)$, shown in Figure 2.8c. Of

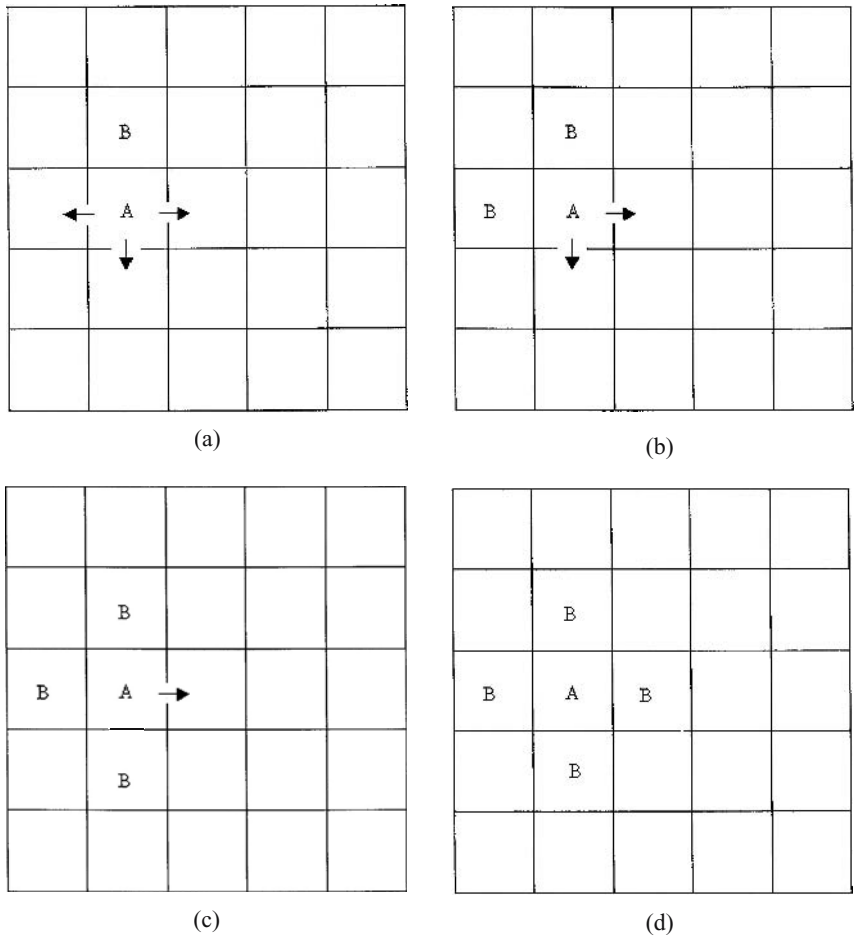


Figure 2.8. The possible directions of ingredient A breaking away from ingredient B: (a) A is bound to one B cell, (b) A is bound to two B cells, (c) A is bound to three B cells, and (d) A is bound to four B cells

course, if ingredient A is surrounded by four ingredients (Figure 2.8d), it cannot move.

2.3.7. Relative gravity rules

The simulation of a "gravity" effect can be introduced into a cellular automaton model in two different ways. Separation phenomena like the demixing of immiscible liquids can be simulated using a relative gravity rule [15]. For this, a boundary condition is first imposed at the upper and lower edges of the grid to apply vertical limits on the motions of the ingredients (a cylindrical grid). The differential effect of gravity on different ingredients A and B is simulated by

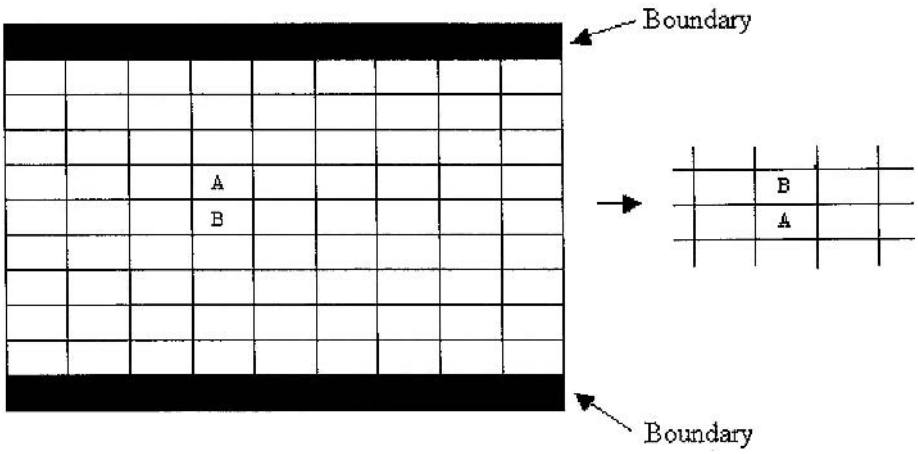


Figure 2.9. Illustration of relative gravity rules influencing cells A and B

introducing reciprocal rules governing their tendencies to exchange positions when they come together. When one ingredient moves to a position on top of the other, the rules are applied. The first rule, $G_R(AB)$, applies when A is above B and is the probability that ingredient A will exchange places with ingredient B, so that A will appear below, and B above. The complementary rule is $G_R(BA)$, which expresses the probability that molecule B, originally above A, will exchange positions with A and end up below. These rules are illustrated in Figure 2.9.

When $G_R(AB)$ is greater than $G_R(BA)$, there will be an overall tendency for the A ingredients to congregate below the B ingredients, and when $G_R(AB)$ is less than $G_R(BA)$, the A ingredients will tend toward the upper part of the grid. In the first case the As can be thought of as forming a more dense liquid than the Bs, and in the latter case, a less dense liquid. The G_R rules are probabilities that the events will occur.

2.3.8. The absolute gravity rule

In other simulations an absolute gravity rule, denoted by $G_A(A)$, is more appropriate [16]. This rule favors motion in a preferred direction. For example, one might wish to simulate the motions of different gas molecules, some heavier than others, in a gravitational field. The value $G_A(A) = 0$ is the neutral value, so that the movement probabilities are equal in all four directions. Values greater than $G_A(A) = 0$ increase the likelihood of downward movements. Thus a value of $G_A(A) = 0.2$ would impose a slight tendency on the ingredients of species A to move downward on the grid, and $G_A(A) = 0.5$ would impose a much stronger tendency.

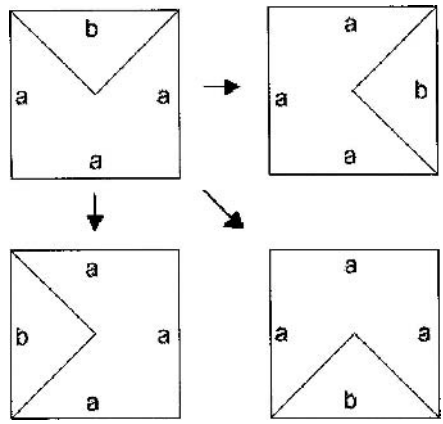


Figure 2.10. Illustration of cell rotation rules

2.3.9. Cell rotation rules

In those cases where variegated ingredients are used (for examples, see Figure 2.4), it is necessary to ensure that there exists a balanced representation of the possible rotational states of these ingredients on the grid. To accomplish this, the variegated cells are rotated randomly, by $-90°$, $+90°$, or $180°$, during every iteration of the run. Only free cells rotate; when a variegated ingredient has a neighboring ingredient in its von Neumann neighborhood, it does not rotate. This rotation process is illustrated for three possible state changes in an iteration in Figure 2.10.

2.3.10. Synchronous or asynchronous application of the rules

A complete time-step (iteration) in a CA model involves the application of all the applicable model rules to all the ingredients on the grid. During an iteration the movement rules can be applied either simultaneously (synchronously), Figure 2.6a, or sequentially (asynchronously), Figure 2.6b. Alert readers will recognize at this point that synchronous application of the governing movement rules for a CA simulation, as outlined above, can lead in some instances to conflicts, e.g., the assigning of two ingredients to move to the same empty cell, just as a similar conflict arises when two cars simultaneously attempt to move into the same vacant parking space. As a result, synchronous rule application is not practical for cellular automata modeling within the framework described here. In asynchronous application of the rules, in contrast, ingredients are selected in random order for application of the movement rules, and such potential conflicts are avoided. Only asynchronous application of the rules to the different ingredients will be used in the applications covered in this book.

The further question of the order to use in applying the different movement rules for a single ingredient does not arise in the present case, since the P_m, $P_B(AB)$, and $J(AB)$ are joined in a single probabilistic equation when determining whether or not a single ingredient will move. The key factor is that any order assigned should be followed consistently and should be reasonable.

2.4. Running a simulation

Having defined both the grid type and size and the governing rules for a simulation, the latter by assigning specific values to the parameters described above, one next needs to define the remaining conditions of the simulation. These include (1) the natures and numbers of the starting ingredients, (2) the configuration of the initial state of the system, (3) how many runs of the simulation are to be carried out, and (4) the length of the runs, i.e., how many iterations they should include.

2.4.1. The initial ingredients

Before beginning, it is necessary to define the starting condition of the system. Here one first declares what types of ingredients should be present at the start of the run and how many of each type should be present. For example, one might wish to work with four species, but start with just two of them on the grid, the others being generated as the system evolves. One might then designate the initial numbers as, say, 250 A ingredients and 500 B ingredients, with zeros for the remaining species C and D. The ingredients are customarily distinguished on the computer screen by different colors. In the present example, the As might be blue, the Bs green, and the Cs and Ds, which are not initially present, red and brown, respectively.

In order for ingredients to move on the grid, there must be empty cells available to accommodate them. For studies of aqueous systems, it has been found that leaving about 31% of the cells on the grid empty provides a reasonable description of the actual condition for motion of the ingredients representing water molecules [17, 18]. Therefore in a water study using a 50×50 grid with 2500 cells, there should be 1725 cells occupied by "water" ingredients and 775 empty cells.

2.4.2. The initial conditions

The default condition for placement of the starting ingredients is to position them randomly on the grid. For some studies, however, a different distribution might be needed. For instance, when one wants to examine the dissolving of a block of ingredients into a surrounding liquid, one might wish to place an array of the ingredients in the center of the grid and then surround them with

the ingredients of the liquid. In another case one might wish to examine the diffusion of a gas into an open area or the passage of ingredients through a membrane, so that appropriate structures for these studies must be constructed as part of the initial configuration on the grid. In the simulations to be described in this and later chapters, any required special constructions or distributions will be set up automatically by the program, so that it will not be necessary for the reader to create them.

2.4.3. The CA runs

It needs to be emphasized at this juncture that when the CA rules are stochastic, i.e., probabilistic, each simulation run is, in effect, an independent "experiment." This means that in principle, the results from separate runs can possibly be quite different, as chance would have it. This, indeed, does occur in some studies, as we shall see. In general, the behavior of a single ingredient is completely unpredictable. However, for most examples, we shall examine which *collective outcome*, either that from a run with a great number of ingredients or that from a great number of runs with few ingredients, tends to display a similar pattern. In the same way that laboratory experiments when repeated tend to yield similar results, so to will CA simulations yield similar patterns. These patterns are the emergent properties associated with the simulations. In the same way, laboratory workers customarily repeat their experiments several times in order to establish the statistical validity of their results, yielding, e.g., an average result and a standard deviation, which is a measure of the experimental error associated with the measurements. This same procedure for the CA runs, repetition of the runs, yields an analogous indication of the uncertainty involved in the simulations. We shall see that as a rule relative error tends to decrease as the sample size increases, or as more runs are employed, just as it does in laboratory experiments.

Accordingly, two further simulation details need to be established, the number of independent runs to be performed and the length (in iterations) of these runs. These numbers will depend very much on the nature of the simulation to be performed. In some cases a relatively short run of, say, 70 iterations, might be enough to make the point needed. But in other, more typical situations, runs of several thousand iterations might be more appropriate, and one might wish to perform several runs in order to establish the uncertainties in any numerical results that appear. In some cases it will be desirable to allow the runs to proceed long enough for some sort of steady-state or equilibrium condition to be achieved. Such a stability point can normally be recognized when the output values exhibit a relatively constant average value over a number of iterations.

It is useful to note here that the CA simulations to be described in this chapter and later chapters tend to be ergodic in the sense that the time average value for a particular property of a single system (i.e., the average taken over a

long time period after the system has reached its steady-state condition) and the ensemble average value for this property (i.e., the average obtained from a large number of runs for the system at a specific time after reaching steady state) are closely identical [19]. Normally it is much simpler to allow a single simulation to run for some time and to obtain property averages from the postequilibrium portion of the run, than to perform a large number of separate runs.

2.5. The output

The output of a CA simulation carried out on a computer comes in two different forms: a visual output that is displayed on the computer screen, and numerical data summaries compiled in output files that are generated during each run. The visual output allows the observer to follow the system as it evolves, and can be very helpful in comprehending the overall process of the system's evolution. The data summaries in the output files are more suitable for quantitative analysis of the details of this evolution.

The numerical data files list the values of specific properties, or attributes, of the system as they change with time. The overall configuration of the system, i.e., the specific arrangement of its ingredients, evolves in time as the cellular automaton rules are applied during each iteration. Accordingly, the system's attributes evolve with time. For kinetic studies the relevant attributes are normally the counts of the different species present and the numbers of the various types of transitions that take place during each iteration. In studies of liquids the numbers of ingredients engaged in different types of "bonding" arrangements are typically listed as they vary with the iterations. The positions on the grid of the different ingredients might also be of interest in some studies. The variations in the system attributes with time can often be related to important macroscopic phenomena taking place in the real systems being simulated.

2.5.1. Liquid simulation outputs

For typical simulations used in the study of aqueous and other liquid systems, several attributes are customarily recorded and used in comparative studies. These attributes used singly or in combination are useful for analyses of different phenomena (Examples of the use and significance of these attributes will be described in later examples). The commonly collected attributes for the liquid systems relate mainly to the states of bonding, i.e., the numbers of adjacent ingredients in the von Neumann neighborhood, of the ingredients. Their designations are as follows:

f_0—fraction of ingredients not bound to other ingredients,

f_1—fraction of ingredients bound to only one other ingredient,

f_2—fraction of ingredients bound to two other ingredients,
f_3—fraction of ingredients bound to three other ingredients, and
f_4—fraction of ingredients bound to four other ingredients.

In addition, the average distance that a cell travels might be another datum collected, as might be information related to the positions on the grid of the different types of ingredients.

2.5.2. Chemical kinetic outputs

In chemical kinetic studies the most relevant attributes are the counts of the various species present and the numbers of transitions of various types that occur during each iteration. For example, in a study of three types of reacting ingredients, A, B, and C, the numbers of each species will change with time, and this variation can reveal important information about the kinetics of the reactions involved. Also informative will be the numbers of transitions, say, from A $\rightarrow$ B and A $\rightarrow$ C, that take place in each iteration.

2.6. Putting it all together

In this section, we will examine four examples that illustrate the steps, procedures, choices, and outputs involved in conducting some elementary cellular automata model simulations. The reader is advised to consult Chapter 10 to find the appropriate ways for entering parameters and making appropriate selections for each study. Following each prearranged example, some brief further studies are indicated that will expand on, and further illustrate, the concepts involved in the example.

Application 2.1. Movement on the grid

Except for the constraints imposed on an ingredient's movements by the presence of other ingredients, or in the special case of a gravitational effect, an ingredient will move randomly about the grid. The pace of its movement is determined by the value assigned to the free movement parameter P_{m}. To illustrate this movement we will start our simulation with a single ingredient placed at the center of a 51 $\times$ 51 cell grid, the starting point being designated as position (0, 0), where the numbers in the parentheses indicate the x and y coordinates, respectively. Following a practice to be used throughout this book, the ingredient will be indicated by a colored cell, with the empty cells white. Individual movements can only occur to adjacent, empty cells. Each step counts as one unit, so that a sequence of steps will normally take the ingredient to a new location on the grid, (x', y'). For example, a sequence of two steps to the right and one downward would lead to the position (2, -1). For the present

example we will use $P_m = 1$, the default setting. Our simulation will involve a single run consisting of 60 steps (iterations).

Example 2.1. Random walk

The Parameter Setup for this Example is shown below. After opening the application CD to Example 2.1 (see Chapter 10 for instructions), the reader should start the simulation run. The ingredient will very rapidly move to some new position. It has executed a "random walk," ending up in some position we can call (x_1, y_1). Start again with the ingredient in the center and observe the result. More than likely, the ingredient will end up in an entirely different place on the grid, at a position we will call (x_2, y_2). Repeat the process several times and note the final positions of the ingredient (For a general discussion of random walks, see Hayes [20]).

Parameter setup for Example 2.1. Random walk

One ingredient placed at the center of a 51 × 51 cell grid on a torus Parameters: $P_M = 1.0$, $P_B(AA) = 1.0$, and $J(AA) = 1.0$ Run for 60 iterations. Repeat three times Report the final values of x, y, and d the displacement for each run Note: The grid dimensions for this example cannot be altered.

The final position of the ingredient after a run consisting of "n" iterations is not predictable. What *is* predictable, however, and quite interesting, is the collective average result after a great number of independent runs. In an unbiased random walk, there should be no overall tendency to move to either the left of the grid or to the right. Therefore the average x displacement $\langle x \rangle$, counting displacements in both the positive and negative x-directions, after many runs of, e.g., 60 iterations each, should logically be $\langle x \rangle_{60} \approx 0.0$. Of course, in a finite set of runs we don't expect $\langle x \rangle_{60}$ to be exactly 0.0, but it should be close to this value, within the uncertainty, or standard deviation, of the experiment. Using the same reasoning, there should be no bias toward upward or downward movement on the grid, and we expect that the average y displacement $\langle y \rangle_{60}$ after 60 iterations should also be near zero, i.e., $\langle y \rangle \approx 0.0$. But—and perhaps this is surprising—the average total displacement $\langle d \rangle = \langle \sqrt{(x^2+y^2)} \rangle = (1/n) \sum_i \sqrt{(x_i^2 + y_i^2)}$ after 60 iterations will not be zero (Can you, dear reader, explain this?). In fact, the average displacement $\langle d \rangle_n$ after n iterations tends to increase as the square root of n: mathematically, $\langle d \rangle_n \approx (\text{constant} = s)\sqrt{n}$, where the constant, s, corresponds to the step size. We now will examine these interesting aspects of the random walk process in our first study.

Physicist George Gamow has playfully pictured a random walk as the path a drunken man might follow in walking away from a light post: first a step in one direction, then a step in another, random, direction, and so on [21]. In fact, random walks play a significant role in natural science [22,23]. They are central to the concept of diffusion of substances in solution [24], and they account for such diverse phenomena as the "Brownian motion" of pollen grains in water, first noted by the botanist Robert Brown in 1827 [25], the migrations of fruit flies from a release point [26], the end-to-end distances of polymers [27], and the diffusion of photons of light from the center of the sun to its surface [21]. In the last case, a straight-line passage of a light photon along the sun's 700,000 km radius would take only about 2 s, but since the photon suffers about 5×10^{21} collisions in its journey from the sun's center, the actual time required for its escape to the surface is roughly 5000 years! [21] Some observers have even compared fluctuations in stock market prices to a random walk [28] (But also see Ref. [29]).

Study 2.1a. Random walk statistics

In this study the reader is introduced to the procedures to be followed in entering parameters into the CA program. For this study we will keep $P_M = 1.0$. We will first carry out 10 runs of 60 iterations each. The exercise described above will be translated into an actual example using the directions in Chapter 10. After the 10-run simulation is completed, determine $\langle x \rangle_{60}$, $\langle y \rangle_{60}$, and $\langle d \rangle_{60}$, along with their respective standard deviations. Do the results of this small sample bear out the expectations presented above? Next, plot $\langle d \rangle_n$ versus $\sqrt{n}$ for $n = 0$, 10, 20, 30, 40, 50, and 60 iterations. What kind of a plot do you get? Determine the "trendline" equation (showing the slope and y-intercept) and the coefficient of determination r^2 (the fraction of the variance accounted for by the model) for this study. Repeat this process using 100 runs. Note that the slope of the trendline should correspond approximately to the step size, $s = 1$, and the y-intercept should be approximately zero.

Note that the dependence of the average absolute distance d on $\sqrt{n}$ is an emergent property that appears only when the results of a significant number of separate runs are collected and analyzed. A single run with one ingredient will not generally exhibit this phenomenon. The emergent dependence of $\langle d \rangle$ on $\sqrt{n}$ does normally arise rather quickly, and the use of only 10 runs often shows a reasonable correspondence between these variables.

Caution

Do not employ too many iterations in your runs (say, more than 200), since eventually some ingredients will move beyond the edges of the torus grid and appear on the opposite sides. Such movements cause the corresponding

distances to be incorrectly calculated (underreported) by the program. You might even wish to test this by increasing the number of iterations and noting the point at which the linear relation between $\langle d \rangle$ and $\sqrt{n}$ no longer holds.

Study 2.1b. Slowing things down

In this example we will repeat the previous study but now slow down the action by reducing the free-moving probability of the ingredient to $P_m = 0.5$. Repeat the same calculations as in Study 2.1a, but now using 100 iterations as the reference time. In this case plot $\langle d \rangle$ against $\sqrt{n}$ for $n = 0, 10, 20, 30, \ldots, 100$. Again, comment on your results. How does $\langle d \rangle$ after 60 iterations compare with the corresponding result from Study 2.1a above? Explain any difference.

Example 2.1a. A one-dimensional random walk

The random walk problem can be simplified by studying motion in just a single dimension. For this parameter setup, open Example 2.1A in the program, *CASim*. Employ a grid consisting of just one horizontal row with 51 cells. The single ingredient should start in the central cell, at $x = 0$. Perform 10 runs of 60 iterations each. Record the 10 individual final positions, the average final position $\langle x \rangle_{50}$, and the standard deviation for your results. Are your results in accordance with expectations? Repeat with 100 runs. In this case use a bar chart to plot the relative final populations of the cells. The results after many runs should converge to a "bell-shaped" (in mathematical terms, "Gaussian") distribution about the central position.

Parameter setup for Example 2.1A. One-dimensional random walk

1 ingredient placed at the center of a 1×51 cell linear grid
Parameters: $P_M = 1.0$, $P_B(\text{AA}) = 1.0$, $J(\text{AA}) = 1.0$
Run length = 60 iterations
First employ 10 runs, then 100 runs
For the 100-run simulation, prepare a bar chart of the relative final positions of the ingredients
Note: For technical reasons, the grid dimensions for this example cannot be changed.

Application 2.2. How other ingredients influence movement on the grid

We have just observed that in the absence of other influences an ingredient will move about the grid in a random walk pattern. Things get more interesting, and more realistic, when the moving ingredient encounters other ingredients

and is either attracted to them or repulsed by them. In this study we will set up a grid and introduce ingredients. For the present, we might regard them as molecules or other things and allow them to move about for a while. We will observe the motions and draw some conclusions from the study.

We begin with a grid that is a square, 100×100 cells. Since we want to model a physical phenomenon, we recognize that our tiny grid of 10,000 cells is an insignificant fragment of what constitutes a bulk physical system. The presence of boundaries, such as the walls of a container, would normally have only a very minimal influence on the few ingredients we will study. Accordingly, we shall set up our model with no boundary conditions. Thus we choose the option of a torus for the grid.

Example 2.2. Ingredient interaction

The next step is to introduce some ingredients: 1000 in the present case for study. We can scatter them about or put them in selected locations. We choose to position the ingredients randomly on the grid. The exact position of each ingredient is identified by reference to its column and row numbers, stored in the program. The "introduction" of ingredients is really the assignment of nonzero states to the cells chosen. We next select an asynchronous computation of the system state and the appropriate transition rules for each ingredient. When each ingredient has in its turn executed its movement and transition rules, an iteration of time has transpired.

The first rule to choose is the probability that each ingredient will move, the free-moving probability P_{M}. In this study we shall choose $P_{\mathrm{M}} = 1.0$, so that the ingredients move during every iteration for which movement is possible. The reader can experiment with this rule by selecting different values, as explained in Chapter 10.

With 1000 ingredients moving about, there will be created at some times a condition in which two of the ingredients will touch, that is, they will occupy adjacent cells. The frequency of such encounters is influenced by the joining rule J, which establishes a tendency for the molecules to move toward, or away from, each other. If two molecules touch, there then occurs the opportunity for them to either separate or remain in contact. This "separation tendency" is established by the second movement rule, the breaking probability P_{B}. Both the J and P_{B} rules have been described above, and they should become very familiar to the reader. If a high incidence of attraction is desired in the model, a high J value should be selected. If a high tendency for the two molecules to separate is desired, a high breaking probability P_{B}, near unity, should be assigned. The behaviors imparted by various combinations of these rules will be markedly different, as the reader will see by experimenting. For our starting example, as a point of reference, we will observe the behavior of the ingredients under "neutral" conditions, i.e., conditions where there is no

tendency to move either toward or away from each other, and no tendency to stick together if they meet. For these conditions we set $P_B(AA) = 1.0$ and $J(AA) = 1.0$. The Parameter setup file shown below will be employed initially.

Parameter setup for Example 2.2. Ingredient interactions

1000 ingredients located randomly on a $100 \times 100 = 10{,}000$ cell torus grid Parameters: $P_M = 1.0$, $P_B(AA) = 1.0$, $J(AA) = 1.0$ Run for 1000 iterations. Observe the motions of the ingredients

How would you describe (qualitatively) the movements of the ingredients under these circumstances?

Studies 2.2a–d. Effects of varying the P_B and J parameters

In the following four studies, the P_B and J parameters are to be varied systematically to reveal their influences on the properties recorded. The reader should now select, sequentially, the following four sets of rules, following the instructions for this in Chapter 10. Compare the behaviors of the molecules in each case with those observed for the neutral conditions prevailing in Example 2.2. Do the ingredients tend to move toward or away from one another? Do they tend to stick together or separate?

Set 1: $J = 2$, $P_B = 0.2$ (Study 2.2a)
Set 2: $J = 0.2$, $P_B = 0.2$ (Study 2.2b)
Set 3: $J = 2$, $P_B = 0.8$ (Study 2.2c)
Set 4: $J = 0.2$, $P_B = 0.8$ (Study 2.2d)

After some intuition about the effects of these rules is acquired, the reader is encouraged to try other combinations.

Application 2.3. Changes of state

This application shows how a cellular automaton can model a phenomenon in which the agents, or ingredients, change their states. This could refer to a chemical reaction, a nuclear decay, or, more generally, any process in which an item converts to another form.

Example 2.3. Changes of state

We shall employ a 20×20 cell grid, this time with all the 400 cells initially occupied and having the same state "A" reflecting their occupied conditions. The state "A" might represent a particular molecular or nuclear species, a specific

quantum state of an atom, or some other entity. In this example movement is irrelevant and will be excluded, so that attention can be focused on the state of each cell, and the state in which transformations can take place. The rule governing these transformations is the transition probability $P_T(AB)$ for the transition A $\rightarrow$ B.

If we select $P_T(AB) = 0.1$, then during each iteration every A ingredient has a 10% chance of changing to state B. During any iteration, therefore, one expects that about 10% of the A ingredients should convert to state B, but as is common for random events with so few ingredients, significant deviations from exact adherence to this schedule are to be expected. The reader is next asked to run the dynamics using the program on the CD. Choose a run of 100 iterations. The numbers of A and B after any iteration can be reckoned by stopping the run and counting the numbers of the A and B ingredients. Alternatively, and much more conveniently, the program automatically keeps a record of the counts of ingredients of each type, as the system evolves. From the numbers of A and B ingredients at different times, a plot can be prepared, showing the decay of the A numbers and the growth of the B numbers. Changes in the $P_T(AB)$ value produce changes in such a plot, an experiment that the reader is encouraged to carry out. The Parameter Setup file shown below will be employed initially.

Parameter setup for Example 2.3. Changes of state

Start with 400 A (blue) cells on a 20 $\times$ 20 grid.
Parameters: $P_M = 0.0$, $P_B(AA) = 1$, $J(AA) = 1$, and $P_T(AB) = 0.1$
Run for 100 iterations
A cells are blue, B cells are green
Repeat three times. Compare the times (number of iterations) at which half of the A cells have switched states to B (this time is called the "half-life" for the transformation A $\rightarrow$ B).

From your knowledge of chemical reactions, you may recognize that the dynamics involved in the above example represent a first-order decay process for the substance A. Modifications can be made to this simple experiment to represent more complicated situations. First, one might want to slow down the conversion of A to B by reducing the value of $P_T(AB)$, and see how this affects the results. Also, by adding a transition rule $P_T(BA)$ for the conversion of species B back to species A, i.e., the process A $\leftarrow$ B, one can model first-order equilibrium. One might also add a transition rule $P_T(BC)$ for conversion of species B to a new species C, so that a series of transitions A $\rightarrow$ B $\rightarrow$ C can be represented. A wide variety of more elaborate sets of transitions can be imagined and conducted with this general setup. A more detailed

and quantitative examination of these first-order kinetic processes will be presented in Chapter 7, but here we will examine some of the basic features of such reactions.

Study 2.3a. Slowing the rate

For this study you are asked to change the $P_T(AB)$ value to $P_T(AB) = 0.05$ and compare the results to those in the example. Because of the lower transition probability employed, you should now also extend the run length to 500 iterations to assure that most ingredients will convert from A to B during the course of a run. Try to focus on some individual cells and note when they convert from A to B.

Study 2.3b. Adding a back reaction A ← B

Most chemical reactions are reversible, at least to some extent. We will use $P_T(AB) = 0.05$ again, and start with all of the cells in the "A" state. To add the back reaction B → A to the forward reaction A → B, we need to add a transition probability $P_T(BA)$ for the B → A transition. We will initially take $P_T(BA) = 0.08$ and observe the results, again after 500 iterations. How many A ingredients remain after 500 iterations? How many remained in Study 2.3a?

Now run this experiment three times and note the numbers of A and B remaining after each run. Let [A] equal the number of A remaining at the end of the run and [B] the number of B. From your three runs, determine average values and standard deviations for these numbers. We can define the equilibrium constant for the interconversion of A and B as $K_{eq} = [B]/[A]$. Determine an average value and standard deviation for this ratio based on your results. What value would you expect for K_{eq} based on the transition probabilities? Does your calculated value from the simulations agree with this value?

Application 2.4. Introducing gravity

A gravitational field introduces a directional preference into the movements of the ingredients on the grid. This preference could just as well reflect the influence of any type of gradient, for example, an electric field, a chemical gradient, or a light gradient that might affect the movements of an agent. To investigate this influence we will again study a simple neutral system first, and then introduce the influence we wish to examine, noting its effect.

Example 2.4. Absolute gravity

We will start with a collection of 300 ingredients, free to move about a cylindrical 50 × 50 cell grid. We will assume that they have no tendencies to stick together or move toward or away from each other. A run of 1000 iterations

will give us some time to observe the motion. The Parameter Setup file shown below will be employed initially.

Parameter setup for Example 2.4. Absolute gravity

300 ingredients placed randomly on a 50 × 50 cell horizontal (H) cylindrical grid. Parameters: $P_M = 1.0$, $P_B(AA) = 1.0$, $J(AA) = 1.0$, and $G_A(A) = 0.0$ Run for 1000 iterations. Observe the motions of the ingredients (same as Example 2.2).

Note your general impressions of the motions of the ingredients. (This need not be detailed, just a general impression.)

Study 2.4. Adding a gravitational effect

Now you should introduce a gravity effect into the simulation by adjusting the absolute gravity parameter $G_A(A)$. First set $G_A(A) = 0.15$, and have the simulation run for 2000 iterations. Record your general impression of what happens. Next try $G_A(A) = 0.3$ and repeat the experiment (You might think of these two cases as representing, in a very loose way, the gravitational situations on Earth and on the planet Jupiter, respectively). Report your general observations.

2.7. Cellular automata models

All models are simplifications, abstract representations of real systems. In cellular automata models we employ very simple tools: cells on a grid, some occupied, some not, and a set of fairly simple rules. We postulate correspondences between the ingredients on the grid and certain objects in nature. In chemistry these objects are often molecules, but in a broader sense they might be almost anything; ants in an ant colony, bacteria in a solution, cars in traffic, flowers on a field, or people in society. The ingredients represented by occupied cells do not possess the elaborate structural details usually associated with the real objects of interest. In CA models these structural details and their associated equations are replaced by simple local rules for the interactions between the ingredients. In the case of a square cell, the rules define how the four faces of an ingredient will interact with objects in its environment. In this way a CA model provides a set of constraints on the dynamic encounters and state changes undergone by ingredients that are only defined relationally, i.e., by their interactions. The same is true to a large extent; of course, for the more elaborate traditional models discussed in Chapter 1. Indeed, many models in science rely on abstract

equations and completely ignore the structural details and the interactions involved. As we have seen in the studies above, the rules define the affinities that the ingredients have for one another and the probabilities that ingredients will transform to new species.

The CA paradigm thus represents a drastic simplification, and the question arises whether such a model yields sufficiently realistic representations of natural phenomena to be useful. Our aim in this book is to answer this question. We will do so by examining the applications of CA models to a variety of natural systems. As primitive as the basic notions employed might seem to the first-time observer, experience shows that these notions encapsulate to a surprising extent the universal features responsible for the behaviors observed for ingredients in natural systems. This is, indeed, the secret of any successful model that it recognizes and properly incorporates the major elements affecting behavior. The inherent simplicity of CA models provides two great advantages: this simplicity permits a clear view of the roles of different features responsible for the system's behavior, and it allows rather complicated systems to be studied using only modest computational resources.

One of the most important considerations of a scientist planning a research project is experimental design. The design process begins with several questions: What do I assemble in the lab (or on the computer)? What features do I control (model)? What observations should I make (calculate)? How many times should I repeat the experiment (runs)? What relationship does what I am going to do have to understanding nature (applicability)? There are accordingly a number of practical considerations that go into the planning of a CA experiment, and the following chapters will illustrate these considerations. The initial conditions, the starting pattern on the grid, and the applied rules should bear a sensible relationship to the real system being examined. The configurations of the CA system should, as they evolve, supply a useful portrayal of the actual chemical or physical phenomenon that is being investigated. In any successful model the key known features of the physical, chemical, biological, or other process should be properly replicated.

With these caveats satisfied, CA models contain a further potential: they allow investigators an opportunity to go beyond the simple replication of natural phenomena and carry out "what if?" experiments. What happens, for instance, if a certain rule is changed or if the starting ingredients are altered? *In silico* explorations based on such questions are relatively easy to perform and can lead to new insights and the revelation of heretofore unrecognized features and relationships. Indeed, CA models, through their use of independent rules, are capable of parsing out dependencies and relationships that are not readily separable or distinguishable in normal physical and chemical experiments. For example, one can change the temperature of the solvent in a simulation while leaving the temperature of its solute unchanged, thus revealing the isolated

influence of solvent temperature on a process, independent of other factors. Or one can follow the movement of a single ingredient in a collection of identical ingredients—a feature extremely difficult, and perhaps impossible, to follow in a traditional laboratory experiment. Put briefly, beyond their ability to duplicate what is already known, a virtue in itself, the CA models described in the following chapters, may, in the hands of creative readers, also provide a means for discovering previously unknown aspects of nature. It is with this hope that we have written this book, and we trust that readers will carry CA "experiments" into new fields that the authors have not even dreamed of.

References

1. S. Wolfram, Cellular automata. *Los Alamos Sci.* 1983, 9, 2–21.
2. S. M. Ulam, Random processes and transformations. *Proc. Int. Congr. Math*. 1952 (held in 1950), 2, 264.
3. S. M. Ulam, *Adventures of a Mathematician*. Charles Scribner's Sons, New York, 1976.
4. J. von Neumann, in *Theory of Self-Replicating Automata*, A. Burks, ed. University of Illinois Press, Urbana, 1966.
5. K. Zuse, The computing universe. *Int. J. Theor. Phys*. 1982, 21, 589.
6. T. Toffoli and N. Margolus, *Cellular Automata Machines*. The MIT Press, Cambridge, MA, 1987.
7. M. Schroeder, *Fractals, Chaos, Power Laws*. W. H. Freeman, New York, 1991, p. 371.
8. J. Signorini, Complex computing with cellular automata. In *Cellular Automata and Modeling of Complex Physical Systems*, P. Manneville, N. Boccara, G. Y. Vishniac, and R. Bidaux, eds. Springer-Verlag, New York, 1990, 57–70.
9. G. Y. Vichniac, Simulating physics with cellular automata. *Physica D.* 1984, 10, 96–116.
10. T. Toffoli, Cellular automata as an alternative to (rather than an approximation of) differential equations in modeling physics. *Physica D*. 1984, 10, 117–127.
11. S. Wolfram, Preface. *Physica D*. 1984, 10, vii–xii.
12. S. Wolfram, *A New Kind of Science*. Wolfram Media, Champaign, IL, 2002.
13. S. Kauffman, Emergent properties in random complex Automata. *Physica D*. 1984, 10, 145–156.
14. L. B. Kier, C.-K. Cheng, and B. Testa, A cellular automata model of micelle formation. *Pharm. Res*. 1996, 13, 1419–1422.
15. C.-K. Cheng and L. B. Kier, A cellular automata model of oil-water partitioning. *J. Chem. Inf. Comput. Sci*. 1995, 35, 1054–1059.
16. L. B. Kier, C.-K. Cheng, and H. T. Karnes, A cellular automata model of chromatography. *Biomed. Chromatogr*. 2000, 14, 530–534.
17. L. B. Kier and C.-K. Cheng, A Cellular Automata Model of Water. *J. Chem. Inf. Comput. Sci*. 1994, 34, 647.
18. L. B. Kier, C.-K. Cheng, and P. G. Seybold, Cellular automata models of aqueous solution systems. In *Reviews in Computational Chemistry*, vol. 17, K. B. Lipkowitz and D. B. Boyd, eds. Wiley-VCH, New York, 2001, 205–254.
19. P. G. Seybold, L. B. Kier, and C.-K. Cheng, Simulation of first-order chemical kinetics using cellular automata. *J. Chem. Inf. Comput. Sci*. 1997, 37, 386–391.
20. B. Hayes, How to avoid yourself. *Am. Sci.* 1988, 86, 314–319.

21. G. Gamow, *One, Two, Three, . . . Infinity*. Dover Publ., New York, 1988.
22. H. C. Berg, *Random Walks in Biology*. Princeton University Press, Princeton, NJ, 1983.
23. G. H. Weiss, Random walks and their applications. *Am. Sci.* 1983, 71, 65–71.
24. M. F. Shlesinger and J. Klafter, Random walks in liquids. *J. Phys. Chem*. 1989, 93, 7023–7026.
25. B. H. Lavenda, Brownian motion. *Sci. Am.* 1985, 252(2), 70–85.
26. J. R. Powell and T. Dobzhansky, How far do flies fly? *Am. Sci.* 1976, 10, 179–185.
27. G. Slade, Random walks. *Am. Sci.* 1996, 84, 146–153.
28. B. G. Malkiel, *A Random Walk Down Wall Street*, 7th ed. Norton, New York 1999.
29. A. W. Lo and A. C. MacKinlay, *A Non-Random Walk Down Wall Street*. Princeton University Press, Princeton, NJ, 1999.

Chapter 3

WATER AS A SYSTEM

I loved touching water. Physically. Sensually. Water fascinated me.
—Jacques-Yves Cousteau [1]

It has long been recognized that water plays an essential part in chemical events, especially those associated with biological phenomena. It is viewed as an active participant in complex systems such as solutions of ingredients necessary for life. It is important to add to the understanding of this essential fluid and the complex systems it forms when solutes enter its embrace. When solution ingredients interact, there emerges in the system a set of properties not clearly recognizable as additive contributions from the ingredients. There forms a complex system characterized by emergent properties. The subjects of complexity and emergent properties in chemical research have recently been reviewed by Kier and Testa [2]. The complex nature of water and solutions has been recognized in recent times and has prompted some investigators to derive models based on nonlinear combinations of ingredients [3]. In particular, we have witnessed the growth of Monte Carlo and molecular dynamic simulations of water that have added to our understanding of its complex character [4]. Molecular dynamics has provided an approach toward a better understanding of water as a complex system; however, the large amounts of computer time required coupled with its assumptions of specific force fields produce certain limitations. This has led to the study of water and solution phenomena using cellular automata models of dynamic processes.

A current model of water is that of an extended network of hydrogen-bonded molecules that lack a single, identifiable, long-lived structure. The hydrogen bonds continually form and break producing a constantly changing mosaic when viewed at the molecular system level. This model lends itself to simulations using dynamic methods such as cellular automata. In a recent study, Kier and Cheng [5] have created such a model of liquid water using rules governing the joining and breaking of water-designated cells.

3.1. The dimensionality of the model

In models used for studies of aqueous phenomena, the organization of liquid water molecules is assumed to follow the pattern of the hexagonal ice lattice (Figure 3.1a). Each vertex in that figure denotes a water molecule, while each edge denotes a bonding relationship. This three-dimensional network can be dissected into a contiguous series of vertices arranged tetrahedrally around a central molecule (Figure 3.1b).

Some or all of the vertices in each fragment may be representative of a water molecule. The trace of each fragment may be mapped onto a two-dimensional grid (Figure 3.1c). This trace is equated with the mapping of a cellular automaton von Neumann neighborhood. The cellular automata transition rules operate randomly and asynchronously on the central cell, i, in each von

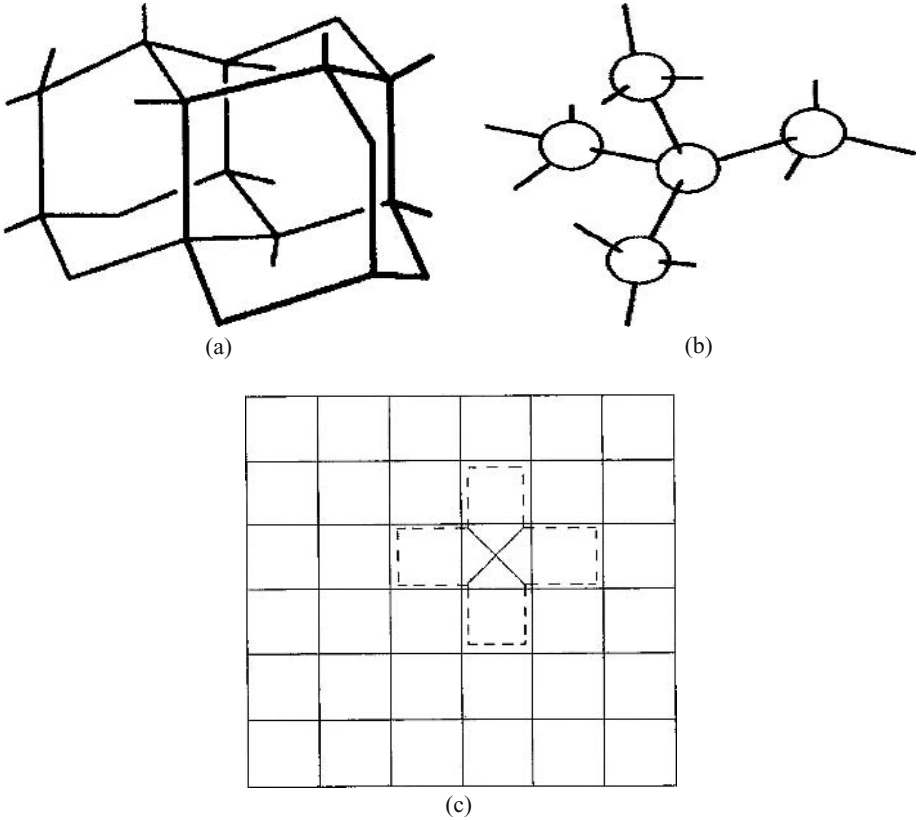


Figure 3.1. (a) Hexagonal array of water molecules in the solid state. (b) Tetrahedral arrangement of bound water molecules. (c) Trace of the tetrahedral arrangement if there are five bound water molecules on a surface. This mapping is equivalent to the von Neumann neighborhood

Neumann neighborhood. As a consequence, the new configuration for each cell, i, and its neighborhood is derived independently of all other cells outside this neighborhood. The configuration of the system, achieved after all cells respond in random order to the rules, constitutes one iteration. This configuration is a composite of the collective configurations achieved in all of the von Neumann neighborhoods. Each of these neighborhoods is a two-dimensional mapping of a tetrahedral fragment of the original three-dimensional model. The model is a representation of the configuration of a three-dimensional system on the basis of it being an ensemble of discrete, separate events occurring within that system. A number of studies using this approximation have been reported [6].

3.2. Experimental design

The square cell is convenient for a model of water because water is quadrivalent in a hydrogen-bonded network (Figure 3.2). Each face of a cell can model the presence of a lone-pair orbital on an oxygen atom or a hydrogen atom. Kier and Cheng have adopted this platform in studies of water and solution phenomena [5]. In most of those studies, the faces of a cell modeling water were undifferentiated, that is no distinction was made as to which face was a lone pair and which was a hydrogen atom. The reactivity of each water cell was modeled as a consequence of a uniform distribution of structural features around the cell.

3.3. The grid

The size of the grid can vary, but it should be large enough to allow a statistically consistent set of results to be gathered. On the other hand, a very large grid slows the calculation time, but this may not be a real problem for

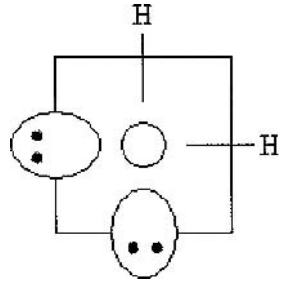


Figure 3.2. The quadrivalent structure of a water molecule with two hydrogen atoms and two lone pairs of electrons

a fast computer. A 40 × 40 cell grid is probably the minimum size for water and solution studies, while a 100 × 100 grid should satisfy any space needs for numerous ingredients and should ensure reproducibility. In many of our studies we have used a 55 × 55 grid. The grid should be boundary free for most studies since the walls of the water vessels are not usually the focus of attention in a study. In order to achieve movement modeling reality, the computing of states should be asynchronous. The run times should be long enough to achieve relatively constant values of the attributes collected.

3.4. The representation of water

Evidence shows that bulk water contains a significant amount of free space referred to as cavities or voids. It is obvious that water could not permit the diffusion of solutes through it if there was no space between water molecules. In ice, this is not the case, water molecules are bound to approximately four other water molecules. The choice of how many water molecules should be represented on a grid of a certain size was explored by Kier and Cheng [5]. Two approaches were taken. The first approach was on the basis of estimates of the volume of a water molecule and the number of water molecules in a mole, an estimate of about 69% occupancy of a grid was deduced. The second approach was to conduct CA runs with varying water concentrations. The attributes of the CA configuration were interpreted and compared with experimental values. For example, after a sufficient number of runs, the average number of cells joining each cell was recorded. This attribute was judged to be a model of the average number of hydrogen bonds per molecule of water. A good correspondence of these two values was found for a water concentration of about 69% of the grid cells. Another attribute from these experiments, the number of free, unbound water molecules, was recorded. This small percentage of the total number of waters was compared with the number of free waters from the experiment. The best correspondence was found for a CA system containing about 69% water molecules in the grid. From this information, a system modeling water was adopted using this percent of water molecules in the grid.

3.5. Water density consideration

Adopting a grid density of 69% water molecules is a reasonable model, but it is known that the density of water decreases slightly with increased temperature of above 4°C. It might be necessary in a few studies to refine the model of water density in a CA grid to conform to the experimental observation. This is easily done by reducing the grid density by a decrement corresponding to a lower experimental density at a given temperature. Table 3.1 shows the

Table 3.1. Water density and water cell concentration for water temperatures

$100 \times P_B$(WW)	Specific gravity	Concentration (cells) of water[a]
10	1.0000	2099
20	0.9982	2096
30	0.9957	2091
40	0.9922	2084
50	0.9881	2075
60	0.9832	2065
70	0.9778	2053
80	0.9718	2041
90	0.9653	2027

[a] Number of water cells in a 55×55 grid.

water temperature, the corresponding specific gravity, and the adjusted number of water cells for a 55×55 cell grid. Example 3.3, below, examines this issue.

3.6. The movement rules

Three rules must be chosen to impart a "water character" to the occupied cells that we designate. The first of these is the movement probability P_m. There is no practical reason to believe that anything other than $P_m = 1.0$ for water has any real significance. This choice characterizes water as a freely moving molecule whenever it is possible. The other two rules governing the joining and breaking of water molecules are critical to their behavior and to the emergent attributes of the CA system. Recall that the joining rule, J, encodes the probabilities of water molecules to join others to form a bond (a hydrogen bond in the case of water). The breaking probability, P_B, describes the tendency of bound waters to break apart. The selection of these rules is essential if the model is to have any validity.

3.7. The J and P_B rules for water

The linkage between rules J and P_B can be made relative to a range of values of one of them. As described earlier, the P_B value ranges from zero to one; therefore, the J value may be chosen as a function of P_B. Later, the wisdom of this choice can be tested by comparing the attributes of the system with the physical properties. The studies of Kier et al. [7] led to a relationship as shown in Eq. (3.1).

$$\log J = -1.5\ P_B + 0.6 \tag{3.1}$$

These rules are shown in Table 3.2 for several values of P_B. The f_x attributes computed for different values of P_B and the corresponding J rules are plotted

Table 3.2. Breaking (P_B) and joining (J) rules for the equation used for water ($\log J = -1.5\ P_B + 0.6$)

P_B(WW)	J(WW)
0.05	3.35
0.10	2.80
0.15	2.37
0.20	2.00
0.25	1.68
0.30	1.40
0.35	1.19
0.40	1.00
0.45	0.84
0.50	0.71
0.55	0.60
0.60	0.50
0.65	0.42
0.70	0.36
0.75	0.30
0.80	0.25
0.85	0.21
0.90	0.18
0.95	0.15

in Figure 3.3. If we assign a "temperature" of the water to a P_B value according to the relationship of Eq. (3.2),

$$T(^{\circ}C) = 100\ P_B \tag{3.2}$$

then the sets of f_x values can be related to the temperatures of water. From this relationship, selected physical properties at different temperatures can be related to f_x values at those temperatures [8]. An analysis of several properties demonstrates the relationship with selected f_x values and the general validity of the CA model of water. Some of these analyses are shown in Table 3.3. The conclusion that selected values of the movement rules from Eq. (3.1) produce a meaningful profile of f_x values, makes it possible to proceed with some confidence that this CA model of water has validity.

At this point in the discussion of experimental design, it is necessary to reflect on what is known about water molecules in terms of their propensity to join and break apart. A condition in which many water molecules join and few break apart is a state of the system associated with cold water. In contrast, the condition in which there are less joined water molecules and more isolated molecules is a state expected of warmer water.

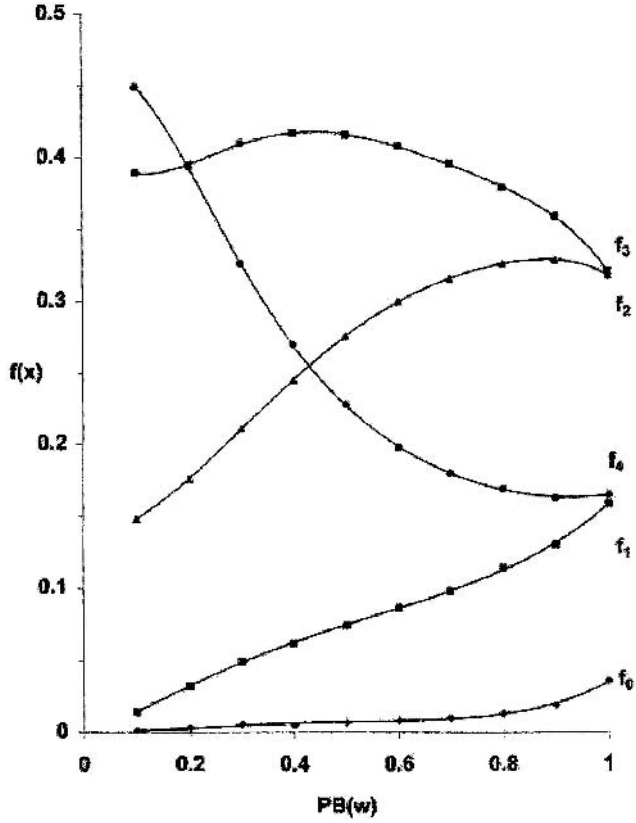


Figure 3.3. A plot of relative bound water types, f_x, formed at various P_B parameter values

This intuition leads us to recognize that the J and P_B rules must be complimentary and that they may model states of water associated with temperature. The choice of these two rules must therefore lead to the attributes of the CA dynamics that have some relationship to the physical properties that change with temperature. These properties then become references for the validity of

Table 3.3. Water properties related to cellular automata attributes

Property	Equation	r^2 correlation
Vapor pressure	$\log P_v$ (mm Hg) $= 13.77\ (f_0 + f_1) + 0.795$	0.987
Dielectric constant	$\varepsilon = -224\ f_1 + 86.9$	0.989
Viscosity	η (centipoise) $= 3.165\ f_4 - 0.187$	0.989
Ionization	$-\log K_w = -20.94 f_H + 16.43$	0.999
Surface tension	γ (dynes/cm) $= 16.07\ N_{HB} + 22.35$	0.970
Compressibility	κ ($\times 10^6$/Bar) $= -53.82\ f_3 + 66.66$	0.953

the model. The choice of J and P_B must produce attributes in the CA dynamics, which can be recorded for this validation.

3.8. The attributes recorded

A CA run leads to a configuration that is constant in an average sense. Several attributes of this configuration may be recorded and used for further study. The configuration may be analyzed for the numbers of molecules with no neighbor, one neighbor, and so on up to four neighbors. The fraction of each state can be represented as shown in Figure 3. The distribution of these fractional values becomes a profile of the state of a molecule. It is observed that these states change with different J and P_B rules and that these states have some correspondence to the temperature of the system.

Another attribute of CA dynamics is the average size of clusters of cells. This is certainly influenced by the choice of J and P_B rules and should parallel some physical properties. Other attributes that can be measured and used to link with physical properties include the average number of joined faces of a molecule and the number of free cell faces. Each of these attributes has been exploited for their ability to relate to the physical properties. The choice of J and P_B rules, therefore, must be made with some attention to their attributes produced and their ability to mirror physical properties as a function of systematic rule changes.

3.9. Statistical considerations

An important issue in these studies is the number of runs necessary to achieve a reproducible value for an attribute and how many iterations are necessary for each run. These questions do not have obvious answers since the emergent attributes from any study are not predictable from a simple sum of the attributes of the ingredients. It is important to test both the necessary length of time for a meaningful run and how many runs should be made.

3.10. Other liquids

So far we have focused on models reproducing properties of water. What about other solvents? Can we manipulate the J and P_B rules to achieve a model that reflects the properties of a liquid other than water? This is an unexplored territory, but we will address the issue here in terms of what a useful approach may be.

The description of molecules such as water in a CA experiment is a blend of rules that are relational rather than intrinsic. The rules describe how a molecule

may interact with another, not what the molecule is by itself. Any variation from a description of a molecule to characterize another molecule, must go through the medium of the rules that we have introduced. It may be possible to exploit this characteristic of a system in the CA method to study a molecule other than water. The rules create the conditions for the attributes of the model to emerge. The attributes are studied for their ability to relate to physical properties in order to validate the model. The logical place to begin the attempt to model a liquid other than water is to look at a list of properties of liquids that includes water.

In an earlier work, Kier and Cheng [5] have shown a close correlation between the viscosity of water, η (centipoise), and the f_4 fraction calculated in a CA model, at a range of temperatures (Figure 3.4). The relation is

$$\eta\,(\text{centipoise}) = 3.165\,f_4 - 0.187$$
$$r^2 = 0.987$$

The experimental and predicted values of η (centipoise) are shown in Figure 3.4. From this close correlation we infer that the f_4 structural attribute is influential on the viscosity. This relationship leads to the expectation that a second liquid could be defined in part by the f_4 attribute, at least as far as the

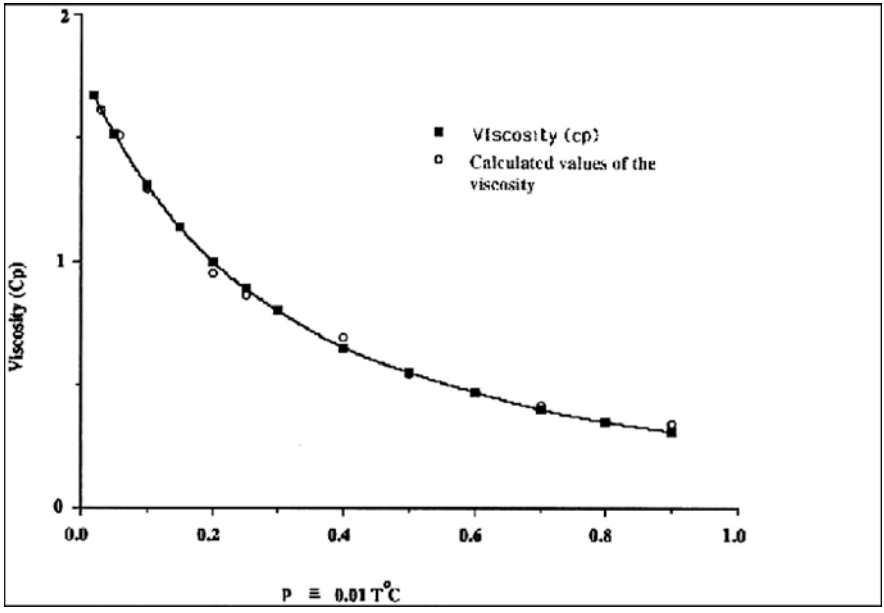


Figure 3.4. A plot of water viscosity (in centipoise) versus the fraction of f_4 water types

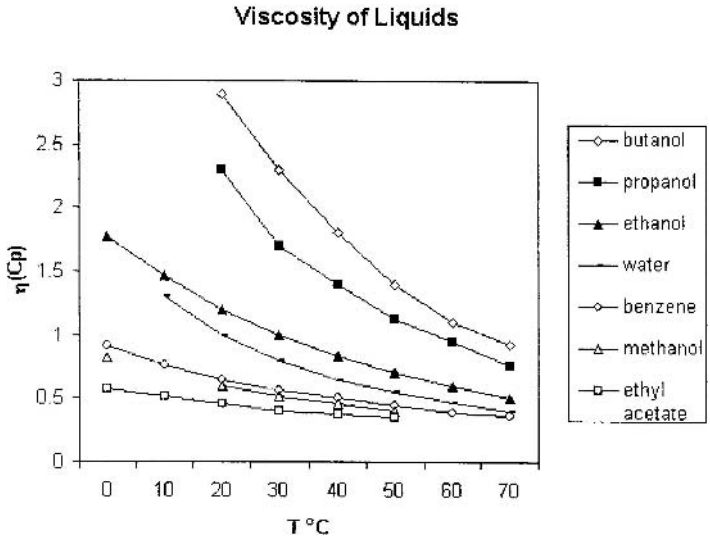


Figure 3.5. Plot of viscosity of several liquids as a function of the temperature

property of viscosity is concerned. A comparison of the viscosities of some common liquids versus temperature is also shown in Figure 3.5.

The value of f_4 in a system of molecules is due to the rules selected to run the CA dynamics. Therefore, the equation relating the J and P_B rules is the target for study and change. If we write Eq. (3.1) in general terms we get

$$\log J = \alpha P_B + \beta \tag{3.3}$$

It is the α and β coefficients that must be manipulated to obtain a new set of attributes that correspond to a liquid other than water. Of course, this relationship is only with the value of the viscosity, but it may relate to other properties.

If we look at Figure 3.5, we see that some liquids present a curve of viscosity versus temperature above that of water while others lie below. By systematically varying the α and β values in Eq. (3.3) and using these to calculate the f_4 values versus the temperature, we obtain the information about their influence on the viscosity versus temperature relationship. Those plots that lie below the water curve in Figure 3.5 may represent the rules needed to model liquids such as benzene, methanol, or ethyl acetate. The α and β coefficients that produce f_4 values leading to plots above water in Figure 3.5 are candidates for the modeling of liquids such as ethanol, propanol, or butanol.

This approach leads to a model of a liquid other than water that has a calculated viscosity different from water but that may be all that is needed for a particular study. It should be noted that the vapor pressure presents a similar

opportunity to create a model of another liquid, this one is based on the attributes f_0, f_1, and f_2. Much more could be done along these lines to model other liquids, an opportunity for the reader to participate in some interesting research.

Application 3.1. The structure of water at various temperatures

Water exhibits variations in physical property values when the temperature is varied. The underlying cause of this variation is the change in the structure of water when the temperature changes. What is the structure of water and how can it be revealed and defined in a quantitative way? One approach to this has been described above. An amenable method is to relate the profile of f_x values with the temperature. This profile is shown in Figure 3.3. With this profile, it is possible to define a state at a certain temperature by selecting the parameters corresponding to a given temperature and then running the CA dynamics to obtain these results. From there it is possible to simulate a temperature in another study. The following example and studies introduce the reader to this approach to the modeling of water.

Example 3.1. Model of water at room temperature (20°C)

Run a model of water using the parameters shown in Example 3.1 in the Program CASim.

Parameter setup for Example 3.1. Model of water at room temperature

Grid 55 × 55 on a torus
2100 A (blue) cells
$P_M(A) = 1.0$, $P_B(AA) = 0.2$, $J(AA) = 2.0$
10 runs at 1000 iterations each run
Record the average values of f_0, f_1, f_2, f_3, f_4, and the average cluster size.

Run the dynamics for 1000 iterations or until you are satisfied that the run is producing a relatively constant set of attributes. Repeat the run 10 times and compute the average values for each f_x attribute. Convert each f_x value to a fraction of the total number of water molecules, 2100 in this case. The sum of the f_x values then equals 1.0. These fractions become a structural profile of water at any simulated temperature.

Study 3.1a. Model of "hot" water

Repeat the runs described above but change the values to hot water, $P_B(AA) = 0.8$ and $J(AA) = 0.25$. Notice the change in the water clusters

and the cavity behavior. Compare the attributes described by the f_x profiles for the "cold" water model with those of the "hot" water model. Note the change in the average cluster size.

Study 3.1b. Other temperatures of water

Using the parameter setups for Example 3.1, change the "temperature" of the water by changing the P_B(WW) and J(WW) values according to the relationships shown in Table 3.2. For example, use $P_B(WW) = 0.50$ and $J(WW) = 0.71$. Run several of these "temperatures" from hot to cold water and collect the f_x attributes. Convert the f_x values to a fraction of 1.00 and then plot the set of f_x values versus the "temperature." This set of relational values is a set of structures of water at different simulated temperatures (see Figure 3.3). They can be used as independent variables to explore the relationships of water versus various physical properties at different temperatures as shown in Table 3.2.

Application 3.2. Density-adjusted grid occupancy

We have described above the choice of a concentration of water-designated cells to mirror the density of water. Using the f_x values over the temperature range of liquid water, we have achieved success in modeling many physical properties of water. We have used a uniform density in these studies. In fact, the density of water varies slightly over the liquid temperature range as seen in Table 3.1. Strict adherence to this variation would lead us to vary the grid occupancy by an amount corresponding to the variation shown in Table 3.1. In this example, the variation is built into the parameters for runs to compare with the results from Example 3.1.

Example 3.2. Density-adjusted grid occupancy for a water model

In this study the concentration of the water in the grid is adjusted to correspond to the density (specific gravity) of water at various temperatures. These adjusted concentrations are shown in Table 3.1. Run the example 3.2 for the adjusted concentrations of water in Example 3.2 in the program CASim for a temperature of 20°C.

Parameter setup 3.2. (Example 3.2) Density-adjusted "cold" water

Grid 55 × 55 on a torus 2096 A (blue) cells $P_M(A) = 1.0$, $P_B(AA) = 0.2$, $J(AA) = 2.0$ 10 runs at 1000 iterations each run Record the average values of f_0, f_1, f_2, f_3, f_4, and the average cluster size.

Study 3.2a. Density-adjusted "hot" water

Repeat Example 3.2 using a simulated temperature of 80°C using $P_B(AA) = 0.8$ and $J(AA) = 0.25$. Change the water concentration according to Table 3.1. Collect the f_x values and convert them to a fraction of 1.0. Compare these values with those from Study 3.1.

Study 3.2b. Density adjusted for intermediate temperatures of water

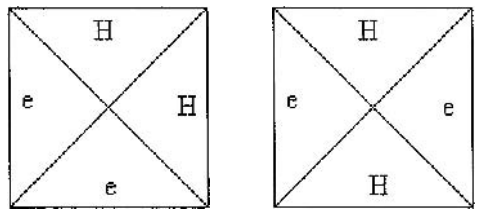
Repeat the studies, selecting intermediate temperature values using the relationships between P_B and J values for water shown in Table 3.2. For example, use $P_B(WW) = 0.50$ and $J(WW) = 0.71$ in Example 3.2**.** Compare the f_x values with the Studies in Examples 3.1and 3.2. A plot of each $f_x$value from Studies in 3.1 and 3.2 will reveal the influence on these attributes with and without a density consideration. Also compare the average cluster sizes between these two groups of studies. How much difference is found in the f_x values when water density is accounted for?

Application 3.3. Use of a variegated water cell

In this exercise, a comparison is made between a water cell with the same set of parameters for each face of the cell and a water cell with a different set of parameters reflecting the difference between a hydrogen and a lone pair of electrons on water. This latter case makes use of the variegated cell architecture described in Chapter 2, Figure 2.4. In the variegated cell, we assume that the two water molecules will bond only through the encounter of a hydrogen atom, H, and a lone pair of electrons, e, of an oxygen atom. The encounters of two hydrogens or two lone-pair orbitals are modeled as an event contributing nothing to the bond formation.

Example 3.3. Variegated water cell models

This example uses the variegated cell pattern aabb shown in Figure 3.6. Use Example 3.3 in the program CASim. Results from this example are compared



(a) (b)

Figure 3.6a and b. A variegated cell depicting two patterns equivalent to the mapping of a tetrahedrally bound water molecule as shown in Figure 3.2

with the attributes that emerge from the dynamics using uniform cells in Examples 3.1 and 3.2. A temperature of 20°C is used in this example.

Parameter setup 3.3. (Example) Water modeled with a variegated cell of aabb pattern

Grid 55 × 55 on a torus
2100 variegated A cells. The faces of A are designated aabb
$P_M(A) = 1.0$, $P_B(aa) = 1.0$, $J(aa) = 0$, $P_B(bb) = 1.0$, $J(bb) = 0$, $P_B(ab) = 0.2$, $J(ab) = 2.0$
10 runs of 1000 iterations each run
Record the average f_0, f_1, f_2, f_3, f_4 values, and convert them to fractions of 1.0
Record the average cluster size.

Study 3.3. Hot water modeled using pattern aabb

Repeat the dynamics using a temperature simulation of 80°C. Compare these results with the results from Studies 3.1a and 3.2a. An interesting question for the student to contemplate is whether the use of the alternative variegated cell, abab, would produce different results in these studies. Reach a conclusion as to whether the more detailed aabb cell has any significant differences from the uniform cell employed in Applications 1 and 2. The future use of the variegated cell may be important when different features of the same molecule are to be encoded. More on this in later chapters.

Application 3.4. Self-diffusion

It is possible to model what a single molecule of water is doing in the presence of bulk water. It is necessary to distinguish one molecule as being different, say a different color and identity within the program. The state and the movement rules are kept the same for this molecule as for the rest of the water molecules. The movement of the designated water molecule away from a reference point as a function of time is the diffusion rate.

Example 3.4. Water self-diffusion

A single water molecule is identified by a different color, but it has the same states and trajectory rules as that of the other water molecules. In this ways, its behavior can be singled out for separate recording. This is the procedure necessary to evaluate a property such as self-diffusion. In this study, evaluate the average distance of movement of the designated cells after a common number of iterations.

Parameter setup 3.4. (Example 3.4) Water self-diffusion at 20°C

Grid 55 × 55 on a torus
2099 A (blue) cells
1 B (red) cell located at row 27, column 27
$P_M(A) = P_M(B) = 1.0$, $P_B(AA) = P_B(AB) = 0.2$, $J(AA) = J(AB) = 2.0$
10 runs at 1000 iterations each run
Record the average distance that cell B travels from the starting position after 1000 iterations.

Study 3.4. Water self-diffusion at various temperatures

In this study the water temperature is changed to other values using the $P_B(WW)$ and $J(WW)$ parameters shown in Table 3.2. A profile of self-diffusion as a function of temperature can be derived from these results.

Application 3.5. Evaporation of water

Modeling of water may be extended to properties involving the movement of molecules into space, a process of evaporation. For this the grid must be structured at the initial setting to have two different areas, one with occupied cells and the other with unoccupied cells (Figure 3.7). The rate of evaporation can be measured from a model allowing for water movement into an empty part of the grid. This is illustrated in Example 3.5.

Example 3.5. Evaporation of water

Water molecules are placed in the lower half of the grid, leaving the upper half empty. A temperature is selected using the P_B and J parameters and the CA is allowed to run for a specified time. The number of water molecules in each row of the upper half of the grid is counted. The grid is defined as a cylinder with the upper and lower boundaries stationary. This prevents water movement past the bottom boundary. A profile of evaporation versus temperature can be obtained by varying the simulated temperature. Use Example 3.5 in the Program CASim.

Parameter setup 3.5. (Example) Evaporation of cold water

Grid 55 × 55 cells in a cylinder (top and bottom boundaries)
1050 cells W located in the lower half of the grid
$P_M = 1.0$, $P_B(WW) = 0.3$, $J(WW) = 1.4$
Run 1000 iterations for 20 runs
Average, every 100 iterations, the counts of cells in each row in the upper half of the grid.

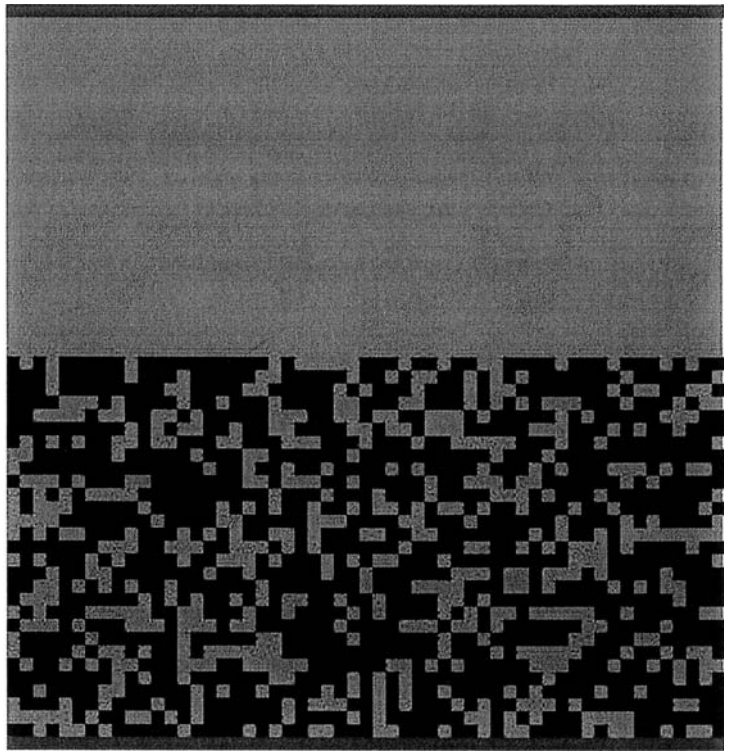


Figure 3.7. A grid organized to show the lower half occupied with water. The grid is bounded on all four sides.

Plot this average value against the distance from the original interface. Calculate the rate of evaporation.

Study 3.5a. Evaporation of hot water

In this study, the water molecules are heated up using P_B (WW) = 0.70 and J(WW) = 0.36. The rate of evaporation is recorded at intervals of the iteration count. Compare these results with the results from Example 3.5.

Study 3.5b. Evaporation of water at various temperatures

Repeat this study using various intermediate temperatures by altering the P_B(WW) and J(WW) values in Parameter Setup 3.5. From the recorded data it is possible to compute the rate of evaporation. In later chapters we introduce a second ingredient into the grid. At that time the student is encouraged to repeat these studies using more than one ingredient, varying their parameters to obtain differential evaporation or distillation.

Application 3.6. Modeling physical properties with structure, f_x values, of water

We have defined above a way of quantifying the structure of water based on the profile of f_x values that encode the number of each possible joined state of a molecule. It is now possible to use this profile as a measure of the structure of water at different temperatures. As an application of this metric it is possible to relate this to physical properties. We have shown the results of our earlier work in Table 3.3. The reader is encouraged to repeat these and to explore other structure–property relationships using the f_x as single or multiple variables. A unified parameter derived from the five f_x values expressed as a fraction of 1.0, might be the Shannon information content. This could be calculated from all the data created in the above studies and used as a single variable in the analysis of water and other liquid properties.

References

1. G. Jonas, *The New York Times*. June 26, 1997, p. A16.
2. L. B. Kier and B. Testa, Complexity and emergence in drug research. *Adv. Drug Res.* 1995, 26, 1–22.
3. T. H. Plumridge and R. D. Waigh, Water structure theory and some implications for drug design. *J. Pharm. Pharmacol.* 2002, 54, 1155–79.
4. L. M. Haile, *Molecular Dynamics Simulation*. John Wiley, New York, 1992.
5. L. B. Kier and C.-K. Cheng, A cellular automata model of water. *J. Chem. Inf. Comput. Sci.* 1994, 34, 647.
6. L. B. Kier, C.-K. Cheng, and B. Testa, Cellular automata models of biochemical phenomena. *Future Gen. Comput. Sys.* 1999, 16, 273.
7. L. B. Kier, C.-K. Cheng, and P. Seybold, Cellular automata models of chemical systems. *SAR QSAR Environ. Res*. 2000, 11, 79–98.
8. L. B. Kier, C.-K. Cheng, and P. Seybold, Cellular automata models of aqueous solution systems. *Rev. Comput. Chem*. 2001, 17, 205–254.

Chapter 4

SOLUTION SYSTEMS

Complexity is a maddeningly slippery concept. . . . The cellular automaton is to the study of complexity what E. coli or a planarian is to biology—a relatively simple preparation used to open a window on perplexing phenomena.

—George Johnson [1]

Water becomes much more interesting when there is something in it. Forming a solution is what water does best on this planet, and so we address this complex system as a focus for cellular automata modeling. In the previous chapter we developed approaches to the CA modeling of water. Critical issues such as the grid density, movement rules, and attribute collection were laid out. In this chapter, we introduce an ingredient, usually a solute, into water. This consideration raises new issues that must be considered in the experimental design and data collection. Several examples will illustrate this aspect of the chemical applications of CA models.

4.1. The rules

In a cellular automata model of a solution, there are three different types of cells with their states encoded. The first is the empty space or voids among the molecules. These are designated to have a state of zero; hence, they perform no further action. The second type of cell is the water molecule. We have described the rules governing its action in the previous chapter. The third type of cell in the solution is the cell modeling a solute molecule. It must be identified with a state value separate from that of water.

Recall that the rules for water molecules are the J and P_B rules, influencing the movement toward and away from each other. Their effect is to produce a dynamic system modeling liquid water. We will adopt a standard protocol for the naming of these rules since the variety increases significantly when modeling

solutions. The J rule for water–water encounters is designated as J(WW). Similarly, for the breaking rule for two water molecules bound together we adopt the descriptor, P_B(WW). Recall that the P_B(WW) rule can range from zero to one and that it has a close correspondence to the water temperature in °C. The J(WW) value is a ratio of probabilities that have a wide range of positive values.

When we introduce a new molecule type into the system, it is designed to be different from water by virtue of its rules. The new rules pertaining to the solute must not only govern its joining and breaking actions among like molecules, but they must also govern its actions with water molecules. Thus encounters between solute and solvent molecules are made possible and influenced by these heterogeneous rules. The behavior of the solute in the solutions is dependent on these rules.

4.2. The solute rules

The interactions among solute molecules are a reflection of their chemical structure. The manifestations of this structural influence are the physical properties associated with their intermolecular binding. As examples, the melting point reflects the ease of crystal disruption to bring about a state change. The solubility is another phenomenon derived from the structural influences on the binding of solute molecules in a crystal.

The breaking and joining rules pertaining to the solute may then be assigned on the basis of some anticipation of the magnitude of these intermolecular events [2]. A large value of the joining rule, J(SS) where S denotes the solute molecule, encodes a high incidence of joining among the molecules. In contrast, a low value of J(SS) implies a low possibility of solute molecules coming together. The companion rule, P_B(SS), encodes the ease or difficulty in breaking a pair of joined solute molecules. To illustrate the cooperativety of these two rules for a solute, a molecule with rules $J(\mathrm{SS}) = 2.0$ and $P_B(\mathrm{SS}) = 0.2$ would be expected to remain as an aggregate, a crystal, for most of the time. Its expected properties would be a high-melting point and relative insolubility. In contrast, the solute molecule with rules $J(\mathrm{SS}) = 0.5$ and $P_B(\mathrm{SS}) = 0.8$ would likely have a lower melting point and would be more soluble. The choice of these rules dictates the general behavior of the model of solute molecules among themselves.

4.3. The solute–water encounter rules

The solute molecules are going to encounter water molecules, an event that is governed by J and P_B values encoding this. The joining and breaking rules

for a water–solute molecule encounter, $J(\mathrm{WS})$ and $P_{\mathrm{B}}(\mathrm{WS})$, are assigned on the basis of the choice of behavior for the solute. This behavior manifests itself in the physical property of solubility. A solute molecule with rules of $J(\mathrm{WS}) = 1.5$ and $P_{\mathrm{B}}(\mathrm{WS}) = 0.3$ would be expected to interact with water resulting in joining of the two molecules. In contrast, a solute molecule with rules $J(\mathrm{WS}) = 0.3$ and $P_{\mathrm{B}}(\mathrm{WS}) = 0.8$ would be likely to have limited joining events with water. These behaviors model the relative solubility and insolubility of a solute molecule. The character of the solute molecule, sometimes referred to as its polarity, is thus encoded in these rules. Kier et al. used these rules to study the behavior of a solute in the water surrounding it [3].

Application 4.1. Modeling aqueous solutions

A 55 × 55 grid is used with 2100 water cells, corresponding to a density of 69%. A number of solute molecules are then added. If 100 solute molecules are used, then this number would be subtracted from the 2100 water molecules to maintain 69% cells in the grid. The assumption is made that the volume of all molecules in the grid is about 69%. This assumes that the dissolution in this study produces an overall expansion of the volume of the system to 3125 occupied cells, but we are modeling only 3025 of these as water cells. Volume expansion on addition of a solute is recognized, but it may not be a universal phenomenon. The reader is invited to explore this concept.

Example 4.1. The influence of water temperature on solubility

This example is designed to measure some attributes reflecting a dissolved state, specifically to model the effect of water temperature on the solubility. The assumption is made that the concentration of solute molecules not bound to other solute molecules characterizes the dissolved state. The attribute used to reflect this state is the f_0 value for solute molecules. The higher this value, the more dispersed are the solute molecules among the water molecules. The average cluster size is also a significant attribute characterizing a solution. The water temperatures used in this example is 30°C. The solute component is represented by 100 cells. The polarity of the solute molecule is described by $P_{\mathrm{B}}(\mathrm{WS}) = 0.5$, $J(\mathrm{WS}) = 0.7$, $P_{\mathrm{B}}(\mathrm{SS}) = 0.5$, and $J(\mathrm{SS}) = 0.7$. At each water temperature, the attributes of the solute, f_0, cluster size, and number of solute–solute bonds are recorded. These attributes are interpreted as modeling the extent of dissolution. It should be noted that only the temperature of the water changed in this study, no changes in other rules are made. The changes recorded in the attributes are emergent properties of a new system, a solution. Run these values 10 times and average the values of the attributes collected.

Parameter setups for Example 4.1. The influence of water temperature on solubility

Grid 55 × 55 cells on a torus
Water, W (blue), 2000 cells
Solute, S (red), 100 cells
$P_M(W) = 1.0$, $P_B(S) = 1.0$
$P_B(WW) = 0.3$, $J(WW) = 1.4$, $P_B(SS) = 0.2$, $J(SS) = 2.0$, $P_B(WS) = 0.5$, $J(WS) = 0.7$
Run the dynamics for 1000 iterations, for 10 runs
Record the average of the largest cluster size of the solutes, S, and the average f_x values for the solute, S.

Studies 4.1a and 4.1b. Effect of water temperatures of 60°C and 90°C on solubility

Repeat this example in the following studies, for water temperatures simulating 60°C, $P_B(WW) = 0.6$, $J(WW) = 0.5$ and 90°C, $P_B(WW) = 0.9$, $J(WW) = 0.18$. Also record the f_x values and the average cluster size for the solute. See Table 4.1 for some results previously reported [2].

Table 4.1. Effect of water temperature on solubility

Simulated water temperature (°C)	$f_0(S)$[a]	Average cluster size	Average number of solute bonds
30	0.11	4.5	2.00
60	0.33	2.0	1.25
90	0.44	1.5	1.00

[a] Fraction of solute molecules not bound to other solute molecules.

Example 4.2. The influence of solute concentration on water structure

In this example, the effect of increasing the solute concentration on the water structure is explored [4]. The attributes of water $f_0(W)$ and the average number of water–water bonds are measured. This latter attribute corresponds to the average number of hydrogen bonds. Again in this study, the change in a single variable, the solute concentration, results in an emergent set of properties associated with the water structure. Run these parameters 10 times and average the values of the attributes collected.

Parameter setup Example 4.2. The influence of concentration

Grid 55 × 55 cells on a torus
Water, W (blue), 2080 cells
Solute, S (red), 20 cells
$P_M(W) = 1.0$, $P_M(S) = 1.0$
$P_B(WW) = 0.3$, $J(WW) = 1.4$, $P_B(SS) = 0.5$, $J(SS) = 0.7$, $P_B(WS) = 0.5$, $J(WS) = 0.7$
Run the dynamics for 200 iterations, for 50 runs
Record the average f_x values for water, W.

Studies 4.2a and 4.2b. Alternate concentrations

Repeat this example using 2060 water cells and 40 solute cells in the Example 4.2 Parameter Setup. This is approximately a 2% solution. Repeat the dynamics again with a higher concentration such as 2020 water cells and 80 solute cells, using Example 4.2 Parameter Setup. Compare the structures of water as characterized by their f_x profiles and average cluster sizes. Some measures of the structure change in water as a function of the concentration are shown in Table 4.2.

Table 4.2. Effect of solute concentration on water structure

% Solute	f_0(W)[a]	Average number of hydrogen bonds
0	0.004	3.013
1.8	0.005	3.008
3.5	0.006	2.993
5.2	0.007	2.928
6.9	0.008	2.906

[a] Fraction of water molecules not bonded to other water molecules.

Application 4.3. The hydrophobic effect

The hydrophobic effect of a solute is its influence on nearby water molecules resulting in an increased organization or structure change. This organization manifests itself as a greater degree of water–water hydrogen bonding producing an increased microviscosity in this region. The hydrophobic effect results from the influence of a soluble but relatively hydrophobic (nonpolar) solute. Kier et al. have modeled this phenomenon using cellular automata [3].

4.4. The influence of the solute–water rules on water structure

A study by Kier et al. explored the influence of the solute–water rules on the water structure [3]. The rules governing this relationship impart to the solute what is called the hydropathic state. This is the relative degree of water affinity or rejection. A $P_B(WS)$ value of 0.2 implies a strong association between the water and the solute. The solute is said to be "polar" or "hydrophilic." In contrast, a $P_B(WS)$ of 0.9 is a rule describing a very "nonpolar" solute, one that is hydrophobic. These two terms constitute the relative hydropathic state of a solute. The study by Kier et al. used water at 25°C, a solute with $P_B(SS) = 0.10$, $J(SS) = 1.0$, and $P_B(WS)$ values ranging from 0.2 to 0.8; thus, a large range of hydropathic states of the solute was explored. The influence on the water structure was measured as the number of hydrogen bonds and the $f_4(W)$ value. Some of these results are shown in Table 4.3. They reveal that a decreasingly polar solute, in the presence of water, produces an increasing degree of organization or structure in the water. This corresponds to the hydrophobic effect. From the graphics exhibited by solutes of varying degrees of hydrophobicity, the hydrophobic effect can be recognized. In the case of a relatively polar solute say $P_B(WS) = 0.2$ and $J(WS) = 1.0$, the solute molecules are largely distributed among the clusters of water as seen in Figure 4.1a. This figure demonstrates the hydration of polar solutes. If we change the water–solute rules to $P_B(WS) = 0.8$ and $J(WS) = 0.5$, this relatively nonpolar solute is observed to be away from the interior of water clusters and is found most of the time in the cavities (Figure 4.1b). This figure demonstrates the hydrophobic effect. The reader is encouraged to set up these rules and to observe the dynamic behavior leading to this effect. To monitor the progress toward this effect, record the f_4 fraction at regular intervals until it is roughly constant.

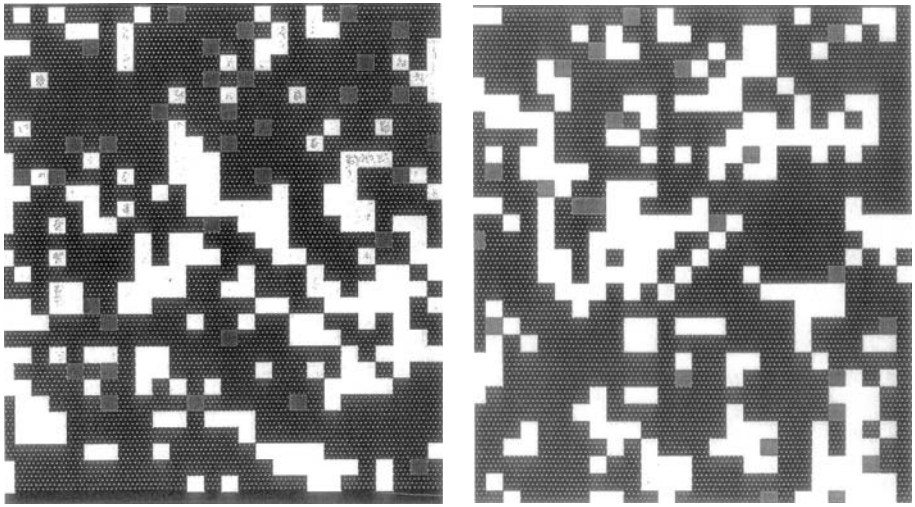
Table 4.3. Influence of solute polarity on water structure

$P_B(WS)$	$f_4(W)$[a]	Average number of hydrogen bonds
0.2	0.269	2.89
0.4	0.298	2.93
0.6	0.313	2.97
0.8	0.320	2.97

[a] Fraction of water molecules bonded to four other molecules.

Example 4.3. The hydrophobic effect

Use the parameter setups described in the Example 4.3 with a water temperature (WW) of 20°C and the rules for the hydropathic state of the solute of



(a) (b)

Figure 4.1. (a) A cellular automata model of hydrophilic solutes in water. (b) A cellular automata model of hydrophobic solutes in water

$P_B(WS) = 0.2$ and $J(WS) = 2.0$. Run the dynamics for 1000 iterations for 10 runs. Collect the average f_x values and the average cluster size for water.

Parameter setup Example 4.3. The hydrophobic effect

Grid 55 × 55 cells on a torus
Water 2020 cells (blue), Solute 80 cell (red)
$P_M(W) = 1.0$, $P_M(S) = 1.0$ $P_B(WW) = 0.2$, $J(WW) = 2.0$, $P_B(SS) = 0.5$, $J(SS) = 0.7$, $P_B(WS) = 0.2$, $J(WS) = 2.0$
Run the dynamics for 200 iterations for 10 runs
Record the average f_x values for W and the average cluster size for W at the end of each run.

Studies 4.3a and 4.3b. Variation of P_B and J rules for (WS) on the hydrophobic effect

Increase the nonpolar character of the solute by using rules $P_B(WS) = 0.5$ and $J(WS) = 0.7$. Keep all the other rules constant. Run each experiment 10 times and collect and average the f_4 values. Repeat the study using a more nonpolar parameter set for the solute, for example $P_B(WS) = 0.8$ and $J(WS) = 0.25$. Other parameters are retained as in Example 4.3. Record the f_x values and the average cluster size for water at the end of each run.

Figure 4.2. A cellular automata model of a crystal in water

Application 4.4. Solute dissolution

Another aspect of the complex system called a solution is its formation from the interaction of the ingredients, water and a solute. Kier and Cheng have modeled the dissolution of a solute crystal into water using CA [5]. The crystal was modeled as a compact group of cells in the center of a grid (Figure 4.2). Surrounding it is a field of water molecules dispersed as a 69% concentration in the remaining part of the grid. Several sets of rules were used in multiple studies to examine the influences on the crystal disruption leading to dissolution. The extent of dissolution was monitored by the change in the number of solute–solute bonds and the average distance that the solute molecules moved from the crystal. The fewer the number of solute–solute bonds, the greater the extent of solute dissolution. Similarly, the greater the separation of solute cells from the original crystal mass, the greater the dissolution. It was observed that an increase in the water temperature produced an increase in the rate of formation of the isolated (f_0) solute molecules. This corresponds to the generally observed temperature effect on dissolution.

The question as to which rule has the greatest influence on the dissolution, P_B(SS) or P_B(WS), was addressed in another study. Again using the f_0 (solute) attribute as an indication of the extent of dissolution, the results from variations in these two rule sets are compared. The studies indicated that the P_B(SS) and J(SS) rules had the greatest influence. This result suggest that the self-affinity of a solute, reflected by the P_B(SS) value, is a greater determinant of the solubility than the hydropathic state, reflected by the P_B(WS) value [5]. The validity of these findings is open to debate.

Example 4.4. Solute dissolution

The reader is invited to examine this phenomenon by running the models described above, by varying these two sets of parameters. The solute is modeled as a 10 × 10 block of 100 cells in the center of a 55 × 55 cell grid. The water content of the grid is 69% of the spaces around the solute block, randomly placed at the beginning of each run. The water temperature (WW), solute–solute affinity (SS), and hydropathic character of the solute (WS) are presented in the parameter setup for Example 4.4. The extent of dissolution as a function of the rules and time (5000 iterations) is recorded as the f_0 and the average cluster size of the solute (S).

Parameter setup Example 4.4. Solute dissolution

Grid 55 × 55 cells on a torus
Water, W, 2000 cells
Solute, S, 100 cells in a 10 × 10 block in the center of the grid
P_M(W) = 1.0, P_M(S) = 1.0. P_B(WW) = 0.3, J(WW) = 1.4, P_B(SS) = 0.5, J(SS) = 0.7, P_B(WS) = 0.5, J(WS) = 0.7
Run for 1000 iterations
Record the average distance of S from the center of the grid, (row 27, column 27).

Study 4.4a. Variation of solute self-affinity (SS) effect on dissolution

These studies vary the relative aggregation of the solute encoded in the P_B and J values for (SS). Select several sets of these parameters using the setup in Example 4.4, run the dynamics for 1000 iterations, and record the average distance the S molecules have migrated from the center of the grid.

Study 4.4b. Variation of hydropathic state on dissolution

This study looks at the effect of changes in the solute hydropathic state (WS) on the dissolution of the solute. The P_B and J values for (WS) determine

this attribute of the solute. Using the setup in Example 4.4, vary the P_B(WS) and J(WS) values. Record the average distance the S molecules have migrated from the center of the grid. Another variable not studied by Kier and Cheng is the influence due to the water temperature. The problem now becomes a three-dimensional one with these three variables. Have a look at this problem.

Observing the dissolution process

So far, most of the observations of CA dynamics that we have described have arisen from an inspection of data reported in the form of counts of structural types as a function of time, temperature, or joining and breaking rules. It is also possible to make some observations, even discoveries, by observing directly the evolving picture of the dynamics. Watching the crystal "dissolve" in these studies reveals that the process does not take place because of the intrusion of water molecules into the solute–crystal mass. What occurs is the "intrusion" of cavities into the crystal structure. These move about as though they were real entities; they grow in number until the crystal is blown apart. At this point there is a major intrusion of water molecules into the crystal. The reader is invited to run these experiments and make this observation. A typical scene is shown in Figure 4.3, and on the cover of the book. How close this model is to reality remains to be seen.

Application 4.5. Solute diffusion in water

The diffusion of solutes in water is an important event in many biological processes. The influences of water temperature and hydropathic states of the solute are expected to be of importance in this process. A study modeling diffusion using CA was reported by Kier et al. [6]. The study revealed increases in diffusion rates with higher temperatures and higher solute hydrophobicity. More recent studies indicate that the diffusion rate may be maximum at an intermediate level of hydrophobicity and temperature [7].

Example 4.5. Diffusion in water

The rate of diffusion of a solute in water may be studied as a function of the temperature of water and the hydropathic state of the solute. For this experiment, the setup for Example 4.4 can be used, with a single cell positioned in the center of the grid. The distance between the center of the grid and the average position of the solute cell is recorded at every 100 iterations. This gives the rate of diffusion. Varying the rules for (SS) and (WS) give models of the influences of the solute characteristics on the diffusion rate. Use the following parameter setups to model these effects.

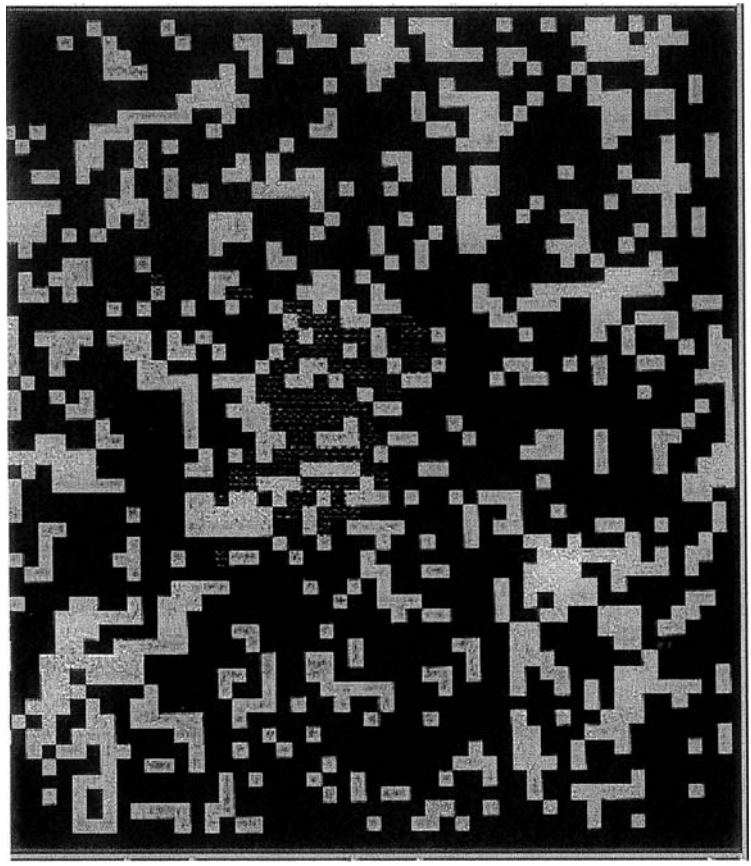


Figure 4.3. A cellular automata model of a partially dissolved crystal in water

Parameter setup for Example 4.5. Diffusion in water

Grid 55 × 55 cells
Water, W, 2099 cells (blue)
Solute, S, 1 cell (red). Initial position is in the center of the grid
$P_M(W) = 1.0$, $P_M(S) = 1.0$ $P_B(WW) = 0.3$, $J(WW) = 1.4$, $P_B(WS) = 0.5$, $J(WS) = 0.7$
Run the dynamics for 1000 iterations
Run the simulation for 50 times
Caution: Adjust the number of iterations so that the S cell does not pass over an edge to the opposite side of the grid. Test the best choice of the number of iterations, then record the average distance that S travels from the initial position.

Study 4.5a,b. Influence of variation in hydropathic state of the solute on diffusion

By varying the hydropathic states, $P_B(WS)$, of the solutes, it is possible to model this influence on the rate of diffusion. Repeat Example 4.5 in the following studies to gather data on this influence. Using the parameter setup in Example 4.5, study the effect of solute polarity, as reflected by its hydropathic state, on the diffusion characteristics in water. Modify the parameters for (WS), for example, choose $P_B(WS) = 0.2$ and $J(WS) = 2.0$ as descriptors of a high state of solute polarity. Another, more nonpolar state of the solute may be selected using parameters $P_B(WS) = 0.8$ and $J(WS) = 0.25$. Run these and other sets of hydropathic state rules to obtain some insight into the influence on diffusion. Record the average extent of diffusion from the center of the grid, after each 100 iterations.

Study 4.5c. Influence of temperature on solute diffusion

Using Example 4.5**,** vary the temperature of the water using $P_B(WW)$ and $J(WW)$ values from Table 3.2 in Chapter 3. Remember that temperature in degrees C $= 100 \times P_B(WW)$. From the Studies 4.5a and b, create a profile of the diffusion of a solute in water as a function of water temperature and solute hydropathic states.

Application 4.6. Freezing point depression

It is known that the introduction of a polar solute into water will produce an effect lowering the freezing point of water. The extent of the lowering is related to the concentration of the solute. This effect can be modeled with CA even though the phase change to a solid state is not a part of the model. In Chapter 3, Figure 3.3, the fractions of bound states of water, f_x, are shown as a function of water temperature. A change in this pattern of f_x values corresponds to a change in the temperature. We can say that the pattern of f_x values is a system structural description of what is modeled as temperature. If a solute that is introduced changes this pattern of f_x values from those corresponding to 20°C to those corresponding to 25°C, then the freezing point has been lowered by this temperature increment.

An important question is, how can we quantify the f_x pattern so that it can be numerically compared with temperature? One way is to consider the ensemble of f_x values as information describing the system. By design, $\Sigma f_x = 1.00$ holds for this set of values and so the f_x values should be converted to fractions of 1.00. The information content, H, is a value that is

Table 4.4. Information content of water cell types at different temperatures

	Shannon information content
Temperature as P_B(W)	$-\Sigma f_x \log f_x$
0.1	0.467
0.2	0.507
0.3	0.536
0.4	0.548
0.5	0.558
0.6	0.563
0.7	0.570
0.8	0.581
0.9	0.595

calculated using the Shannon equation, Eq. (4.1).

$$H = -\Sigma f_x \log f_x \tag{4.1}$$

As an example, consider the f_x values for water at 20°C, in Figure 3.3 of Chapter 3. The information content is

$$H = -f_0 \log f_0 + f_1 \log f_1 + f_2 \log f_2 + f_3 \log f_3 + f_4 \log f_4 = 0.507$$

If we calculate the H values for various water temperatures, we see results as shown in Table 4.4. The importance of the information content encoded in the H value in these studies is that it is a single-numerical description of the system, water in this case, that can be used to relate to physical property changes occurring at different temperatures. This approach can be used to evaluate a property change such as the freezing point depression.

Using the information content, H, to describe the structure at any temperature, it is possible to estimate the "new" temperature of water when a solute has been added. An increase in this temperature corresponds to the freezing point depression because the water must experience a greater decrease in temperature in order to arrive at the point of solidification.

Example 4.6. Modeling the freezing point depression due to a solute

In this example, the attributes associated with water and solutions containing various concentrations of a polar solute are calculated and transformed into "effective temperatures." The comparisons between these two values are reckoned to be a model of the depression in the freezing point of the system. The

rules in Example 4.6 are used as the reference state in modeling the influence on the water structure in the presence of a polar solute, S.

Parameter setup Example 4.6. Reference f_x profile

Grid 55 × 55 cells
Water 2100 cells (blue)
Solute 0 cells (red)
$P_M(W) = 1.0$, $P_M(S) = 1.0$ $P_B(WW) = 0.20$, $J(WW) = 2.0$, $P_B(SS) = 0.5$, $J(SS) = 0.7$, $P_B(WS) = 0.2$, $J(WS) = 2.0$
Run the dynamics for 200 iterations, 10 times
Record the average f_x values for water, then convert these to fractions of 1.00, then calculate the Shannon information content.

Study 4.6. Influence of a solute on the freezing point

The rules in Example 4.6 are used to estimate the effective temperature resulting from the presence of a solute. In this study, replace 30 water molecules with 30 solute molecules. Use parameters for these solute molecules reflecting a moderately polar character, such as $P_B(SS) = 0.5$, $J(SS) = 0.7$ and $P_B(WS) = 0.2$, and $J(WS) = 2.0$. Run the dynamics and collect the f_x values for the water. Convert these f_x values as fractions of 1.00. Compute the Shannon information content, H, for this set of parameters.

From these results it is possible to make another estimate of a property of the solution system. It is known that the freezing point of a solvent is lowered by approximately 1.86°C for every mole of the solute present. From the estimates of the temperature of the solvent and the solution modeled above, the decrease in the temperature can be estimated. From this value, the number of cells comprising a "mole" of solute may be reckoned. Thus, a value may be stated for an imaginary "molecular weight" of the cells used in the study.

4.5. A model of solute aggregation

At the molecular level in biology, the solution is the common method of transport of nutrients and wastes throughout the organism. The solution is a system, ubiquitous in every living being; it may be thought of as a part of the definition of "living." Recognizing its importance, we may ask some searching questions about its character; when is a substance "dissolved"?; what is the nature of the "dissolved" state?; why does a solute remain "dissolved"?; what conditions promote this state? Experiments probing these issues are complicated by the difficulty in isolating just one variable for study in a complex mixture of influences. We have found one approach to this problem in earlier

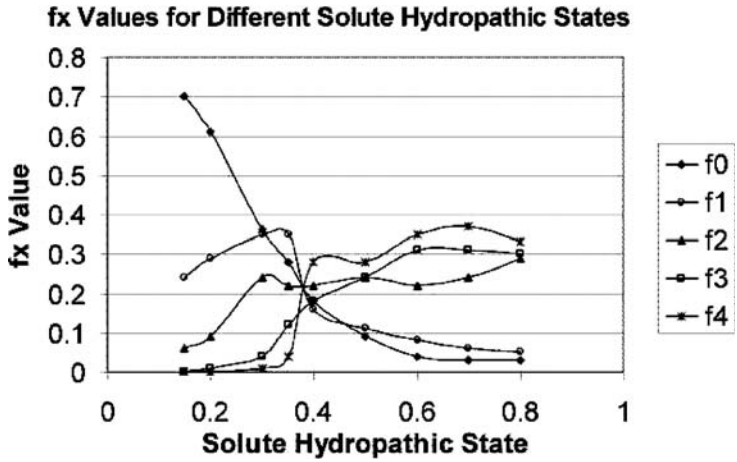


Figure 4.4. A plot of solute structure types as a function of hydropathic state

studies on water and solutions, namely the use of cellular automata to model the system.

Study 4.7. Modeling solute aggregation

In this study we are interested in the attributes of a solute in water as conditions change in the entire system. For example, we want to examine the profile of solute molecules through their f_x values when conditions are varied. As an example, what is the f_x profile of a solute when the hydropathic state of the solute is changed from polar to nonpolar. In Figure 4.4, the hydropathic state is varied using various parameters for P_B(WS) and J(WS). The fractions of solute existing in each of the f_x states is recorded for various parameter sets. Figure 4.4 reveals an interesting pattern. An inquisitive and creative student should explore these ideas.

References

1. G. Johnson, *Fire in the Mind: Science, Faith, and the Search for Order.* Knopf, New York, 1995.
2. L. B. Kier and C.-K. Cheng, A cellular automata model of water. *J. Chem. Inf. Comput. Sci.* 1994, 34, 647–652.
3. L. B. Kier, C.-K. Cheng, B. Testa, and P. A. Carrupt, A cellular automata model of the hydrophobic effect. *Pharm. Res.* 1995, 12, 615–620.
4. L. B. Kier and C.-K. Cheng, A cellular automata model of solution phenomena. *J. Math. Chem.* 1997, 21, 71–77.

5. L. B. Kier and C.-K. Cheng, A cellular automata model of dissolution. *Pharm. Res*. 1995, 12, 1521–1526.
6. L. B. Kier, C.-K. Cheng, and B. Testa, A cellular automata model of diffusion in aqueous systems. *J. Pharm. Sci.* 1997, 86, 774–779.
7. L. B. Kier and T. M. Witten, in *Cellular Automata Models of Complex Systems, in Complexity in Chemistry, Biology and Ecology*, D. Bonchev and D. H. Rouvray, eds. Springer, New York (in press).

Chapter 5

DYNAMIC AQUEOUS SYSTEMS

When water turns to ice does remember one time it was water? When ice turns back into water does it remember it was ice?

—Carl Sandburg

We have introduced the use of cellular automata modeling of water and possibly some other solvent, and have observed the influence of solutes on the emergence of properties in these complex systems. In this chapter we consider a few, more complex chemical systems that may lend themselves to cellular automata modeling. We will discuss several of these and then suggest some studies for the reader.

5.1. Oil–water demixing

Systems composed of two liquids that possess low intersolubility are of considerable interest in chemistry because they mirror some biological phenomena. In particular, a two-phase liquid system possesses some characteristics of membranes and their aqueous surroundings. The two-phase system of nonpolar liquid and water has been used to quantify the property of hydrophobicity, the propensity of a solute molecule to partition in some ratio between two immiscible liquids. This propensity is the attribute used to model the expectation of a drug molecule to cross cell barriers and to enter the brain. Much effort has gone into measuring and predicting this property of a candidate molecule in drug research.

The behavior of two liquids that are sparsely soluble in each other is a familiar one to any experimental chemist. If we shake ether and water in a separatory funnel and observe the system at equilibrium, we see that the liquids have settled in two layers. The reader is asked to identify which liquid is the bottom layer. If water and chloroform are shaken, the system again equilibrates

to a two-phase system. Which layer is the water? The process of separating, or demixing, is a dynamic event that is interesting to watch. Questions arise as to the behavior of the two liquids as they separate, and the nature of the interface. Another interesting question is the behavior of a solute, shaken with the two liquids. Where does the solute end up? What properties of a solute make it more concentrated in the water; or in the ether?

These questions are addressed in a well-designed laboratory experiment. But in the lab we can only see certain things, all associated with the bulk liquids made up of vast numbers of molecules. It would be interesting to attempt to model the demixing and partitioning phenomena at the level of a few molecules. This is the molecular system level that we have described in Chapter 2. Some models have been described using molecular dynamics and Monte Carlo simulations. Kier and Cheng have reported on a cellular automata model of demixing and partitioning of a solute [1,2]. We describe this here and introduce some new procedures and rules.

5.1.1. Experimental design

In order, for the two liquids to separate into two phases, they must be very weakly soluble in each other. When exposed to each other by mixing or shaking in a separatory funnel, they may not interpenetrate each other's realm to any extent. At the molecular level, we infer that the two species of molecules have no significant affinity for each other, rather they are predominantly attracted to other molecules with the same structure. To model this aversion, the joining and breaking rules must encode this behavior. The cells of liquids X and Y must respond to rules typified by those shown in the parameter setup tables below. With these rules we anticipate that liquid X will favor associating with other X molecules, while molecule Y will be found predominantly among other Y molecules.

It is observed that the CA dynamics, with these rules, results in a pattern in which clusters of X are scattered among clusters of Y. This is not what is seen in the laboratory. In reality, two immiscible liquids will separate into two layers, each responding to a different influence from gravity. What the CA model shows is the separation experienced in the weightlessness of space. A gravity rule must be added.

5.1.2. The gravity rule

The gravity rules were described in Chapter 2. They are selected to impart a different orientation among molecules X and Y. If the gravity rule, $G_R(XY)$ favors X over Y, then X molecules will tend to be the upper layer. The relationship of up and down for relative gravity now requires a reference in the grid construction for these gravity rules to be operative. There must be an upper and a lower boundary on the grid. Therefore, the X and Y molecules cannot pass

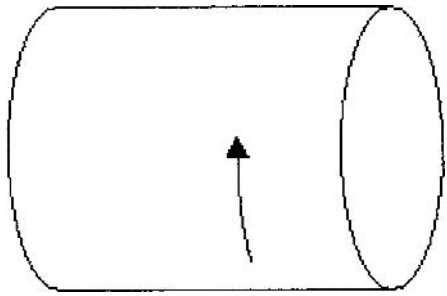


Figure 5.1. A cellular automata grid on the surface of a cylinder. This design permits the use of a parallel set of boundaries, while the other two sides are continuous

through the bottom of the grid as is the case of an unbounded grid on the surface of a torus. In effect we use a grid on the surface of a cylinder (Figure 5.1).

Application 5.1. Immiscible solvent demixing

In this application the reader will examine the influence of the interrelation between the two liquids on the extent and rate of the demixing process. In Example 5.1, the two liquids have a modest affinity for each other, characterized by rules describing a relatively low breaking probability rule between them. These setup data are found in Parameter setup 5.1.

Example 5.1. Demixing of two immiscible liquids

This example describes two liquids, X and Y, with parameters that describe their aversion to each other. This is accomplished with $P_B(XY) = 0.5$ and $J(XY) = 0.7$. The gravity terms are created to impart a preference for one ingredient to move lower in the grid than the other ingredient. In this example setup, the Y is slightly favored to move lower than the X ingredients. Use Example 5.1 in the program CASim to run this example.

Parameter setups for Example 5.1. Demixing

Grid 55 × 55 cylinder with top and bottom boundaries
Liquid X 1050 cells (blue)
Liquid Y 1050 cells (green)
$P_B(XX) = 0.4$, $J(XX) = 1.0$, $P_B(YY) = 0.4$, $J(YY) = 1.4$, $P_B(XY) = 0.5$, $J(XY) = 0.7$
$G_R(XY) = 0.35$, $G_R(YX) = 0.20$
Run the simulation 1000 iterations, and then adjust the run time to observe a well-formed interface.

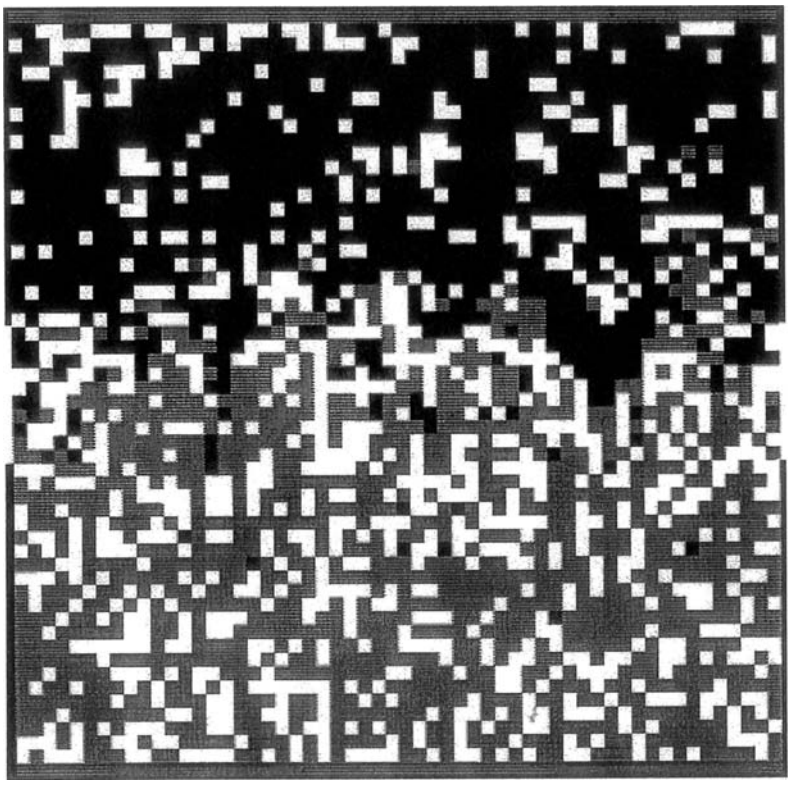


Figure 5.2. A cellular automata model of the interface between two immiscible liquids, after the demixing process has reached an equilibrium

Observe the progress of the demixing. Make a rough estimate of the time required to achieve a clear interface with this set of parameters. The interface seen after a long run is shown in Figure 5.2.

Studies 5.1a and b. Variation of rules in demixing studies

Repeat the study using the same conditions but change the rules relating to the two ingredients. In this study use $P_B(XY) = 0.9$ and $J(XY) = 0.18$. Estimate the number of iterations needed to achieve an interface at a rough equilibrium. Repeat the study using the same conditions but with the rules describing a closer relationship between the ingredients. Use $P_B(XY) = 0.2$ and $J(XY) = 2.0$. Compare the extent of demixing among these three studies, and note what influence the similarities in polarity have on the demixing process. Draw some conclusions about the necessary properties of any two liquids that make them immiscible and that permit demixing when they are homogenized. It is also very interesting to observe the progress of the demixing. This process can be

halted along the way to inspect the structures at the modeled molecular system level.

Application 5.2. Partition coefficients

The modeling of the partitioning of a solute between two immiscible liquids is now possible using the parameter setups described in Example 5.1 and Studies 5.1a,b. The solute may be assigned P_B and J values, reflecting a preference for one or the other solvents. For example, we may introduce a solute into the mixtures described in Example 5.1, by replacing an equal number of X and Y molecules with a third ingredient designated as solute, S. The solute, S, should be given joining and breaking parameters, reflecting its relationship to liquids X and Y.

Example 5.2. Partitioning of a solute between two liquids

In this example, a solute is introduced into the system described in Example 5.1. We introduce 50 molecules of solute, S, by subtracting 25 cells each from liquids X and Y. Joining and breaking parameters are selected for the SX and SY relationships, shown in Parameter Setup 5.2.

Parameter setups for 5.2. (Example) Partitioning

Grid 55 × 55 cylinder with top and bottom boundaries
Liquid X 1025 cells (blue)
Liquid Y 1025 cells (green)
Solute S 50 cells (red) random initial distribution
$P_B(XX) = 0.4$, $J(XX) = 1.0$, $P_B(YY) = 0.4$, $J(YY) = 1.4$, $P_B(XY) = 0.5$, $J(XY) = 0.7$ $P_B(SX) = 0.1$, $J(SX) = 2.8$, $P_B(SY) = 0.7$, $J(SY) = 0.36$, $P_B(SS) = 0.5$, $J(SS) = 0.7$ $G_R(XY) = 0.35$, $G_R(YX) = 0.20$
Run the dynamics for 2000 iterations, then record the number of solute, S, molecules in each phase.

The partition coefficient is the log of this ratio. Figure 5.3 shows an example of the partitioning of a solute between two phases.

Study 5.2. Variations in the hydropathic state of the solute

In the following studies change the parameters of the solute shown in Example 5.2. For example, use $P_B(SX) = 0.7$, $J(SX) = 0.36$, $P_B(SY) = 0.1$, and $J(SY) = 2.8$, replacing the corresponding parameters shown in Example 5.2. Run the dynamics and observe the difference from Example 5.2. Look at the cover of the book. Repeat with a variety of parameters for the solute, S.

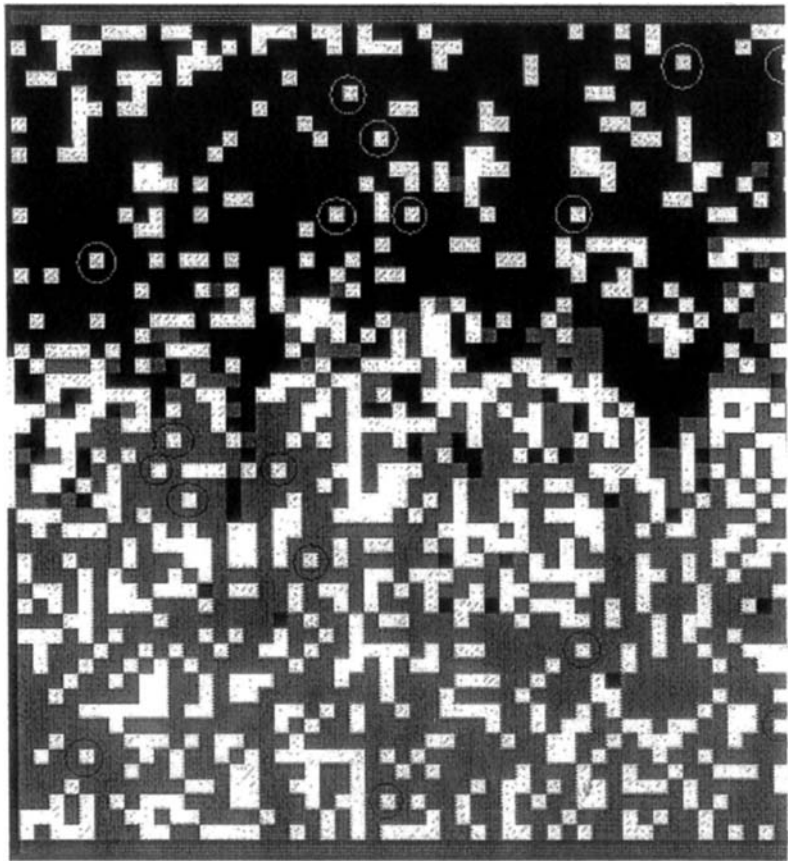


Figure 5.3. A cellular automata model of the interface between two immiscible liquids, after the demixing process has reached an equilibrium. A solute (encircled cells) has partitioned into the two phases according to its partition coefficient

5.1.3. Observations of runs

In each of the studies described above, the reader is encouraged to watch the development of the two phases from the original homogeneous mixture. Note that there is significant clustering of like molecules well before two discrete phases form. Another observation is the ragged nature of the interface. It is not a smooth discrete separation in this molecular system level model.

5.2. Modeling micelle formation

We have seen how solutes behave in terms of dissolution, hydrophobic effects, and diffusion. With cellular automata it is possible to examine the

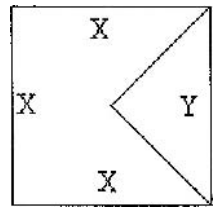


Figure 5.4. An example of variegated cells that may be used in modeling an amphiphile molecule with different trajectory rules for the different types of edges

behavior of certain molecules that have distinct topological features that influence more complex patterns of organization. One such molecule is the amphiphile. These are molecules with two different structural features imparting both hydrophobic and hydrophilic inter-molecular interaction behavior. Amphiphiles are prominent as surfactants, bile salts, and components of cell membranes. Typically they have a large fragment that is quite hydrophobic in nature. This fragment may be a long-chain alkyl group or a phenyl group. The other part of the amphiphile is often a carboxylate or phosphate group with notable affinity for interaction with water.

Application 5.3. Formation of micelles

The amphiphile can be modeled using one of the variegated structures described earlier [3], as shown in Figure 5.4. Each part of the cell modeling the amphiphile is endowed with a separate set of rules imparting the appropriate hydropathic state. Because the hydrophobic fragment, X, is usually much larger, it is appropriate to model it with three of the four cell faces. This is the light portion of the cell in the simulation. The hydrophilic fragment, Y, is modeled by the dark cell face.

Example 5.3. Modeling micelle formation

An example of parameters for these two molecular fragments is: $P_B(WY) = 0.2$, $J(WY) = 2.0$ and $P_B(WX) = 0.80$, $J(WX) = 0.25$, where Y is the hydrophilic fragment and X is the hydrophobic fragment. The other parameters are modeled with $P_B(XX) = 0.2$, $J(XX) = 2.0$, $P_B(YY) = 0.5$, and $J(YY) = 0.70$. The water temperature can be varied to study its influence on the emergent properties. Use a grid of 55×55 with 100 amphiphiles and 2000 water molecules.

Parameter setups for Example 5.3. Micelle model

Grid 40 × 40 cells on a torus
Water, W are 1000 cells
Solute molecules, S, are 100 cells, each with 3 hydrophobic faces, X, and one hydrophilic face, Y
$P_B(WW) = 0.25$, $J(WW) = 1.70$
$P_B(XX) = 0.2$, $J(XX) = 2.0$, $P_B(YY) = 0.5$, $J(YY) = 0.7$, $P_B(XY) = 0.5$, $J(XY) = 0.7$
$P_B(WX) = 0.8$, $J(WX) = 0.25$, $P_B(WY) = 0.20$, $J(WY) = 2.0$
Run the dynamics for 1000 iterations, pausing when an interesting structure appears.

The reader is encouraged to run this model and collect the average cluster size of amphiphile cells. Observing the run reveals a view of the emergent property known as micelle formation. Periodic halting of the run when these micelles are prominent will be of interest. Try a screen grab of several good examples.

Study 5.3. Water temperature effects on micelle formation

Systematic variation in the water temperature, (WW), will produce a profile reflecting this influence. Vary the $P_B(WW)$ and $J(WW)$ values in Example 5.3 to simulate different water temperatures. Run the dynamics for these different water temperatures to observe its influence. Note whether this is a linear or nonlinear effect on the cluster size. The structures formed may be quantified by recording the average micelle cluster size. The typical pattern looks like the examples in Figure 5.5.

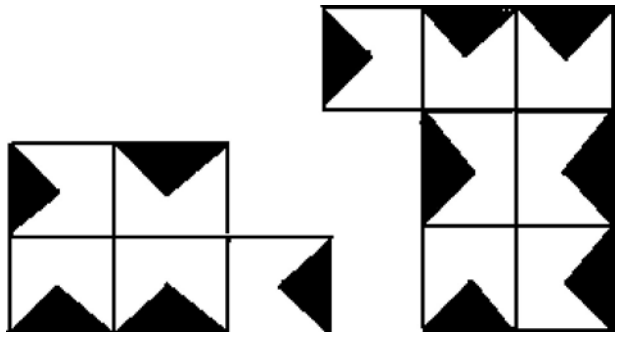


Figure 5.5. Examples of a cellular automata modeling of micelle formation. The dark faces of each cell model the hydrophilic part of the amphiphile, while the light faces model the hydrophobic features of the amphiphile molecule

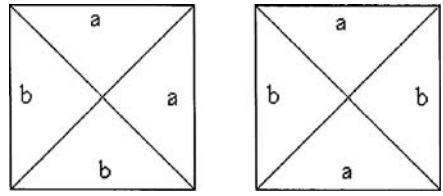


Figure 5.6. Cell types that may be used in modeling crystal formation

5.3. Modeling crystal formation

The micelles just discussed, represent a phenomenon arising from a heterogeneous set of surface properties of a molecule. Part of a molecular surface is hydrophobic and part of it is hydrophilic. In the presence of water the affinities of each part may lead to a pattern of self-adhesion, producing a micelle. The structure of the micelle is such that the system remains in solution. An alternative consequence of the encounters of solute molecules is the formation of aggregates of solutes, large enough to destroy their ability to remain in solution. These structures leave the water environment as precipitates. Sometimes the form that they assume is a highly regular aggregate called a crystal.

Application 5.4. Crystal formation

Crystal formation occurs as a local event, self-organizing into a reoccurring system of molecules that have a uniform arrangement throughout the aggregate. The formation of such a system lends itself to modeling with cellular automata. If we select cells with variegated faces, as in Figure 3.4 in Chapter 3, we would anticipate formation of aggregates of different patterns of organization. As an example, consider the possible variegated cells in Figure 5.6.

If we prescribe rules for one type of cell in Figure 5.6, where the, a, faces form strong bonds with each other, while the, bb, and, ab, junctures are not strong, we anticipate a particular pattern of crystal structure to form.

Example 5.4. Crystal formation with a selected rule

Use the first cell type aabb shown in Figure 5.6 for this study. Use the following parameter setups.

Parameter setup Example 5.4. Formation with aabb cell

Grid 40 × 40 on a torus
Water 900 cells (blue)
Solute 200 variegated cells with face pattern aabb

$P_B(WW) = 0.2$, $J(WW) = 2.0$ $P_B(aa) = 0.1$, $J(aa) = 2.8$, $P_B(bb) = 0.9$, $J(bb) = 0.18$
$P_B(ab) = 0.9$, $J(ab) = 0.18$
$P_B(Wa) = 0.5$, $J(Wa) = 0.7$, $P_B(Wb) = 0.5$, $J(Wb) = 0.7$
Count the number of unions of (a) faces with each other.

Observe the patterns created by this set of rules. Pause the run when a particular pattern develops and save the image.

Study 5.4. Crystal formation with the abab cell type

Repeat the run using abab cells and the parameter sets in Example 5.4. Other sets of rules can be used, varying the parameters reflecting strong and weak probabilities in Example 5.4. The variety of cells in Figure 5.6 can be considered for further studies. Rules of joining and breaking among the different cell types can lead to a rich mixture of aggregates modeling different crystal types.

5.4. Modeling percolation

Percolation is a phenomenon associated with ingredients in a system reaching a critical state of association, so that information may be transmitted from one ingredient to another across or through the system without interruption. This can be demonstrated by considering a system with a grid of spaces shown in Figure 5.7. Some objects under study are randomly distributed throughout this grid (Figure 5.7a). Because of the sparcity of these objects, little or no physical contact is encountered. No information is exchanged within the system. If enough additional objects are randomly added to the system (Figure 5.7b), a finite probability arises such that some of these objects may be associated to form clusters. Some exchange of information occurs within the clusters, but the clusters are isolated, and so the information exchange is confined within each cluster. If enough objects are randomly added to the system, the possibility arises that some clusters may appear united as a single cluster, which spans the entire length or width of the system. This spanning cluster produces a conduit through which an uninterrupted flow of information is possible across the system. This flow of information takes place within a process called percolation (Figure 5.7c). The minimum number of objects randomly placed in the system, necessary to have a finite probability of percolation occurring, is called the percolation threshold or the percolation point.

Percolation is widely observed in chemical systems. It was first recognized as a method to describe how small, branched molecules react to form polymers,

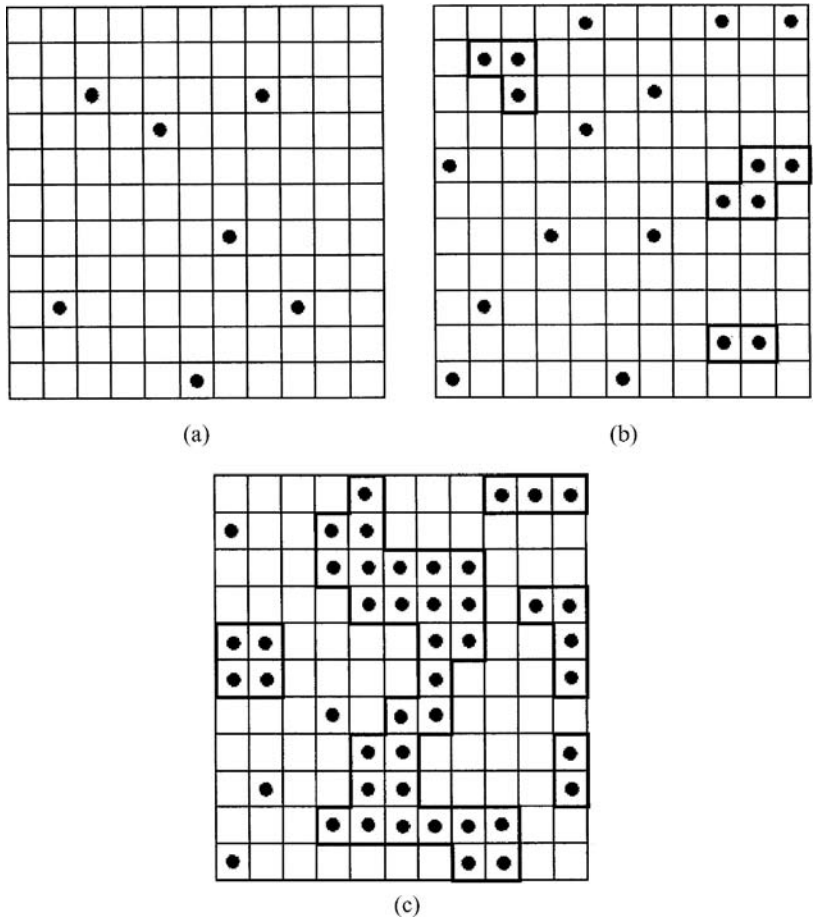


Figure 5.7. (a–c) Stages of grid occupancy leading up to a percolating cluster. In this case there is a continuous path of connected, occupied cells leading from one edge of the grid to the opposite edge

ultimately leading to an extensive network connected by chemical bonds. The relevance of percolation theory to the critical behavior of sols and gels was a major advance in understanding the physics and chemistry of these systems. The structured continuum of water is an example of a percolating system.

Application 5.5. Percolation

The configuration of a system in which percolation may occur is classically treated as one in which the ingredients do not move. Considerable work has been devoted to these static models, leading to numerical solutions of the critical concentrations and cluster sizes associated with a percolation threshold.

These have been worked out mathematically for several grid structures and dimensions. In reality, the ingredients in a system are not static, but they may exhibit dynamic behavior within the system. This may give quite different results for the example in Figure 5.7c, when the percolation threshold is sought. One approach to the study of discrete, dynamic systems is through the use of computer simulations using cellular automata. Kier et al. and Testa et al. have used CA to model a two-dimensional dynamic system of varying concentrations, to study the onset of percolation and large-cluster behavior, both visually and numerically [4,5].

In the following example and studies, the reader will form an estimate of the number of cells (concentration) that produces a 50% probability of forming a static percolating system. Using increasingly larger concentrations, say 100 cell increments on a 55 $\times$ 55 cell grid, compute the average number of cells necessary to produce a 50% probability of a percolation cluster to form. Repeat this with several concentrations of randomly introduced nonmoving cells. It is not necessary to run any dynamics since the initial random placement of cells is a permanent condition. This study becomes a reference against which a dynamic system may be compared.

Example 5.5. Percolation in static systems

Run this example using the following parameter setup. Record the average cluster size and the percent of percolations over a number of runs. The number of iterations will be zero and only the initial grid configuration will be used. A suggested number of runs is 1000.

Parameter setup 5.5. (Example) Percolation in static systems

Grid 55 $\times$ 55 cells on a torus
Cells, A, 300 random in initial configuration of the grid
P_B and J values are all zero
$P_m = 0$
Use only the initial random placement of A cells
Run 100 times and report the average cluster size
Report any possible percolations as a percentage of 100 runs.

Studies 5.5a–c. Parameter variation in static systems

Run these studies with the parameters shown in Example 5.5. Vary the number of A cells using 600, 900, and 1200. Record the average number of percolations over a number of runs. Estimate the number of A cells that produce

a percolating event in 50% of the runs. This is the percolation point for static ingredients.

Studies 5.5d. Percolation among dynamic ingredients

Repeat 5.5a–c, only allow the ingredient cells, A, to move freely, using $P_m = 1.0$, $P_B(AA) = 0.4$, and $J(AA) = 1.0$. At each concentration level, 300, 600, 900, and 1200 A cells, average the number of percolating clusters over some constant number of iterations, say 100. Repeat each concentration study 50 times, compute the percentage of percolation at each concentration, and estimate the concentration producing 50% of the time, a percolating system. Compare this value with the result from a static system, as in Example 5.5.

References

1. C.-K. Cheng and L. B. Kier, A cellular automata model of oil–water partitioning. *J. Chem. Inf. Comput. Sci.* 1995, 1996, 1054-1-59.
2. L. B. Kier and C.-K. Cheng, A cellular automata model of partitioning between liquid phases. In *Lipophilicity in Drug Research*, Pliska, Testa, and Waterbeemd, eds., VCH Publ., 1996.
3. L. B. Kier, C.-K. Cheng, and B. Testa, Cellular automata model of micelle formation. *Pharm. Res*. 1996, 13, 1419.
4. L. B. Kier, C.-K. Cheng, and B. Testa, Dynamic percolation modeled by cellular automata. *J. Chem. Inf. Comp. Sci.* 1999, 39, 326.
5. B. Testa, L. B. Kier, C.-K. Cheng, and J. Mayer, A cellular automata study of constraints (dissolvence) in a percolating many-particle systems. *Entropy*. 2001, 3, 28.

Chapter 6

WATER–SURFACE EFFECTS

I have been carried . . . into the sanctuary of minuteness and of power, where molecules obey the laws of their existence, clash together in fierce collision, or grapple in yet more fierce embrace, building up in secret the forms of visible things.

—James Clerk Maxwell [1]

In this chapter we address several phenomena involving a solvent, principally water, and a stationary surface. These include various wetting and wall effects, chromatography, and membrane passage. Some of these phenomena have been modeled with cellular automata, and a brief description of those studies will be presented. Each of these examples opens up a wealth of possibilities for future work, and the reader is urged to pursue some studies that these may inspire.

6.1. Water–vessel surface encounters

Interest in relationships between water solutions and stationary surfaces is revealed in a variety of research applications, including studies of wetted surfaces, heterogeneous catalysis, fluid flow, flow through blood vessels, colonic pathologies, and ligand–protein encounters. A number of studies have focused on nonpolar solutes and water as models of what might be occurring at macroscopic, nonpolar surfaces. Effects of a surface on water structure may extend more than 10 Å into the liquid. Weak hydrogen bonding between water molecules and strong orientation effects may prevail near the liquid–solid interface, depending on the hydropathic states of these two entities. Water confined at a Janus interface (consisting of opposed hydrophobic and hydrophilic surfaces) produces a dewetting at the hydrophobic surface and an accumulation at the opposing hydrophilic surface. As a consequence of these contrary tendencies, the confined water exhibits a fluctuating, complex behavior.

Studies described in earlier chapters used cellular automata dynamics to model the hydrophobic effect and other solution phenomena such as dissolution, diffusion, micelle formation, and immiscible solvent demixing. In this section we describe several cellular automata models of the influence of the hydropathic state of a surface on water and on solute concentration in an aqueous solution. We first examine the effect of the surface hydropathic state on the accumulation of water near the surface. A second example models the effect of surface hydropathic state on the rate and accumulation of water flowing through a tube. A final example shows the effect of the surface on the concentration of solute molecules within an aqueous solution.

Application 6.1. Solvent–wall relationships

In this application the influence of the wall's hydropathic state on the accumulation of water molecules near the wall is examined. The system used is a grid of $55 \times 55 = 3025$ cells of which 2100 (69%) are occupied by ingredients representing water molecules. The vertical sides of the grid are composed of fixed cells, B, with defined hydropathic states. Each water molecule, W, is allowed to move and interact with other water molecules according to the water–water joining rule $J(\mathrm{WW}) = 1.10$ and the water–water breaking rule, $P_{\mathrm{B}}(\mathrm{WW}) = 0.375$. As determined in earlier studies, these parameters correspond to a water "temperature" of 37.5°C. The parameters governing interactions between the wall cells, B, and the water molecules, W, define the hydropathic state of the wall. Sets of parameters are used, $P_{\mathrm{B}}(\mathrm{WB})$, and $J(\mathrm{WB})$, shown in Table 6.1. Five different sets of these parameters are used, representing a variation from a hydrophilic wall surface (Cases 1 and 2) to a hydrophobic surface (Cases 4 and 5) listed in Table 6.1.

Example 6.1. Effect of wall hydropathic state on water. Case 1 in Table 6.1.

In these studies the wall hydropathic state is varied to reveal the influence on nearby water molecules. In this example the wall is defined as a polar surface.

Table 6.1. Rules encoding the hydropathic states of wall cells[a]

Case no.	Wall state	$P_{\mathrm{B}}(\mathrm{WB})$	$J(\mathrm{WB})$
1	Polar	0.10	2.80
2	Moderately polar	0.30	1.40
3	Neutral	0.50	0.70
4	Moderately nonpolar	0.70	0.36
5	Nonpolar	0.90	0.18

[a] B = wall cells and W = water cells.

The parameter values are shown in the following Parameter Setup. Fifty runs of 1000 iterations each are carried out, and data collected in every 100 iterations, between 500 and 1000 iterations in each run.

Parameter setup Example 6.1. Effect of wall hydrophobic state on water

Grid 55 × 55 cells on a cylinder
Cylinder border, B, 110 cells
Water, W, 2100 cells (blue)
$P_B(WW) = 0.375$, $J(WW) = 1.1$, $P_B(WB) = 0.10$, and $J(WB) = 2.8$
Run 50 times, with 1000 iterations in each run
Record average number of W cells in each of the five columns adjacent to each border of the cylinder
Collect data in every 100 iterations, between 500 and 1000 iterations.

Studies 6.1a–d. Variation of wall hydropathic state

The wall hydropathic state is varied in these studies, ranging from polar, as in Example 6.1, to very nonpolar in the last study in this series. See Table 6.1 for the (WB) values to be used in the studies. Record the average number of water cells, W, in each of the five columns out from each wall.

Discussion

The numbers of water molecules in each column, starting at column 1 adjacent to each wall and proceeding out five columns, are recorded. Some results (averaged) that we have obtained [2] are shown in Table 6.2. The studies with hydrophilic walls, Cases 1 and 2 shown in Table 6.2, reveal a significant tendency for water to accumulate near the wall. The concentrations in Table 6.2 should be compared with the value of 38.2 water molecules, which is the average concentration per column for a uniform distribution of water within this grid. Cases 1 and 2 thus reveal a typical "wetting" of the polar wall by the polar solvent water. In contrast, when the wall is hydrophobic, as in Cases 4 and 5, there are fewer water molecules adhering to the wall. This is a surface manifestation of the hydrophobic effect, commonly referred to as surface dewetting. The results from all the five Cases, illustrated in Figure 6.1, display the forming of familiar surface shapes, called menisci, arising from the behaviors of wetting and nonwetting fluids at a solid surface.

Studies 6.1e and f. Effect of water temperature on wetting properties

Repeat the runs described in Example 6.1, using a lower and a higher water temperature, for example $P_B(WW) = 0.2$, $J(WW) = 2.0$ and $P_B(WW) = 0.7$,

Table 6.2. Water concentrations in columns influenced by wall hydropathic state[a]

	Case no.[b]				
Column	1	2	3	4	5
1	52.8	45.4	30.9	16.3	7.2
2	39.1	38.4	37.7	36.4	36.0
3	37.4	37.9	38.5	39.3	39.7
4	36.9	37.5	38.6	39.5	39.9
5	36.6	37.4	38.5	39.8	40.1
6	36.7	37.4	38.7	39.3	40.2
7	36.8	37.3	38.4	39.3	40.3
8	36.9	37.7	38.5	39.0	40.1
9	37.3	37.7	38.1	39.0	39.8
10	37.0	37.2	38.7	39.4	39.0
27[c]	37.4	37.2	37.8	38.4	38.8

[a] Water concentrations are averaged from the two comparable columns at either side of the grid.
[b] See Table 1 for descriptions of the wall hydropathic state.
[c] The unique, central column.

$J(\mathrm{WW}) = 0.5$. Record the concentrations of water found in columns 1–5 from each surface, as shown in Table 6.2.

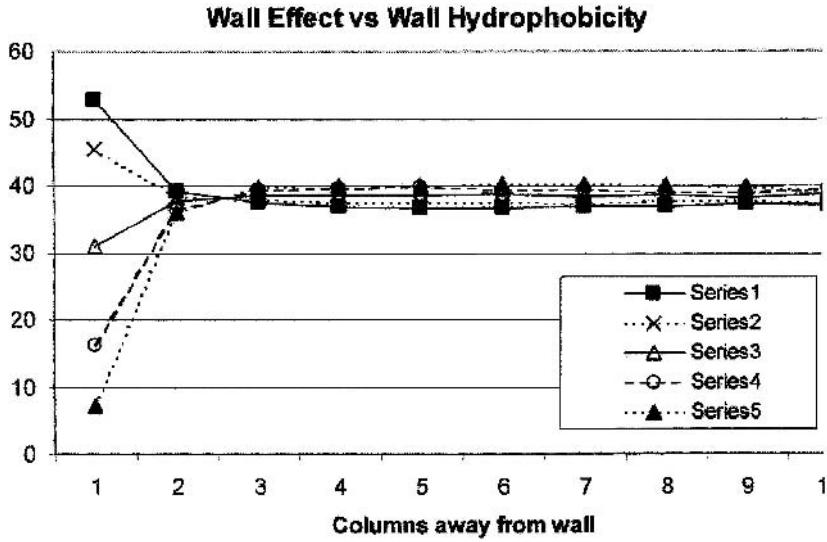


Figure 6.1. Density (number of water molecules) in each column, out from the wall. Each series represents one case varying hydropathic state of the wall

Application 6.2. Effect of wall hydropathic state on water flow

This application is designed to model the influence of the hydropathic state of the wall on the flow rate of water through a tube or other vessel. The system

Table 6.3. Water flow rates influenced by wall hydropathic state (net downward movement of cells in a column per iteration)

	P_B(WB)							
Column no.	0.1	0.3	0.4	0.5	0.6	0.7	0.8	0.9
1	2.0	2.3	4.6	6.2	6.5	6.4	4.9	3.2
2	4.1	4.3	4.5	4.5	4.7	5.3	5.4	4.9
3	4.7	4.7	4.5	4.5	4.7	4.2	4.2	4.1
Middle four columns	4.8	4.4	4.6	4.5	4.2	4.3	4.2	4.0

chosen is a vertical cylinder 55 cells long and 55 cells wide. The vertical cell walls are bounded by fixed rows of stationary cells, B, simulating the wall. The direction of water flow is parallel to the bounded walls of the grid. As in Example 6.1, the water rules are set so as to simulate a temperature of 37.5°C. The flow is modeled using the absolute gravity rule described earlier in Chapter 2. The rate of flow of the water through the cylinder is computed as the difference between the number of water ingredients moving down in a column and the number of water ingredients moving up in the column. The rate is expressed as the average net movement in the 55 cells of each vertical column per iteration. Since one pair of borders in the cylinder is unbounded, molecules moving past the bottom of the grid appear at the top of the cylinder and continue their passage through the system. In addition to the flow rate, the concentration of water in vertical columns is recorded for each set of wall hydropathic state rules, as shown in Table 6.3.

Example 6.2. Influence of water flow rate on wall effect

The flow rate is controlled by the absolute gravity parameter as shown in Parameter Setup 6.2. Run this example to observe the effect of this parameter.

Parameter setup Example 6.2. Influence of flow rate on wetting effect

Grid 55 × 55 cells on a cylinder
Border, B, 110 cells, 55 cells on each side of the cylinder
Water, W, 2100 cells (blue)
$P_B(WW) = 0.2$, $J(WW) = 2.0$, $P_B(WB) = 0.3$, $J(WB) = 1.4$ and $G_a(W) = 0.5$
Record flow rate in each of the five columns next to each border.

Study 6.2a. Influence of wall hydropathic state on the flow rate

Repeat Example 6.2, but change character of the wall to a higher and then to a lower hydropathic state by changing the P_B(WB) and J(WB) values. Report

the effect of these flow rates versus hydropathic states of the wall cells. It would also be of interest to study the effect of water temperature on the flow rate.

Discussion

From these results, and our results shown in Table 6.3, it is seen that the hydropathic state of the wall strongly influences the rate of water flow in the adjacent column (column number 1 in Table 6.3) but less so in neighboring columns toward the interior of the cylinder (column numbers 2 and 3 in Table 6.3). The rate of flow of the column next to the wall increases, as the hydrophobicity of the wall increases up to about $P_B(WB) = 0.6$. At greater values of the wall hydrophobicity, i.e., higher $P_B(WB)$ values, the rate of flow decreases as shown in Figure 6.2. From Table 6.4 it can be seen that the concentration of water next to the wall decreases with increased hydrophobicity. At higher degrees of hydrophobicity, the flow rate in the interior columns decreases slightly; however, the concentrations of water molecules in the interior columns does exhibit a modest increase with increasing hydrophobicity of the wall (Table 6.4). These observations characterized a hydropathic wall effect on the flow rate.

Application 6.3. Effect of hydropathic state of solutes and walls on solutes

This application is designed to analyze the concentrations of solute molecules near the wall when the hydropathic states of both wall and solutes are varied. The simulations are run using a grid of 55 × 55 cells forming a cylinder. The cylinder walls consist of stationary cells, B, that are either polar or nonpolar

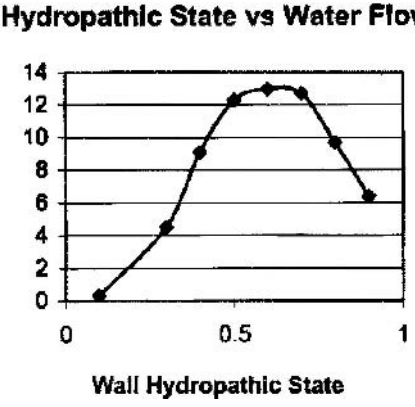


Figure 6.2. Flow rate of water next to a wall, when the hydropathic state of the wall varies

Table 6.4. Water concentration in columns near the walls

Column no.	0.1	0.3	0.4	0.5	0.6	0.7	0.8	0.9
1	193	165	141	114	80	57	40	24
2	140	140	135	139	139	136	136	139
3	135	137	140	140	141	146	146	150
4	129	134	143	141	140	147	147	148
5	133	134	139	140	141	145	149	147
6	133	134	136	140	148	147	147	147
7	134	135	139	143	145	145	146	148
8	136	135	139	145	144	147	145	148
9	138	137	141	145	144	144	147	143
10	140	138	140	142	144	146	150	147
Water concentration change in 2, 3, 4, and 5 relative to the average value of 139	−18	−11	0	+4	+5	+11	+22	+28

in character according to their P_B and J parameters. Four cases are considered in which combinations of the walls and solute molecule hydropathic states are defined as polar or nonpolar. These conditions are summarized in Table 6.5. One hundred and five solute molecules, S, are introduced into the water phase, replacing a like number of water molecules, creating a 5% solute concentration. The solute molecules are given either a polar or a nonpolar state as are the wall cells. These combinations and the results we have found are shown in Table 6.5.

Example 6.3. Wall and solute effects

The water molecules are given a "temperature" of 37.5°C as in the previous examples. The rules for a polar solute relative to water are $P_B(WB)$ and $P_B(WS) = 0.20$; $J(WB)$ and $J(WS) = 2.00$. For a nonpolar solute relative to water, the rules are $P_B(WB)$ and $P_B(WS) = 0.80$ and $J(WB)$ and $J(WS) = 0.25$. The simulation is run and the configurations of each run are recorded after 5000 iterations. The concentrations of both solute and water molecules

Table 6.5. Rules relating wall cells and solute molecules

	Hydropathic state		Solute concentration
Study no.	Solute	Wall	In first column
1	Polar	Polar	1.5 ± 0.1
2	Polar	Nonpolar	0.1 ± 0.03
3	Nonpolar	Polar	0.1 ± 0.02
4	Nonpolar	Nonpolar	18.2 ± 0.4

in columns close to the wall are averaged over the final 200 iterations and recorded.

Parameter setup Example 6.3. Wall and solute effects

Grid 55 × 55 cells, cylinder.
Water, W, 1995 cells (blue).
Solute, S, 105 cells (red).
Border, B, 110 cells.
$P_B(WW) = 0.375$, $J(WW) = 1.1$, $P_B(SS) = 0.5$, $J(SS) = 0.7$
$P_B(WS) = 0.2$, $J(WS) = 2.0$, $P_B(SB) = 0.5$, $J(SB) = 0.7$, $P_B(WB) = 0.2$, $J(WB) = 2.0$

Run the simulation for 1000 iterations. Average, over the last 200 iterations, the count of S cells in each of the five rows next to the border cells.

Studies 6.3a–c. Effect of wall and solute states on solute distribution

The same system is run using a nonpolar solute. The WS parameters are changed to reflect this attribute of the solute using $P_B(WS) = 0.8$ and $J(WS) = 0.25$. Record the number of S cells out of five layers from the wall. Repeat using the other combinations of solute and wall states as shown in Table 6.5.

Discussion

The cellular automata models portray the presence of water with and without a solute, near surfaces of differing hydropathic states. In the first example, the simulation reveals the presence of water molecules in greater than average numbers near a wall composed of relatively hydrophilic cells. This is a manifestation of the "wetting" effect of a polar solvent on a polar wall, resulting in a concave upward meniscus at the wall. In contrast, when the cells composing the wall are relatively hydrophobic, the concentration of water molecules at the wall is less than the average expected in a random distribution, and a nonwetting, concave downward meniscus is formed. This latter result corresponds to a classical "hydrophobic" effect. Using these simulations, it is observed that the effect of the wall on nearby water operates over two or three layers of water molecules, leaving the interior water relatively unaffected, a pattern consistent with experimental and theoretical studies.

The second example simulates the flow of water near the wall surfaces, ranging from very hydrophilic to very hydrophobic in character. These simulations suggest that the water flow rate at the wall first increases as the character of the wall changes to a more hydrophobic state but then decreases at still greater wall

hydrophobicity. This nonlinear behavior is likely a consequence of two competing factors influencing the flow rate. For a hydrophilic wall, adhesion of water to the wall (and its influence on the adjacent columns) retards the flow rate. As the wall becomes less hydrophilic (more hydrophobic) there is a weakening of the affinity of the water for the wall, and the flow rate increases as seen in Figure 6.2. At the same time, there occurs an increase in the hydrophobic effect near the wall, resulting in decreased concentration and increased intermolecular attraction and organization of the water in that region. The lower concentration and increased self-affinity of water near the wall acts to retard the rate of flow, and at still higher wall hydrophobic states, these effects dominate. The overall result of these intersecting effects is a "dome-shaped" variation in the rate of flow with a maximum occurring at an intermediate hydropathic state of the wall, as shown in Figure 6.2.

The third application reveals the effects of wall and solute hydropathic states on the accumulation of solute molecules at the wall. The studies show that hydrophobic solutes accumulate in significant concentration at walls with hydrophobic character.

6.2. A model of chromatography

Chromatography is a process used for the separation of solutes based on differences in their affinity for stationary particles in their midst. It is widely used in purification and analysis, where the identity and concentrations are of interest. The procedure employs a stationary bed of particles which have surfaces covered with a layer of molecules with chosen properties. Solutes in a liquid solvent, called the mobile phase, are allowed to pass through the stationary bed, permitting interactions between the surface molecules and the solutes. If two solutes have characteristics permitting one to be more closely attracted to the particle surface than the other, then the weakly interacting solute will pass through the stationary bed and will eventually separate from the more tightly bound solute. This is a dynamic system, one that lends itself to modeling and study using cellular automata. Kier et al. have reported such a model [3].

6.2.1. Structure of a model

The structure of a chromatography model includes the dimensions of the system, the concentrations of the solutes and the stationary phase, plus the probabilities for interactions between the ingredients. A cellular automata grid of 43×100 cells is used (Figure 6.3). The sides of the long dimension are impenetrable boundaries, while the top and bottom short dimension edges are open. A cell moving below the bottom edge of the grid will reappear at the top

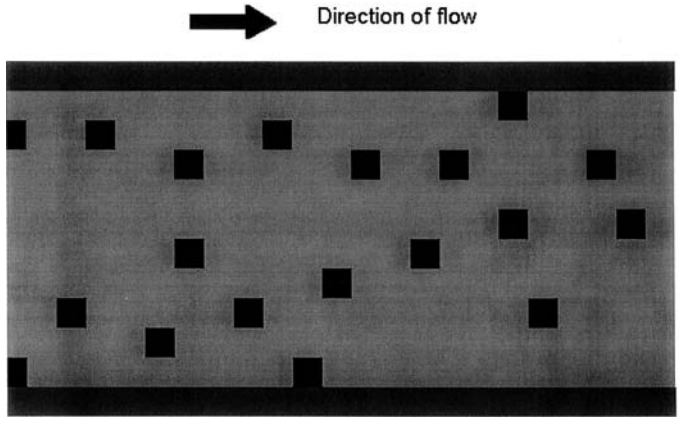


Figure 6.3. A CA grid on a cylinder showing the flow of an ingredient in one direction

of the grid. The grid is thus a cylinder with a continuous surface around the long dimension, as shown in Figure 6.3. The stationary bed is made up of 600 cells randomly dispersed on the grid. No two stationary cells are allowed to be any closer than a three-cell separation. The solvent is modeled by the presence of water cells at a concentration of 69% of the grid space, in the grid. The total grid size is $43 \times 100 = 4300$ cells. If 300 cells of this space are occupied by the stationary cells, then the remaining free space is 4000 cells. The solvent is assumed to occupy about 69% of this free space (as previously assumed in studies using water). Finally, the solutes are represented by a small number of cells with different properties for each cell type. This is the structure of the model as used by Kier et al. to study different properties of a chromatographic separation [3].

6.2.2. The rules

A series of rules describing the breaking, P_B,and joining, J, probabilities must be selected to operate the cellular automata model. The study of Kier was driven by the rules shown in Table 6.6, where S_1 and S_2 are the two solutes, B, the stationary cells, and W, the solvent (water). The boundary cells, E, of the grid are parameterized to be noninteractive with the water and solutes, i.e., $P_B(\text{WE}) = P_B(\text{SE}) = 1.0$ and $J(\text{WE}) = J(\text{SE}) = 0$. The information about the gravity parameters is found in Chapter 2. The characteristics of S_1, S_2, and B relative to each other and to water, W, can be interpreted from the entries in Table 6.6.

6.2.3. The dynamics

The solute cells are placed in a row at the top of the grid. The gravity terms shown in Table 6.6 are designed to simulate a flow of S_1, S_2, and W cells,

Table 6.6. Trajectory parameters

Encounters	P_B(xy)	P_J(xy)	Interpretation
B–B	0.999	0.001	Stationary, at least 3 cells apart
B–S_1	0.90	0.20	Affinity for B
B–S_2	0.10	2.00	Affinity for B
B–W	0.90	0.10	Solvation of B
S_1–S_1	0.90	0.10	Interrelation of S_1
S_1–S_2	0.90	0.10	Interrelation of S_1 and S_2
S_2–S_2	0.90	0.10	Interrelation of S_2
S_1–W	0.90	0.10	Solvation of S_1
S_2–W	0.90	0.10	Solvation of S_2
W–W	0.90	0.10	Relative polarity of solvent
$G(S_1) = 10$			Flow rate of S_1
$G(S_2) = 10$			Flow rate of S_2
$G(W) = 10$			Flow rate of S_2
Column dimensions = 43 × 100 cells			
Water cell content 69% minus 20 solutes and 300 B			
S_1 content = 10 cells			
S_2 content = 10 cells			
B (stationary cells) = 300 cells			
Run "time" = 700 iterations			
Usual number of runs = 10			
System configuration is a cylinder with ingredients flowing back to the top of the system.			

through the long dimension of the column. The cells of the long edges of the grid are stationary boundaries. In other words, the flow created by the gravity terms, allows movement only through the long dimension. Cells reaching the bottom of the 100-cell column pass to the top of the column, as on the surface of the cylinder. Note that the solutes and water always have the same gravity parameters, thus removing any preselected influence on the relative movements of any solute.

Application 6.4. Affinity of a solute for the stationary phase

This application reveals the influence of the parameters on the solute–stationary cell affinity, and the subsequent movement rate of the solute through the column.

Example 6.4. Solute affinity for the stationary phase

In this example, one solute is used on the column. The $P_B(BS_1)$ and $J(BS_1)$ values control the affinity of the S_1 solute for the stationary phase. The parameters are shown in the Parameter Setup 6.4. In the report, the position of each solute cell is recorded in groups of five rows from 0 to 100 after a certain

number of iterations (time). Various times are selected, but the solutes must not be allowed to pass through row 100 to reappear at the top of the column. Each time period chosen is repeated 100 times, and the positions of the solute cells summed for each row. A plot of the number of solutes in each five-row section at various times should be made. It should appear as a Gaussian distribution. Repeat this study using other parameters relating the solute to the stationary cells. Note the position of the distribution maximum in each case. This is a faithful model of an actual chromatographic analysis.

Parameter setup Example 6.4. Affinity of a solute for the stationary phase

Grid 43 × 100 cells on a cylinder. The long dimension is lined with 2 columns, 100 cells, E, on each side
Water, W, 2657 cells (blue)
Stationary phase, B, 300 cells (black) are distributed so that no two cells are closer than 3 cells
Solute, S_1, 10 cells (red) are located in the top two rows of the column
$P_M(W) = 1.0$, $P_M(B) = 0$, $P_M(S_1) = 1.0$
$P_B(WW) = 0.5$, $J(WW) = 0.7$, $P_B(WB) = 0.9$, $J(WB) = 0.1$, $P_B(WS_1) = 0.9$, $J(WS_1) = 0.1$
$P_B(S_1S_1) = 0.9$, $J(S_1S_1) = 0.1$, $P_B(S_1B) = 0.70$, $J(S_1B) = 0.36$
$G_a(W) = 3$
$G_a(S_1) = 3$
Run 700 iterations, 10 runs
Record the number of S_1 cells in each group of five rows.

Studies 6.4a–c. Varying the affinity of the solute for the stationary phase

In these studies, the simulation in Example 6.4 is repeated using a variety of P_B and J values for the B–S_1 encounters. Suggested are P_B/J sets of values for (BS_1) of 0.5/0.71, 0.35/1.19, and 0.2/2.0. Run 700 iterations in 10 runs and record the average numbers of S_1 cells in each group of five rows.

Example 6.5. Relative retention of two solutes

In this example, two solutes are modeled as described for one solute in the previous Example and Studies. Note that there is a separation of the two solutes and that the bands representing concentrations differ in their height and width. This is typical of two different solutes in a chromatography separation. Repeat this study choosing different parameters for the S_1–B and S_2–B encounters.

Parameter setup Example 6.5. Relative retention of two solutes

Grid 43 × 100 cells on a cylinder with the 200 cell dimension fixed cells
Water, W, 2650 cells (blue)
Stationary phase, B, 600 cells (black) are distributed so that no two cells are closer than 3 cells
Solute, S_1, 10 cells (red) are located in the top two rows of the column
Solute, S_2, 10 cells (green) are located in the top two rows of the column
$P_M(W) = 1.0$, $P_M(B) = 0$, $P_M(S_1) = 1.0$
$P_B(WW) = 0.5$, $J(WW) = 0.7$, $P_B(WB) = 0.9$, $J(WB) = 0.1$
$P_B(WS_1) = 0.9$, $J(WS_1) = 0.1$, $P_B(WS_2) = 0.9$, $J(WS_2) = 0.1$
$P_B(S_1S_1) = 0.9$, $J(S_1S_1) = 0.1$, $P_B(S_2S_2) = 0.9$, $J(S_2S_2) = 0.1$, $P_B(S_1S_2) = 0.9$, $J(S_1S_2) = 0.1||$, $P_B(S_1B) = 0.70$, $J(S_1B) = 0.36$, $P_B(S_2B) = 0.40$, $J(S_2B) = 1.0$
$G_a(W) = 3$
$G_a(S_1) = 3$
$G_a(S_2) = 3$
Run 700 iterations, 25 runs
Record the number of S_1 and S_2 cells in each group of five rows.

Studies 6.5a and b. Variation in the affinities of each solute for the stationary phase

In these studies, choose different sets of affinities (S_1B) and (S_2B), and run these with the same parameters for the other ingredient encounters, as in Example 6.5. The cellular automata modeling of chromatographic separation produces a very realistic picture of the events taking place. It provides a visual and a tabular representation of the influence of variables on the process. The student is challenged to pursue these models and to compare them with some of the mathematical descriptions possible from chromatography.

6.3. Modeling membrane permeability

The movement of solutes and water across cell membranes has been of great interest for many years. It is now recognized that most of the ingredients entering and exiting a cell pass through channels or are carried across bound to specialized proteins. Of particular interest is the mechanism whereby water passes in and out of a cell to maintain a state of osmotic pressure compatible with a cell's viability. One way is for the water molecules to pass through channels along with ions and small molecules. It is also recognized that water molecules

pass through the cell membrane by a process of passive diffusion. This latter route relates back to the early consideration of Overton, who proposed a correlation between oil/water partitioning and the propensity to diffuse through a membrane [4]. The intramembrane residency of solutes has been shown to be of considerable importance for certain classes of drugs. In these cases the receptor is within the membrane and access to it is through the membrane. The process of membrane passage is a dynamic one with insight coming from experimental evidence. It is also possible that a well-crafted dynamic model of the ingredients of such a system might produce some understanding of influences on the process.

In this section we describe a cellular automata model of a semipermeable membrane separating two compartments [5]. A solute in one compartment has varied parameters to reflect its relative polarity or lipophilicity. The passage of this solute into and through the membrane is observed, as this property is varied.

6.3.1. Structure of the model

We may construct a cellular automata model of membrane permeability using rules from earlier studies of water and solution phenomena. The model, shown in Figure 6.4, is a grid of 55 × 55 cells on the surface of a cylinder. A band of cells, five rows wide across the middle of the grid is designated as a membrane of which 69% of the space is occupied with a nonpolar ingredient modeling a lipid. These cells may move laterally based on probability-based rules but are constrained to remain within the five membrane-designated rows. Above and below this membrane are compartments of water-designated cells comprising 69% of the space. In the five rows immediately below the membrane there are randomly scattered, a number of cells occupied with solute molecules.

6.3.2. Selection of rules for membrane cell ingredients

In these applications we use rules imparting to the membrane lipid cells, M, a very modest degree of self-affinity, using the parameter set $P_B(MM) = 0.9$ and $J(MM) = 0.1$. The water in the two compartments have identical parameters $P_B(WW) = 0.375$, $J(WW) = 1.7$, but the computer tracking of the water cells in each compartment make it possible to follow the movement of water cells from each compartment, even though they comingle after entering the membrane layer or pass through it to the alternate compartment. Water in each compartment designated W_1 and W_2 for the upper and lower compartments, respectively, is 69% of the cells. The $P_B(WW)$ parameter chosen for each is the same and in correspondence with a temperature of approximately 37°C, as described in Chapter 3. The solute cells, S, may vary in number and parameter values, reflecting different concentrations and degrees of polarity.

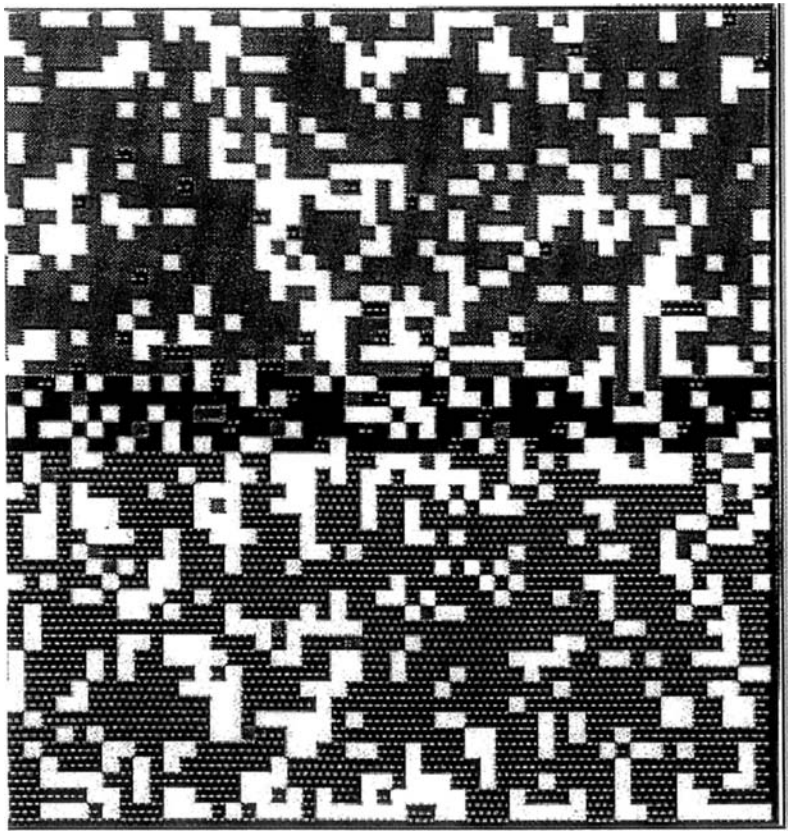


Figure 6.4. Model of a membrane (center horizontal layer of cells) separating two regions of the grid, both being water

A relatively polar solute has a low P_B(WS) value and a high J(WS) value. The rules describing a nonpolar solute have be a high P_B(WS) value and a low J(WS) value. The parameters relating solutes to themselves are held constant at $P_B(SS) = 0.5$ and $J(SS) = 1.0$.

Application 6.5. Effect of solute concentration on water permeation

This application is designed to model the influence of various concentrations of a solute near one edge of the membrane, on the diffusion of water through the membrane. Specifically we are interested in determining whether the model reveals a difference in the flow of water out of one compartment relative to the other. It is well known that if a semipermeable membrane is impervious to a solute on one side of a membrane, a greater flow of water from the other side will occur. This is a model of the osmotic effect, the flow of water through the

semipermeable membrane producing an increase in the concentration on one side. In this model the sizes of the two water compartments are inflexible as to the number of cells they contain. Accordingly we would not anticipate the appearance of a significant increase in the number of water molecules on one side of the membrane. This resistance to volume increase is comparable to the osmotic pressure needed to be exerted to counterbalance the differential flow of water through the membrane.

Using cellular automata we have an opportunity to model the flow of water from each compartment into the membrane, when a solute is present on one side of the membrane. By design, the membrane in our model is composed of 31% empty cells. At iteration zero, in our dynamics, the membrane contains no water. After several iterations, there will be flows of water from the two compartments into the membrane. If we monitor the early stages of this process, we may detect a possible preference for water to flow from one of the compartments. Such a condition would model the early stages of the osmotic effect.

To examine this possibility, we introduce a solute near the lower edge of the membrane. Each solute cell replaces a water-designated cell in that compartment. To create a polar solute, which is unlikely to penetrate the membrane, we give it the set of parameters, $P_B(\mathrm{WS}) = 0.25$, $J(\mathrm{WS}) = 1.7$. This corresponds to a solute being quite polar, hence avid toward water rather than the membrane lipid. We monitored the count of water containing cells entering the membrane from the two compartments over a few iterations at the beginning of the dynamics. Only at this early time is the count valid to determine the source of each water molecule. Later in the dynamics, a water molecule may pass through the membrane and back into it, hence the origin of the water in the membrane may no longer be evident. From these observations we find that a time period between 60 and 150 iterations gives a reliable count of the water movements into the membrane.

The dynamics for each solute concentration study were run 10 times and averaged for each 10 iteration periods. Water molecules from the upper and lower compartments, within the membrane, are designated W_1 and W_2, respectively. It can be seen from these results that the presence of solutes in the lower compartment, W_2, retards the flow of water into the membrane, relative to the flow from the upper compartment, W_1. In addition, this retardation of flow increases as the concentration of solute in W_2 increases. We anticipate that these results are comparable to the situation at the earliest manifestation of the osmotic effect [5].

Example 6.6. Effect of a solute on membrane permeation of water

This example is a model of the influence of a solute, S, near a membrane, on water passing through a membrane. The membrane contains lipid cells

constrained to a five-row section in the middle of the grid. An equal number of water cells lie above and below the simulated membrane. Twenty solute cells, S, lie in a row beneath the membrane.

Parameter setup Example 6.6. Effect of a solute on membrane permeation of water

Grid 55 × 55 on a cylinder
Water, W, 1890 cells, rows 0–25, W_1 945 cells, rows 31–55, W_2 945 cells
Lipid layer, 190 cells, M
Solutes, S, 10 cells (in top three rows of W_2 layer of water)
$P_B(MM) = 0.9$, $J(MM) = 0.1$, $P_B(WW) = 0.37$, $J(WW) = 1.1$
$P_B(MS) = 0.8$, $J(MS) = 0.25$, $P_B(MW_2) = 0.8$, $J(MW_2) = 0.25$,
$P_B(MW_1) = 0.8$, $J(MW_1) = 0.25$, $P_B(SS) = 0.5$, $J(SS) = 0.7$,
$P_B(WS) = 0.25$, $J(WS) = 1.7$
Count average W_1 and W_2 passage into the opposite compartments after 500 iterations in 10 runs.

Studies 6.6a and b. Variation in water temperature on water passage

In the following studies, the water temperature is increased to observe the effect on water passage through the membrane. Change the $P_B(WW)$ and $J(WW)$ values to accomplish these changes. See Chapter 3, Table 3.2 for these values.

Application 6.6. Effect of solute hydropathic state on membrane transport

In this application we examine the influence of solute polarity on its partitioning into, and diffusion through the membrane. The concentration of solute is held constant at [S] = 30 cells in the W_2 compartment. The dynamics are run 10 times for each value of $P_B(WS)$, and the results averaged.

In Figure 6.5, the count of the number of solute cells diffusing from the W_2 compartment, into the membrane, and the number diffusing through the membrane into the W_1 compartment is shown for several values of solute lipophilicity. The permeation of solute through the membrane increases with increasing lipophilicity up to about $P_B(WS) = 0.50$, and then decreases as the lipophilicity increases. The concentration of solute within the membrane is very low, as the $P_B(WS)$ value increases from 0.15 to 0.45. At a value of $P_B(WS) = 0.50$, the count of solute molecules in the membrane exhibits a sharp increase at $P_B(WS) = 0.55$, as shown in Figure 6.5 [5].

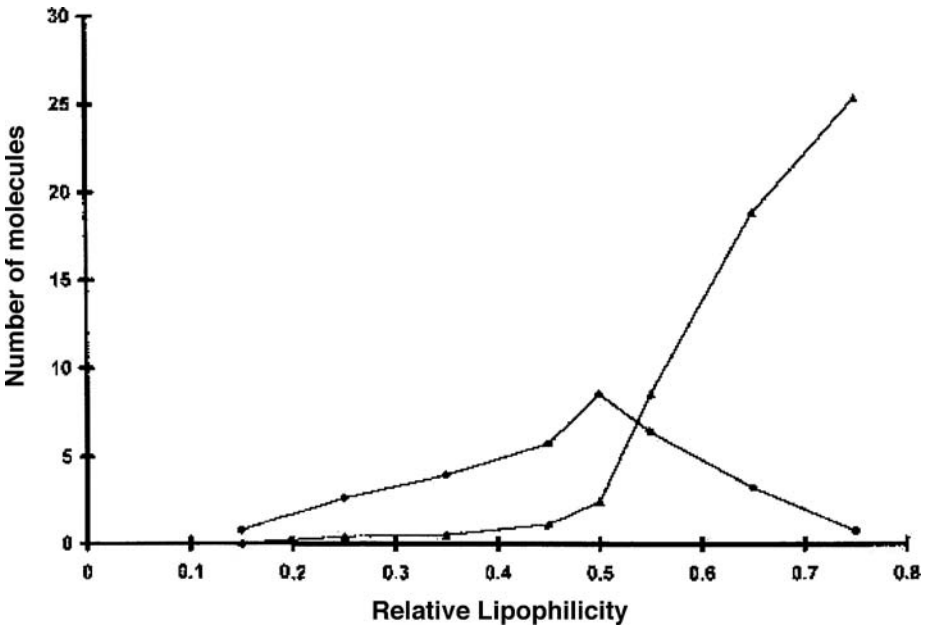


Figure 6.5. The number of solute molecules passing through the membrane, ■, and the number of solute molecules retained in the membrane, ▲, when the hydropathic state of the solute is varied.

Example 6.7. Membrane passage as a function of solute hydropathic state

The hydropathic state of the solutes, S, is varied, and the passage into and through the membrane recorded as a function of this solute property.

Parameter setup Example 6.7. Variation of solute hydropathic state

Grid 55 × 55 on a cylinder
Water, W, 1890 cells, rows 0–25, W_1 945 cells, rows 31–55, W_2 945 cells
Lipid layer, 190 cells, M, in rows 26–30
Solutes, S, 20 cells in the top 3 rows of the W_2 layer of water
$P_B(MM) = 0.9$, $J(MM) = 0.1$, $P_B(WW) = 0.37$, $J(WW) = 1.1$, $P_B(SS) = 0.5$, $J(SS) = 0.7$, $P_B(WS) = 0.20$, $J(WS) = 2.0$
Count W_1 and W_2 passage into the opposite compartments after 100 iterations
Count solute cells, S, in membrane and through it into the W_1 layer.

Studies 6.7a–c. Variation in solute hydropathic state

In this study, the hydropathic state of the solute, S, is varied and the passage of the solute into and through the membrane is recorded for each variation. The other parameters are retained as in Example 6.7. Suggested S hydropathic states are $P_B(W_1S) = P_B(W_2S) = 0.4$, $J(W_1S) = J(W_2S) = 1.0$; $P_B(W_1S) = P_B(W_2S) = 0.6$, $J(W_1S) = J(W_2S) = 0.5$; and $P_B(W_1S) = P_B(W_2S) = 0.8$, $J(W_1S) = J(W_2S) = 0.25$. Count the number of S cells passing into and through the membrane after 100 iterations.

6.3.3. Discussion

In our model of a membrane system, we initially looked for experimental validation of emergent attributes. One example is the difference in water flow into the membrane from the two water compartments, when there is a difference in solute concentration within these compartments. The dynamics reveal that more water flows into the membrane from the less solute concentrated water compartment. These observations are characteristic of the osmotic effect. A second observation from our model is the increase in the passage of the solute through the membrane as the probability rules change, corresponding to a relative increase in lipophilicity. This increase in the passage through the membrane experiences a maximum value at about the midpoint of the relative lipophilicity scale, followed by a sharp decline in the amount of permeated solute. Lieb and Stein (1986) have analyzed transverse membrane permeability data and have concluded that diffusion is dependent upon the volume of a nonelectrolyte [6]. This attribute is factored out in our model, since the solute-designated cells used in the dynamics are of constant size. This is interpreted as reflecting an approximately equal volume among the molecules in the lipophilicity series modeled. The relative lipophilicity values portrayed in Figure 6.5 are thus a consequence of structural differences other than volume. The review of Gupta cites examples of molecular series in which structural attributes other than the volume are the determinant in producing a parabolic relationship between central nervous system penetration and lipophilicity [7].

These models mirror reality to the extent that we have some confidence in the possible significance of other, less documented attributes. One observation from these models that is unexpected is the very abrupt increase in the slope of the line representing the concentration of the solute within the membrane as a function of the rule value $P_B(WS)$, portraying the lipophilicity. This large change in the slope corresponds closely to the $P_B(WS)$ values at which the permeation of the solute through the membrane is maximal (Figure 6.5). To our knowledge, this observation has not been reported. The direct measurement

of solute concentration in a diffusion-functioning membrane would appear to be a daunting task.

The appearance of the curve in Figure 6.5 suggests that a confluence of events is taking place, as the solute molecule is systematically changed from a hydrophilic to a hydrophobic ingredient. The relationship is bilinear, hence we might infer that the aqueous solubility is decreasing while the affinity for the membrane substance is increasing. The superpositioning of these two effects might account for the observed occupancy of the membrane with a small change in the parameters. This pattern of behavior parallels the extent of permeation of the solute through the membrane, observed to have a significant increase in concentration when the solute lipophilicity is at the same midrange value. The suddenness of the change in slope of this concentration increase is unexpected and awaits further detailed interpretation. This sensitivity of the response to relative lipophilicity is a possible subject for further exploration.

We can draw another inference from these models in regard to the flow of water through the membrane. When the concentration of the solute in the membrane increases abruptly with a small change in the lipophilicity, it is likely that the membrane would approach saturation, that is, the cavities among the membrane cells would be extensively occupied. Trauble has proposed that water and small solutes are carried across a membrane by occupying discontinuities or

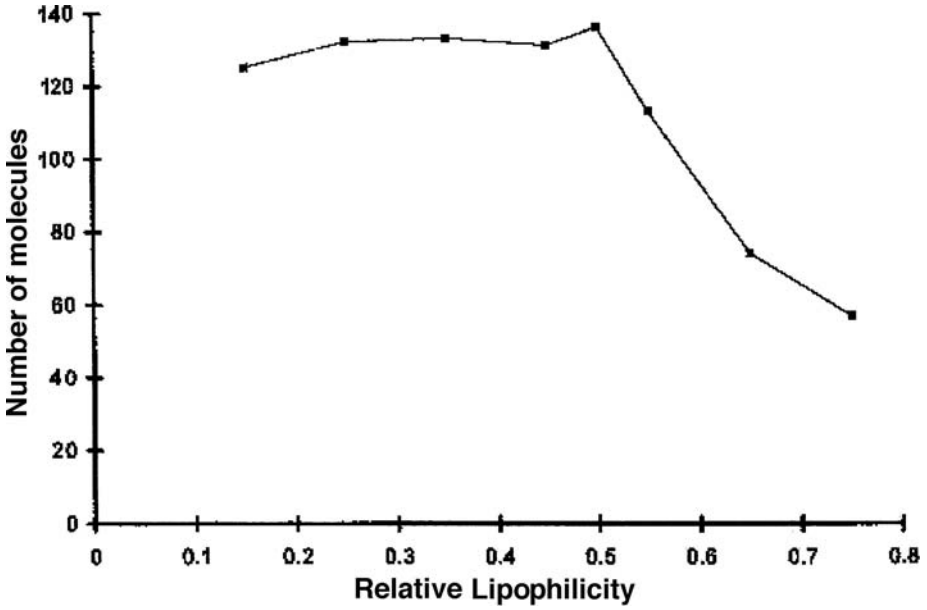


Figure 6.6. The number of water molecules passing through the membrane when the hydropathic state of the solute is varied

kinks in the membrane lipids [8]. We might regard the cavities in our model of the membrane as being surrogates for these kinks, wherein water and solute cells are transported into and through the membrane. If the membrane cavities become mostly occupied at some point in the lipophilicity scale, we would expect that there would be an abrupt decline in the amount of water transported across the membrane. Our modeling of water flux at various degrees of lipophilicity agrees with this expectation (Figure 6.6).

References

1. J. C. Maxwell, *Presidential Address to the British Association for the Advancement of Science*. Liverpool, 1870.
2. L. B. Kier, *Surface Effects Modeled with Cellular Automata*. unpublished.
3. L. B. Kier, C.-K. Cheng, and H. T. Karnes, A cellular automata model of chromatography. *Biomed. Chromatogr.* 2000, 14, 1–7.
4. E. Overton, Uber die allgemeinen osmotischen eigenschaften der zelle. *Naturforsch. Ges. Zurich*. 1899, 44, 87–136.
5. L. B. Kier and C.-K. Cheng, A cellular automata model of membrane permeability. *J. Theor. Biol.* 1997, 186, 75.
6. W. R. Lieb and W. D. Stein, Non-stochesian nature of the transverse diffusion within human red cell membranes. *J. Membr. Biol*. 1986, 92, 111–110.
7. S. P. Gupta, QSAR studies on drugs acting at the central nervous system. *Chem. Rev*. 1989, 89, 1765–1800.
8. H. Trauble, The movement of molecules across lipid membranes: a molecular theory. *J. Membr. Biol*. 1971, 4, 193–208.

Chapter 7

FIRST-ORDER CHEMICAL KINETICS

The grand aim of all science is to cover the greatest number of empirical facts by logical deduction from the smallest number of hypotheses or axioms
–Albert Einstein [1]

The simplest form of a physicochemical reaction takes place when one species simply changes to another. This can be written in a general way as $A \rightarrow B$. The rate of such a reaction is defined as the amount of reactant (the reacting species, A, in this case) or equivalently the product (B) that changes per unit time. The key feature here is the form of the rate law, i.e., the expression for the dependence of the reaction rate on the concentrations of the reactants. For a first-order reaction

$$\text{Rate} = k[\text{A}],$$

where [A] symbolizes the concentration of reactant species A, and k is the reaction rate constant, a proportionality constant that shows how likely the reaction is to occur in a given time period. The rate of a first-order reaction depends directly on how much reactant is present.

There are many familiar examples of first-order processes. The classic example is that of nuclear decay in which one nuclear species decays to another. (The reaction rates for nuclear decays vary enormously: the time frames involved range from tiny fractions of a second to billions of years.) Another example would be the rates of population growth for cities: large cities tend to have greater rates of population growth than small cities since they have higher numbers of people to begin with. In biology, microbial populations increase in direct proportion to the number of microbes present, so long as there is adequate food and wastes are suitably disposed of.

Some chemical reactions also obey first-order kinetics. Isomerization and racemization reactions are normally first-order. Note that whereas nuclear

decays are irreversible, first-order chemical processes, such as isomerizations and racemizations, are generally reversible. That is, in addition to the forward reaction A → B, the reverse reaction B → A can also occur for these processes. Thus if one starts with one pure isomer of a substance, this isomer can undergo first-order transitions to other forms, and in turn these other forms can undergo transitions among themselves, and eventually an equilibrium mixture of different isomers will be generated. The transitions between atomic and molecular excited states and their ground states are also mostly first-order processes. This holds both for radiative decays, such as fluorescence and phosphorescence, and for nonradiative processes, such as internal conversions and intersystem crossings. We shall look at an example of this later in Chapter 9.

Although first-order transitions themselves are inherently straightforward, combinations of first-order transitions can sometimes lead to quite complicated patterns of behavior. We shall examine some of the more basic forms in this chapter [2].

Cellular automata models of first-order kinetic phenomena are relatively simple, since the motions of the ingredients, their encounters with other ingredients, and effects of neighbors do not usually come into play. The grid size employed can range from a single cell to more than 100,000 cells. Only rules governing the transformations between the species need be specified, and each ingredient behaves as an independent actor. It can be instructive to follow the behavior of a single ingredient, but only in the combined behaviors of all of the ingredients do the key patterns called "emergent properties" arise. Normally, it is convenient to fill the grid with the desired starting species, although filling all of the grid cells with ingredients is not at all necessary. Moreover, it will be clear from the examples that follow that as the number of active ingredients increases, the results tend to approach the results expected on the basis of the continuous, deterministic equations.

7.1. Exponential decay

The important phenomenon of exponential decay is the prototype first-order reaction and provides an informative introduction to first-order kinetic principles. Consider an important example from nuclear physics: the decay of the radioactive isotope of carbon, carbon-14 (or ^{14}C). This form of carbon is unstable and decays over time to form nitrogen-14 (^{14}N) plus an electron (e^-); the reaction can be written as

$$^{14}C \rightarrow {}^{14}N + e^-.$$

The *rate* of this reaction, i.e., the amount of ^{14}C disappearing per unit time, depends on the amount of ^{14}C present—the more ^{14}C in a sample, the greater

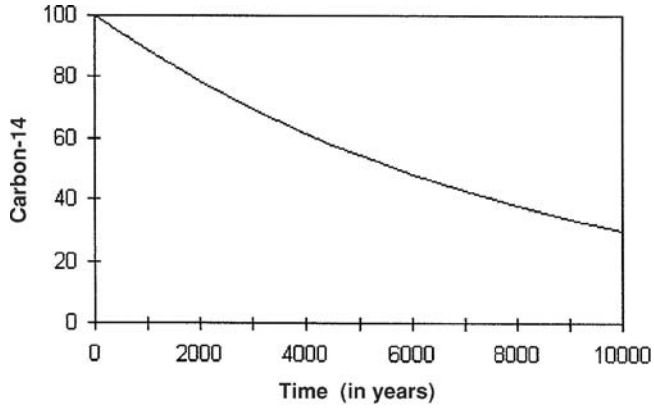


Figure 7.1. Exponential decay

the number of ^{14}C nuclei that decay over a given short time period. Of course, as the ^{14}C disappears, the rate of the reaction slows down. At any point in time, the reaction rate can be expressed as

$$\text{Rate} = k[^{14}\text{C}] \tag{7.1}$$

or in calculus terms,

$$-\frac{\text{d}}{\text{d}t}[^{14}\text{C}] = k[^{14}C] \tag{7.2}$$

In these expressions $[^{14}C]$ represents the concentration of ^{14}C and the "k" is the *rate constant*. From experiments, the rate constant for this process is found to be, $k = 1.21 \times 10^{-4}$ year^{-1}.

We can solve the above differential equation, Eq. (7.2), to determine the concentration $[^{14}C]$ of ^{14}C as a function of time. This solution gives

$$[^{14}\text{C}] = [^{14}\text{C}]_0 \text{e}^{-kt} \tag{7.3}$$

where $[^{14}C]_0$ is the concentration of ^{14}C at time $t = 0$. This is thus an example of the familiar phenomenon called "exponential decay," and if we plot the concentration of ^{14}C against time, where $[^{14}C]_0$ is 100, we obtain the curve shown in Figure 7.1. Taking natural logarithms of both sides of Eq. (7.3) yields

$$\ln[^{14}\text{C}] = \ln[^{14}\text{C}]_0 - kt \tag{7.4}$$

In this case we can plot $\ln[^{14}C]$ against time t to give a straight line as shown in Figure 7.2. (The advantage of such a plot is that it is easier to identify a straight line than to identify other types of curves.) The slope of the line is $-k$. The so-called "half-life" for this process, $t_{1/2}$, is the time it takes for the concentration

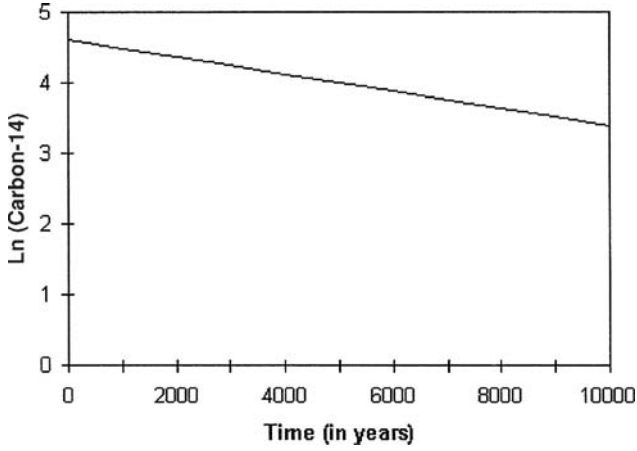


Figure 7.2. A logarithmic plot of the exponential decay of Figure 7.1 showing the linear dependence of ln(concentration) versus time

of the ^{14}C to fall to half its original value. For first-order reactions, the half-life is related to the decay rate constant k by the formula

$$t_{1/2} = 0.693/k \tag{7.5}$$

In the case of radiocarbon ^{14}C, $t_{1/2}$ is about 5730 years.

Exponential decay is quite regular: starting with a given amount of a substance at $t = 0$, this amount will fall to $1/2$ its original value after one half-life, to $1/4$ after two half-lives, $1/8$ after three half-lives, and so forth. This regularity has its usefulness, and the decay of ^{14}C has been widely employed to date archeological artifacts [3].

Application 7.1. Exponential decay

To illustrate these ideas using a CA, let us start with a single ingredient, e.g., a blue ingredient, on a 1×1 grid. We set the transition probability per iteration for the blue cell to turn into a green cell at $P_{\mathrm{T}}(\mathrm{B} \rightarrow \mathrm{G}) = 0.1$ iteration^{-1}, meaning that at each iteration there is one chance in 10 that the blue $\rightarrow$ green transition will occur. Setting the number of iterations to 50 assures that the transition is likely to take place during the time observed. Then, we start the run using the setup for Example 7.1 in the program *CASim*. Since this is a probabilistic model based on a random number generator, the actual transition might occur at any time, perhaps during the first iteration, or possibly during the 9th iteration, or even during the 32nd iteration. Whatever the case, it is clear

that the transition occurs at a very specific time in each trial run: there is no exponential decay at this stage.

Parameter setup for Example 7.1. Exponential decay

Starting configuration: 1 blue ingredient on a 1×1 grid Parameter: $P_T(A \rightarrow B) = 0.01$ Run time = 50 iterations Record the iteration at which the transition occurs Repeat nine more times and record the results.

Do you see any pattern in the 10 results? Determine the median (the "middle" one of the 10) and the average (the total of the times divided by 10) values for the decay times.

Study 7.1a. Exponential decay with 100 ingredients

We then examine a series of 100 runs, or equivalently and more conveniently, one run starting with 100 blue cells using a 10×10 grid. Import the results to a plotting program (e.g., EXCEL©) and plot the number of As remaining versus the number of iterations. Does the pattern look like exponential decay?

Study 7.1b. Exponential decay with 900 ingredients

Now expand the experiment to 900 ingredients on a 30×30 grid. Again, plot the number of A cells versus the number of iterations, n. In this case also plot ln[A] versus n omitting the late part of the decay where there are too few ingredients remaining for meaningful analysis. Obtain the slope (trendline) of this plot with its statistics. Compare the negative of the slope of your plot with the transition probability $P_T(A \rightarrow B)$.

In these examples, we go through a classic example of an emergent property. The phenomenon of exponential decay arises when a very simple rule, a specific transition probability, is imposed on the members of a population of ingredients. But the pattern emerges only from the collective behavior of a significant number of ingredients. The pattern predicted on the basis of a deterministic, differential equation analysis (the kind found in most textbooks) appears as a limiting case of our discrete model, namely as the limit when a very great number of ingredients are involved. This is, of course, what occurs in nature, where the patterns observed from common discrete systems such as collections of atoms or molecules appear to be deterministic and continuous because the observations normally involve huge numbers of actors, often approaching Avogadro's number.

An interesting point to consider here concerns the comparison of the classical deterministic analysis and our discrete analysis. The deterministic

calculus-based rate equation approach assumes that A and t are continuous functions, so that the decay process is represented as in Eq. (7.3) by a smooth exponential decay. The rate constant k represents the probability per unit time for the transition A → B, and appears as the slope $-k$ in a plot of ln[A] against time (Figure 7.2). The corresponding deterministic half-life is $t_{1/2} = \ln 2/(k)$. One might expect that our transition probability P_T should play the same role in the CA model as the rate coefficient k does in the deterministic approach. There is, however, a subtle difference. For the probabilistic, discrete decay that takes place in our CA model (and in nature), the population of As after n iterations ($[A]_n$) is given by

$$[A]_n = [A]_0(1 - P_T)^n \tag{7.6}$$

where P_T is the transition probability per discrete iteration. This analysis yields a "slope" $d[A]/dt = \ln(1 - P_T)$ and a half-life (in iterations) of

$$n_{1/2} = -\ln 2/\ln(1 - P_T) \tag{7.7}$$

The deterministic and discrete expressions for the slopes, and hence the half-lives, are thus not identical, but since mathematically when x is much smaller than unity (i.e., $x \ll 1$), $\ln(1 - x) \approx -x$, in those cases where the transition probability $x = P_T$ is very small compared to 1, the continuous (deterministic) and discrete expressions yield essentially the same values. When P_T is not small compared to 1, the continuous and discrete solutions will differ. For the case $P_T = 0.10$, the half-life solutions are $t_{1/2} = 6.93$ and $n_{1/2} = 6.58$ iteration, a noticeable difference of 5%. For $P_T = 0.01$, $t_{1/2} = 69.31$ and $n_{1/2} = 68.97$ iterations, there exists a difference of just 0.5%. For still smaller values of P_T, the difference between the two expressions continues to decrease.

Study 7.1c. Changing the exponential decay rate

Repeat the study of exponential decay described in Study 7.1b, but now using $P_T(A \rightarrow B) = 0.01$. First use 900 cells on a 30 × 30 grid, as in Study 7.1b. Then use a 50 × 50 = 2500 cell grid. For this second case, plot ln[A] versus time in iterations and obtain the slope of the decay line**.** (Use just the first 80% of the plot to estimate the slope. The final part of the plot has too few active cells to be well behaved.) Compare the slope obtained from your plot with the slopes expected from (1) the deterministic approach with $k = 0.01$ and (2) the discrete cellular automata approach with $P_T = 0.01$.

7.2. First-order equilibrium

In the preceding exercise the process A → B was irreversible. In many practical cases, however, the reverse reaction B → A can also occur. This sets up a dynamic situation in which A is transforming to B and at the same time

B is transforming back to A. The result, after a time, is a dynamic equilibrium condition in which the total numbers of A and B remain fairly constant even though the individual ingredients are constantly switching from one form to the other. The equilibrium constant K_{eq} for this situation is defined as

$$K_{eq} = [B]/[A] \tag{7.8}$$

Within the deterministic approach, this can also be expressed in terms of the forward and reverse rate constants, k_{for} and k_{rev}, and equivalently in our discrete CA model the forward and reverse transition probabilities, $P_T(A \rightarrow B)$ and $P_T(B \rightarrow A)$, respectively

$$K_{eq} = k_{for}/k_{rev} = P_T(A \rightarrow B)/P_T(B \rightarrow A) \tag{7.9}$$

Application 7.2. First-order equilibrium

We can test this notion by setting up a 10×10 grid and filling it with 100 blue (A) cells. We set $P_T(A \rightarrow B) = 0.10$, $P_T(B \rightarrow A) = 0.05$ and the number of iterations to 1000, as shown in Example 7.2. Upon starting the run, the ever-changing individual blue and green (B) ingredients will quickly tend toward a situation in which roughly two thirds of the ingredients are green and about a third are blue. This is illustrated in Figure 7.3. In fact, the same equilibrium condition will be reached regardless of the starting numbers of blue and green cells. Moreover,

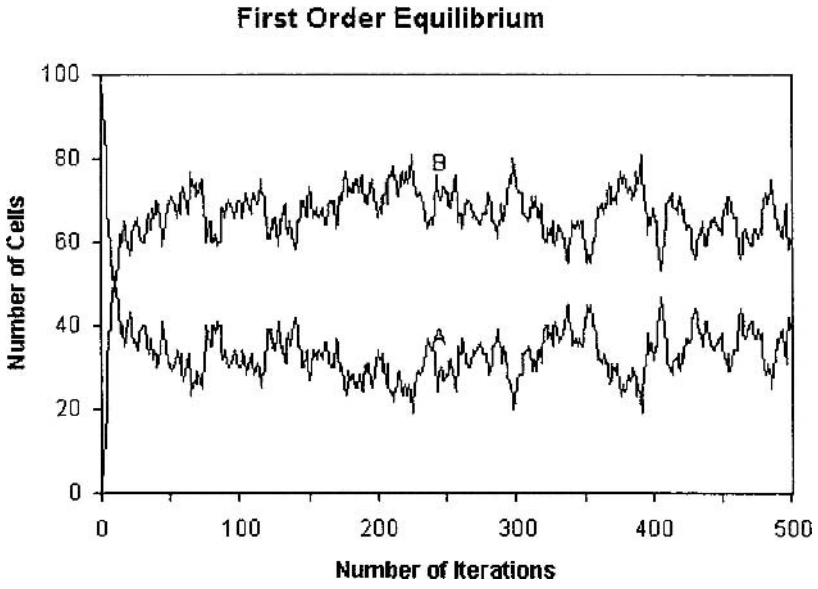


Figure 7.3. A plot showing the transition to an equilibrium condition between A and B ingredients in a 100-cell system as described in Example 7.2. Note the fluctuations in the numbers of A and B cells illustrating the dynamic nature of the equilibrium

as in a real molecular system, even after the equilibrium has been reached, the concentrations of A and B will fluctuate about their average values. In this example, equilibrium is obtained rather rapidly because of the high transition probabilities assumed. Using data from the last 500 iterations of the run, a period well after equilibrium has been reached, we can determine the average equilibrium values for [A] and [B] along with their standard deviations, which provide measures of the fluctuations. From these values we can determine the equilibrium constant K_{eq} and its uncertainty for our experiment.

Parameter setup for Example 7.2. First-order equilibrium

Starting configuration: 100 A cells on a 10 × 10 grid Parameters: $P_T(A \rightarrow B) = 0.1$, $P_T(B \rightarrow A) = 0.05$ Run for 1000 iterations Plot the numbers of A and B cells versus iterations Determine the average values of [A] and [B] and their standard deviations using the last 500 iterations of the run Determine K_{eq} from these average values Compare the K_{eq} with the expected deterministic value.

Study 7.2a. Effect of a larger number of cells on the equilibrium condition

If we increase the number of ingredients in our study, fluctuations will still occur, but because of the greater number of ingredients the fluctuations will tend to become smaller relative to the total ingredient concentrations, so that the population curves we obtain should look smoother. Of course, this same feature prevails in real systems, where the fluctuations become unnoticeable at the very large concentrations normally involved in typical laboratory experiments. In this study keep the same transition probabilities as in the example, but now place 2500 starting A ingredients on a 50 × 50 grid. Again plot the numbers of A and B ingredients as they vary over 2000 iterations. Compare these results with those obtained in Example 7.2. A typical result is plotted in Figure 7.4.

Study 7.2b. Equilibrium from a different starting condition

Repeat the 100-cell equilibrium setup of Example 7.2, but starting with all cells in the B form. What specific result do you obtain for K_{eq} at 2000 iterations? Then, analyze the values for the interval from 1000 to 2000 iterations statistically using EXCEL™ or a similar program to obtain average values for the concentrations [A] and [B] and also for K_{eq} over this period, along with their standard deviations.

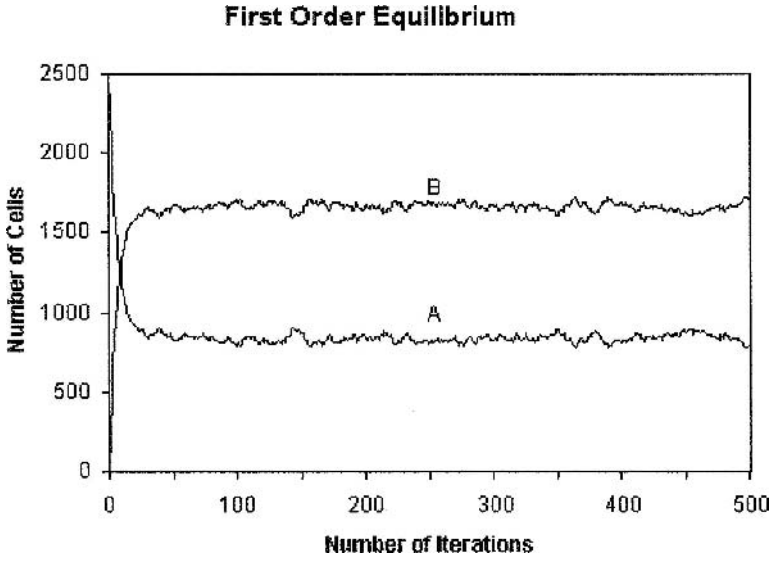


Figure 7.4. The transition to equilibrium for an A ↔ B system with 2500 ingredients showing the effect of an increased number of ingredients

Study 7.2c. Equilibrium with different probabilities

Perform the same analyses as in Study 7.2b, but now using $P_T(A \rightarrow B) = 0.02$ and $P_T(B \rightarrow A) = 0.008$. Compare the results with those obtained in Study 7.2b. Determine the average values of [A] and [B] after equilibrium is reached and the corresponding equilibrium constant K_{eq} with its standard deviation. Compare the measured K_{eq} with the expected limiting deterministic value.

7.3. Series reactions

Some reactions, including both chemical reactions and nuclear decays, occur in a series of steps. We can look at the sequence of reactions $A \rightarrow B \rightarrow C$ and study the changes in the numbers of the three species as the reaction series proceeds.

Example 7.3. Series reactions

In this example, we shall use a $50 \times 50 = 2500$ cell grid originally with all the cells in form A and set $P_T(A \rightarrow B) = 0.05$ iteration^{-1} and $P_T(B \rightarrow C) = 0.01$ iteration^{-1}. The reaction sequence should be monitored for 1000 iterations. The resulting concentration changes are shown in Figure 7.5. As shown in the figure, the number of A cells falls off exponentially, while the number of B cells increases, reaches a maximum, and then falls, as the run continues. The C cells

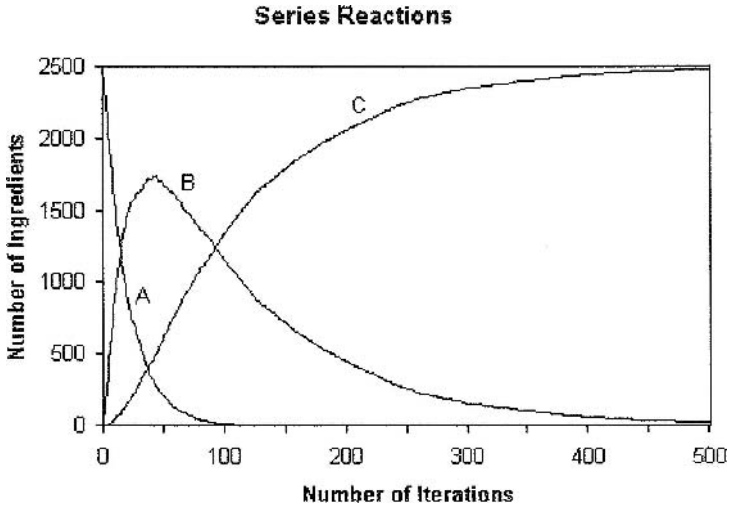


Figure 7.5. Concentrations of A, B, and C ingredients versus time for the reaction sequence $A \rightarrow B \rightarrow C$ as described in Example 7.3

build up in numbers until eventually all of the cells are in the C form. The specific forms of the concentration curves will depend on the relative values of the transition probabilities used for the steps $A \rightarrow B$ and $B \rightarrow C$. This example illustrates the buildup of an intermediate species in the course of a reaction sequence.

Parameter setup for Example 7.3. Series reactions

Starting configuration: 2500 A cells on a 50×50 grid
Parameters: $P_T(A \rightarrow B) = 0.05$ iteration^{-1} and $P_T(B \rightarrow C) = 0.01$ iteration^{-1}
Run for 1000 iterations
Plot numbers of A, B, and C cells versus iterations
Determine the maximum concentration of the B cells ($[B]_{max}$) for this example and note the iteration n_{Bmax} at which $[B]_{max}$ occurs.

Study 7.3a. Sequential reactions

Repeat the study of Example 7.3 for the sequence of reactions $A \rightarrow B \rightarrow C$ but now with the transition probabilities reversed, i.e., setting $P_T(A \rightarrow B) = 0.01$ iteration^{-1} and $P_T(B \rightarrow C) = 0.05$ iteration^{-1}. Plot the results in terms of concentrations versus time. How do the results differ from those in Example 7.3?

Study 7.3b. Rate-limiting step

Examine the reaction sequence $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$ involving five species and four reaction steps. Use a 2500-cell grid with runs of 1000 iterations each.

Set the transition probabilities to 0.02 for all the steps except the step B → C. In separate runs set P_T (B → C) to 0.002, 0.004, and 0.006 iteration^{-1}. Plot [A], [B], and also [E] for all the runs on a single plot. The slow step B → C is an example of a "rate-limiting step" for this reaction series. It is the slowest step and acts as a sort of bottleneck in the series. Note the effect of P_T (B → C) on the rate of production of the final product [E] in this example.

7.4. Parallel reactions

In addition to sequential reactions, reactions can occur in parallel and compete with one another. For example, a single reactant A might form two different products, B and C:

$$\begin{array}{l} A \rightarrow B \\ \downarrow \\ C \end{array}$$

Whether product B or product C will predominate over the course of the reaction depends on the relative values of the transition probabilities, P_T (A → B) and P_T (A → C), connecting A to these products.

Example 7.4. Competing parallel reactions

In this example, we shall again use a 50 × 50 grid with all 2500 cells starting in the A form. We set P_T (A → B) = 0.04 iteration^{-1} and P_T (A → C) = 0.01 iteration^{-1} and follow the process for 200 iterations. In this case the concentrations evolve as shown in Figure 7.6.

Parameter setup for Example 7.4. Competing parallel reactions

Starting configuration: 2500 A cells on a 50 × 50 grid
Parameters: P_T(A → B) = 0.04 iteration^{-1}, P_T(A → C) = 0.01 iteration^{-1}
Run for 200 iterations
Plot numbers of A, B, and C cells versus iterations. Note the final concentrations of B and C.
Repeat two more times and from the three runs determine averages and the final concentrations of [B] and [C]
Compare these values with the results expected from the transition probabilities.

In this particular example reaction, one finds that after 200 iterations roughly 80% of the A ingredients have converted to the B form and 20% have converted to the C form. Thus the ratio of [B]/[C] is about 4:1. In the end, because the

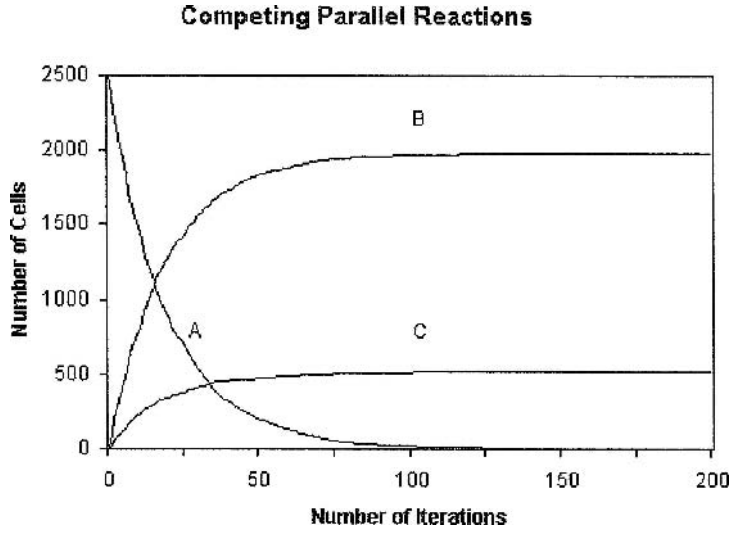


Figure 7.6. Plot of concentrations of A, B, and C ingredients for the competing reactions A → B and A → C as described in Example 7.4

transition probability P_T (A → B) is four times that of P_T (A → C), the B cells predominate over the C cells by about this same ratio. The numerical differences observed in runs of this type are due to the inherently stochastic (random) nature of the processes involved.

Study 7.4. Small-scale and large-scale competing reactions

Repeat the exercise shown in Example 7.4 using P_T (A → B) = 0.01 and P_T (A → C) = 0.03, with a 100-cell grid. Use three trial runs and compute the average final concentrations of species B and C along with their standard deviations. Repeat this using 2500 ingredients on a 2500 cell grid. Compare the two results with special attention to the ratios of the standard deviations to the final concentrations.

7.5. Kinetic and thermodynamic reaction control

Example 7.5. Reversible competing reactions: kinetic and thermodynamic reaction control [4]

One can add reverse reactions to the parallel reaction model to illustrate what chemists refer to as kinetic and thermodynamic reaction control. Often a reactant A can form two (or more) products, one of which (B) is formed rapidly (the kinetic product) and another (C) which forms more slowly (the thermodynamic

product), but which is favored by a greater equilibrium constant. Depending upon which product the synthetic chemist directing the reaction most desires the reaction can be stopped after a short time (to get B) or a longer period (to get C).

To illustrate this, we shall start with 2500 A ingredients and set the transition probabilities to P_T (A → B) = 0.01, P_T (B → A) = 0.02, P_T (A → C) = 0.001, and P_T (C → A) = 0.0005. Note that these values yield a situation favoring rapid initial transition to species B, since the transition probability for A → B is 10 times than that for A → C. However, the formal equilibrium constant K_{eq}[C]/[A] is 2.0, whereas K_{eq}[B]/[A] = 0.5, so that eventually, after the establishment of equilibrium, product C should predominate over product B. This study illustrates the contrast between the short run (kinetic) and the long run (thermodynamic) aspects of a reaction. To see the results, plot the evolution of the numbers of A, B, and C cells against time for a 10,000 iteration run. Determine the average concentrations $[A]_{avg}$, $[B]_{avg}$, and $[C]_{avg}$ under equilibrium conditions, along with their standard deviations. Also, determine the iteration n_{Bmax} at which ingredient B reaches its maximum value.

Parameter setup for Example 7.5. Kinetic and thermodynamic reaction control

Starting configuration: 2500 A cells on a 50 × 50 grid
Parameters: P_T(A → B) = 0.01, P_T(B → A) = 0.02, P_T(A → C) = 0.001, P_T(C → A) = 0.0005
Run for 5000 iterations
Plot numbers of A, B, and C cells versus iterations for the first 200 iterations and determine the maximum concentration reached for ingredient B
Note also the iteration at which $[B]_{max}$ occurs
Determine the equilibrium concentrations for A, B, and C by taking averages (and standard deviations) for these ingredients over the last 2000 iterations.

Figure 7.7 shows a typical plot for this experiment. Typical results for this experiment are $[B]_{max} \approx 750$ and $n_{Bmax} \approx 130$, but these values will vary from run to run.

Study 7.5. Further study of kinetic and thermodynamic reaction control

Repeat Example 7.5 using the following transition probabilities: P_T (A → B) = 0.01, P_T (B → A) = 0.009, P_T (A → C) = 0.001, P_T (C → A) = 0.0001. Determine the same properties as before.

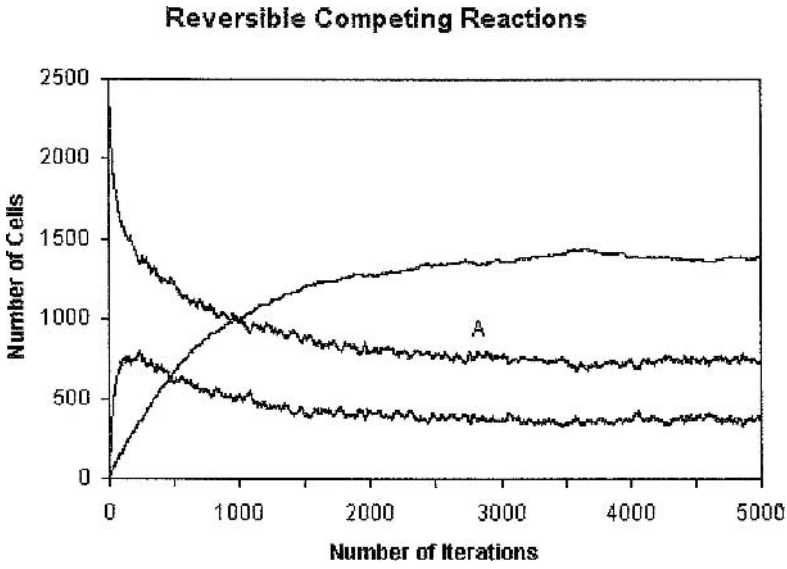


Figure 7.7. Plot of the numbers of A, B, and C ingredients illustrating kinetic and thermodynamic reaction control as described in Example 7.5

7.6. Pre-equilibrium

A number of chemical processes involve a pre-equilibrium in which a reversible process A $\leftrightarrow$ B is coupled with the transformation B $\rightarrow$ C [2,5]:

$$\mathrm{A} \underset{k_2}{\overset{k_1}{\longleftrightarrow}} \mathrm{B} \xrightarrow{k_3} \mathrm{C} \tag{7.10}$$

Frequently the rate constants k_1 for the forward reaction A $\rightarrow$ B and k_2 for the reverse reaction are much larger than k_3, so that the pseudo-equilibrium A $\leftrightarrow$ B is strongly coupled and the reaction B $\rightarrow$ C serves merely to "drain off" the product C from the equilibrium. If we define the "pseudoequilibrium constant" $K_{\mathrm{AB}} = k_1/k_2$, then the rate of formation of C is

$$\text{Rate} = \mathrm{d[C]}/\mathrm{d}t = k_3[\mathrm{B}] \approx k_3 K_{\mathrm{AB}}[\mathrm{A}] \tag{7.11}$$

Accordingly, this arrangement serves to make the rate of the formation of product C largely dependent on the concentration of A.

Example 7.6. A pre-equilibrium condition

We can simulate the pre-equilibrium condition by taking P_{T} (A $\rightarrow$ B) = 0.2, P_{T} (B $\rightarrow$ A) = 0.2, and P_{T} (B $\rightarrow$ C) = 0.004. We can use a $50 \times 50 = 2500$ cell torus grid and start with 2500 blue A ingredients and no B or C ingredients. Note that the k_1 and k_2 values are much greater than k_3, so that when it is

formed an ingredient B will be 50 times more likely to revert to A than to convert to product C. Gradually, however, the product C ingredients will be produced. We can follow the production of C and also the changes of the A and B concentrations as they vary with time over 200 iterations.

Parameter setup for Example 7.6. Pre-equilibrium

Starting configuration: 2500 A cells on a 50×50 grid
Parameters: $P_T(A \rightarrow B) = 0.2$ iteration^{-1}, $P_T(B \rightarrow A) = 0.2$ iteration^{-1}, $P_T(B \rightarrow C) = 0.004$ iteration^{-1}
Run for 200 iterations
Plot numbers of A, B, and C cells versus iterations
Note the slope (in units of ingredients/iteration) of the line for the formation of ingredient C.

Note the slope (in units of ingredients/iteration) of the line for the formation of ingredient C. Figure 7.8 shows a typical plot of the cell concentrations for this example.

Study 7.6. Pre-equilibrium

Repeat the above example using a $100 \times 100 = 10{,}000$ cell grid. Plot numbers of A, B, and C cells versus iterations. Compare the production

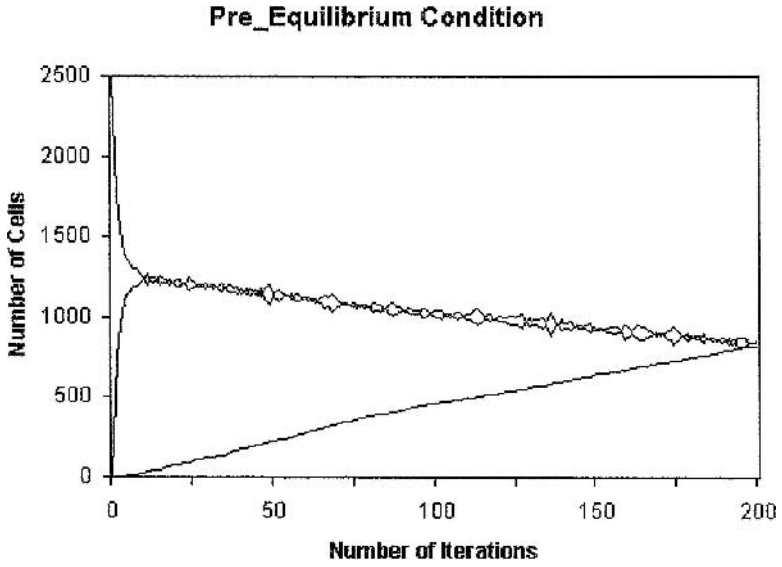


Figure 7.8. A typical plot of the concentrations of A, B, and C ingredients for the pre-equilibrium conditions given in Example 7.6

rate (i.e., the slope) for formation of product C with that obtained in the example.

References

1. Quoted in Life Magazine, Jan. 9, 1970.
2. P. G. Seybold, L. B. Kier, and C.-K. Cheng, Simulation of first-order chemical kinetics using cellular automata. *J. Chem. Inf. Comput. Sci.* 1997, 37, 386–391.
3. T. L. Brown and H. E. LeMay, Jr., *Chemistry: The Central Science*, 3rd ed. Prentice Hall, Inc., Englewood Cliffs, NJ, 1985, 614–615.
4. A. Neuforth, P. G. Seybold, L. B. Kier, and C.- K. Cheng, Cellular automata models of kinetically and thermodynamically controlled reactions. *Int. J. Chem. Kinet*. 2000, 32, 529–534.
5. P. W. Atkins, *Physical Chemistry*, 5th ed. W. H. Freeman, New York, 1994.

Chapter 8

SECOND-ORDER CHEMICAL KINETICS

It seems to me that all great understandings of the big things emerge from meticulous attention to the detailed workings of the minutiae . . . . Even the novel behaviors of complex systems arising from complexity itself are ultimately comprehensible only in terms of the underpinnings—as with Boltzmann's treatment of a huge collection of newtonian billiard balls vomiting up that humdinger of an emergent property, the second law of thermodynamics.

—Chris Miller[1]

So-called second-order reactions—those involving the encounters of two ingredients–lie at the heart of chemistry. Indeed, the classic prototype of a chemical reaction takes the general form

$$A + B \rightarrow C + D, \tag{8.1}$$

where reactants A and B interact to form products C and D. For an *elementary reaction* of this form, i.e., one simply involving molecules A and B reacting by means of a bimolecular collision, the *rate law*, the formula describing the dependence of the reaction rate on the reactant concentrations, is

$$Rate = k[A][B], \tag{8.2}$$

where k is the reaction rate constant (units of time^{-1}concentration^{-1}), and [A] and [B] are the concentrations of the reactants A and B. (By *concentration*, we formally mean the number of ingredients of a given species divided by the total number of grid cells. However, in many cases it is simpler to refer to the number of ingredients of each species as the "concentration" of that species.) Because in this case the reaction rate depends directly on the first powers of [A] and [B] the rate law is said to be first-order with respect to each of these species, and, referring to the total reaction, second-order overall. Stated in the

terms of calculus,

$$-\frac{d}{dt}[A] = -\frac{d}{dt}[B] = +\frac{d}{dt}[C] = +\frac{d}{dt}[D] = k[A][B] \tag{8.3}$$

where the terms of the form $\frac{d}{dt}[X]$ represent the rates of change of the different "X" (= A, B, C, and D) species with time. It is clear that as the concentrations of the reactants A and B fall during the reaction, the concentrations of the products C and D will increase in the same amount.

For the special case in which reactant A reacts with itself to form products, i.e., A + A → C + D, a simplification is obtained, and the rate law becomes

$$\text{Rate} = \text{k[A]}^2. \tag{8.4}$$

Solving this equation yields a formula for the (inverse) concentration of A as a function of time t:

$$\frac{1}{[A]} = kt + \frac{1}{[A]_0}, \tag{8.5}$$

where $[\text{A}]_0$ is the concentration of A at t = 0. Equation (8.5) is seen to have the form of a straight line, y = mx + b, in which y = 1/[A], x = t, the slope m = k, and the y-intercept b = $1/[\text{A}]_0$. For this particular case, the *half-life* $t_{1/2}$, the time it takes for the concentration of species A to fall to half its original value, depends on the starting concentration, as

$$t_{1/2} = 1/\text{k[A]}_0. \tag{8.6}$$

Thus the higher the starting concentration of A, the shorter the half-life, and as the reaction proceeds (and [A] decreases) the half-life lengthens.

Many, if not most, of the key reactions of chemistry are second-order reactions, and understanding this type of reaction is central to understanding chemical kinetics. Cellular automata models of second-order reactions are therefore very important: they can illustrate the salient features of these reactions and greatly aid in this understanding.

8.1. Second-order cellular automata models

In the previous chapter we examined cellular automata simulations of first-order reactions. Because these reactions involved just transformations of individual ingredients, the simulations were relatively simple and straightforward to set up. Second-order cellular automata simulations require more instructions than do the first-order models described earlier. First of all, since movement is involved and ingredients can only move into vacant spaces on the grid, one must allow a suitable number of vacant cells on the grid for movement to take place in a sensible manner. For a gas-phase reaction one might wish to allow at least 5–10 vacant cells for each ingredient, so that on a 100 × 100 = 10,000

cell grid there should be at most 1000–2000 ingredients, the rest of the 8000–9000 cells being unoccupied. Next, one must define the rules that apply to the system in question. For movement, these rules are embodied in the free moving probability P_m and the influences on movements expressed by the breaking and joining parameters, P_B and J, respectively. When reactions are involved one must also designate the reaction probability per encounter, P_R(A,B), for each pair (A,B) of reacting species. This probability defines the likelihood that a transition from reactants (A,B) to products (C,D) will occur when the ingredients A and B come in contact (move to adjacent cells within the von Neumann neighborhood). Usually gravity rules are not invoked in this type of simulation, although they could be if desired.

Second-order simulations typically start with the initial ingredients placed randomly into the cells of the grid (although this is not necessary and can be changed). Thus the initial configuration of the system normally assumes a random character. As the simulation proceeds the ingredients move about the grid under the influences of the P_B and J rules and subject to the designated reaction probabilities P_R. Because of these several random features—which in fact have their counterparts in nature—each simulation run is in effect a "new" experiment, with different starting conditions and almost certainly a different set of configurations through which the system passes in the course of its evolution. Accordingly, one might anticipate that each simulation might lead to a rather different evolutionary sequence. Indeed this is the case if one speaks of specific absolute configurations. However, in practice, although the specific configurations involved will differ, for a sufficiently large sample of ingredients the *collective* pattern that emerges—the "emergent property" of second-order kinetic behavior—tends to look much the same, especially when a substantial number of ingredients are participating.

In what follows, unless specified otherwise the breaking and joining parameters, P_B and J, will be assigned the neutral values $P_B = 1.0$ and $J = 1.0$ appropriate to "hard-sphere" (billiard ball) collisions. In some cases it will be of interest to depart from this simple model and to alter these values to find the influences of intermolecular attractions and repulsions on the results.

The following examples and exercises will exhibit the basic concepts of second-order kinetics. Some more specialized applications related to this topic will be given in Chapter 9.

8.2. Irreversible second-order reactions $A + B \rightarrow C + D$, small scale[2]

Application 8.1. Irreversible second-order reactions, small scale

In order to illustrate the randomness inherent in these systems, let us first examine a small system consisting of a $20 \times 20 = 400$ cell grid containing

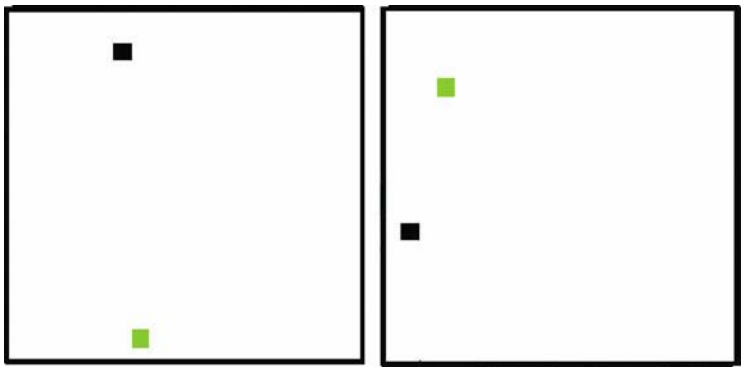


Figure 8.1. Two typical starting configurations for Application 8-1, showing random placement of two ingredients on a 20 × 20 grid.

just a single A ingredient and a single B ingredient on a torus grid. We shall set the reaction probability per encounter of these two ingredients at $P_R(AB) = 0.10$, so that on average one of every 10 "collisions" of these two ingredients will lead to a reaction transforming them into C and D ingredients. We shall employ ten runs of 5000 iterations each in the simulation. What we want to record for each run is the specific iteration at which the transforming "reaction" occurs. Figure 8.1 shows two of the 10 starting configurations generated by the program for this exercise, illustrating typical initial conditions. It should be noted that even if the two ingredients start at positions close to one another, because the ingredients then undergo random walks about the grid it does not necessarily follow that they will encounter each other more quickly, and even if they encounter each other rapidly they will not necessarily "react" more quickly.

Parameter setup for Example 8.1. Irreversible second-order reaction, small scale

Grid: $20 \times 20 = 400$ cells on a torus
Parameters: $P_m = 1.0$, $J = 1.0$, $P_B = 1.0$, $P_R(AB) = 0.1$
A cells (blue) = 1
B cells (green) = 1
C cells (red) = 0
D cells (brown) = 0
Number of runs = 10
Simulation length for each run = 5000 iterations
Report the iteration at which the reaction $A + B \rightarrow C + D$ occurs for each run. Determine the average and median reaction iterations for the set of 10 results.

In one set of ten runs the following reaction times were observed (placed in numerical order): 35, 185, 204, 346, 444, 454, 780, 843, 925, and a long value of 2771. The median time for this set of runs was 449 itn., and the average time for reaction was 699 itn. A second collection of ten runs would very likely yield a quite different set of results.

Study 8.1a. Repeating the small scale reaction

Repeat the example above and compare your results with those obtained for Example 8.1. This study illustrates the stochastic nature of these small-scale reaction systems, and the fact that each simulation is an independent experiment.

Study 8.1b. Changing the reaction probability $P_R(AB)$

Change the reaction probability $P_R(AB)$ to 1.0, and let the simulation run for 1000 iterations. At what time (what iteration) does reaction occur? Repeat this simulation nine more times and tabulate the results. Find the average time and its standard deviation for your results, as well as the median time. Next change $P_R(AB)$ to 0.05, increase the number of iterations for each run to 5000, and tabulate the results for 10 trial runs. Repeat the averaging process above. This study reveals the influence of the reaction probability on the course of the reaction.

Study 8.1c. Changing the "concentration"

In a further test, increase the grid size to 100×100, set $P_R(AB)$ at 1.0, and repeat the process. This test reveals the influence of concentration on the rate, since the two ingredients now have a much larger territory to roam about, making them in effect "less concentrated." Compare your results with the results from the $P_R(AB) = 1.0$ part of Study 8.1b.

8.3. Irreversible second-order reaction $A + B \rightarrow C + D$, large scale

Application 8.2. Irreversible second-order reaction, large scale

As a contrasting example, we now examine the same type of reaction, but with a large number of interacting ingredients. We shall begin with a 100×100 cell grid with 500 A (blue) cells and 500 B (green cells) and a reaction probability $P_R(AB) = 0.05$ for the formation of C (red) and D (brown) products. This means that the conversion from (blue, green) to (red, brown), indicated in Eq. (8.1) will, on average, take place in one of every twenty (blue, green)

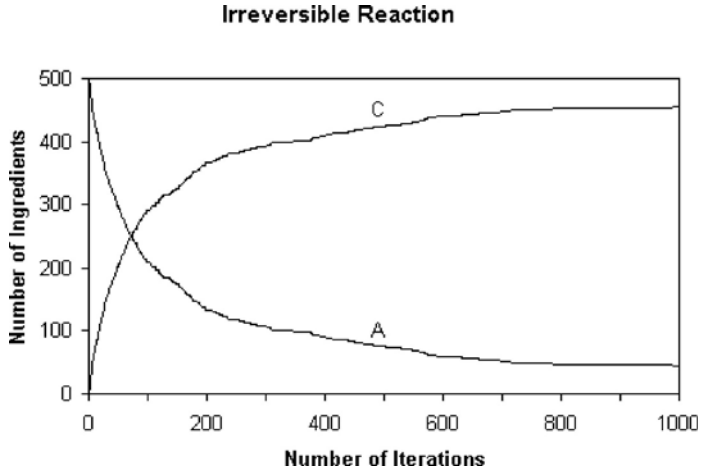


Figure 8.2. Time variations of the concentrations of ingredients A and C for the second-order reaction $A + B \rightarrow C + D$.

encounters. We allow the system to run for 1000 iterations, and observe that indeed as time goes by the A's and B's are converted to C's and D's. The variations in the concentrations of [A] (= [B]) and [C] (= [D]) as the simulation proceeds are illustrated in Figure 8.2. At first the concentration of C rises steeply, later less steeply, and finally [C] asymptotically approaches its limiting value [C] = 500, as the few remaining A's and B's eventually react.

This example illustrates the point that there is "strength in numbers", in this case statistical strength. Although the behavior of any single ingredient is unpredictable, the overall statistical behavior of a large number of identical ingredients does become, within well-defined limits, predictable.

Parameter setup for Example 8.2: Large-scale irreversible reaction

Grid: $100 \times 100 = 10{,}000$-cell torus
Parameters: P_m, J, and $P_B = 1.0$, $P_R(AB) = 0.05$
A cells (blue) = 500
B cells (green) = 500
C cells (red) = 0
D cells (brown) = 0
Number of runs = 1
Simulation length = 1000 iterations
Plot the concentration of the A and C cells vs. iterations n over this time frame. Determine the initial rate from the first linear portion of the [C] vs. n plot; determine k from this rate.

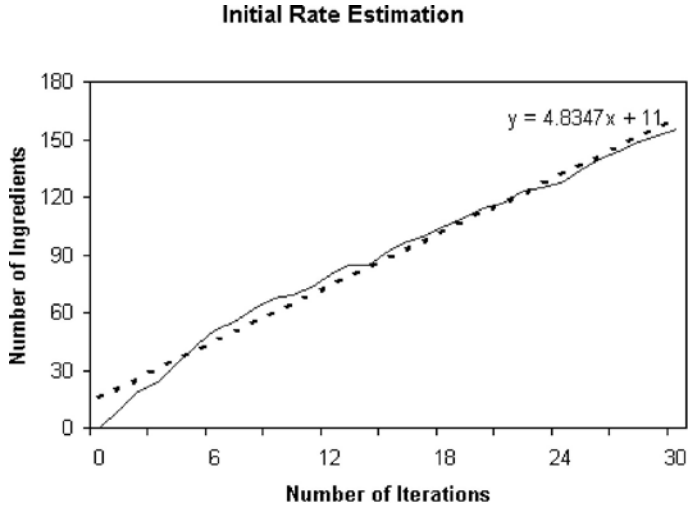


Figure 8.3. Initial variations in the number of product C ingredients for the A + B → C + D reaction of Example 8.2. Note the brief initial adjustment period before the system settles into linear behavior.

The initial rate of the reaction can be determined from the early linear portion of the plot of [C] vs. n. (Note that there is often a brief adjustment period before the system settles into normal kinetic behavior. This brief adjustment period should not be included in the initial linear period.) This is illustrated in Figure 8.3, where the initial slope is found to be 4.83 ingredients/iteration (ingr/itn). Based on Eq. (8.2), the value of the rate constant k for this system is $4.83/(500)^2 \approx 2.0 \times 10^{-5}$ ingr^{-1}itn^{-1}.

Study 8.2a. Changing the reactant concentrations

We have not yet completely examined how the reactant concentrations affect the reaction rate. In this study we shall decrease the initial concentrations of the reactants A and B and observe the results. Keep the parameters the same as in Example 8.2, but now decrease the starting concentrations of A and B to $[A]_0 = 250$ and $[B]_0 = 250$. Plot the results as [C] vs. n and determine the new rate from the initial linear portion of the plot. Also, determine a value of k from a plot of the early portion of [C] vs. n.

Study 8.2b. Different starting reactant concentrations for A and B

We saw in Study 8.2a that decreasing the concentrations of both A and B decreased the reaction rate, but we have not tested what happens when the concentrations of the reactants A and B are not the same. Set up your simulation

as in Example 8.2, but now with $[A]_0 = 400$ cells and $[B]_0 = 600$ cells. Plot the concentrations of the ingredients as they change with time. Determine the initial reaction rate, taking the first early linear portion of the plot of [C] vs. time in iterations. Determine the rate constant and compare this value with the result in Example 8.2. You might wish to repeat the process using different values of $[A]_0$ and $[B]_0$.

Application 8.2a. Self reaction: A + A → C + D

We now can investigate the self-reaction A + A → C + D, as described in the introduction to this Chapter. Here we will use 1000 A ingredients on a 100 × 100 cell grid, and set $P_R(AA) = 0.01$. Run for 1000 iterations. Find k from the initial slope of a plot of [C] vs. n. Compare the k value obtained from the initial slope with that obtained from a plot of $1/[A]$ vs. n as described in Eq. (8.5).

Parameter setup for Example 8.2a. Self-reaction kinetics

Use Example 8.2 with the following conditions:
Grid: 100 × 100 = 10,000-cell torus
Parameters: P_m, J, and $P_B = 1.0$,
$P_R(AA) = 0.01$ for reaction A + A → C + D
A cells (blue) = 1000
B cells (green) = 0
C cells (red) = 0
D cells (brown) = 0
Number of runs = 1
Simulation length = 1000 iterations
Plot the concentrations of the A and C cells vs. iterations n over this time frame. Determine the initial reaction rate from the first linear portion of the [C] vs. n plot; also determine k from this plot. Next plot $1/[A]$ vs. n and determine k from this plot. Compare this value of k with that from the initial reaction rate. Does the y-intercept agree with the expectation from Eq. (8.5)?

8.4. Second-order equilibrium, A + B ⇆ C + D

Application 8.3. Second-order equilibrium

Next we shall expand the previous example to include the back reaction C + D → A + B, yielding the equilibrium condition A + B ⇆ C + D. This

is analogous to the first-order equilibrium discussed in Chapter 7. For this case the equilibrium constant is given by the expression

$$K_{eq} = [C][D]/[A][B]. \tag{8.7}$$

We shall set $P_R(AB) = 0.05$ and set the reaction probability for the back reaction at $P_R(CD) = 0.01$. Since the equilibrium constant K_{eq} can equally well be expressed as $K_{eq} = P_R(AB)/P_R(CD)$, once the simulation has come to a steady state we expect to find the ratio $K_{eq} = [C][D]/[A][B]$ to be about 5. We allow the simulation to run for 1200 iterations.

Parameter setup for Example 8.3. Second-order equilibrium

Grid size $100 \times 100 = 10{,}000$ cells (torus)
Parameters: P_m, J, and $P_B = 1.0$, $P_R(AB) = 0.05$, $P_R(CD) = 0.01$
A cells (blue) = 500
B cells (green) = 500
C cells (red) = 0
D cells (brown) = 0
Number of runs = 1
Simulation length = 1200 iterations
Report the concentrations of A and C cells and plot [A] and [C] versus iterations for the last 500 iterations. Determine the average equilibrium concentrations of A and C, along with their standard deviations. Also determine the equilibrium constant K_{eq}.

From the plot of the concentrations of [A] and [C] in Figure 8.4 it is apparent that a steady-state, or equilibrium, condition is achieved after about 300 iterations. This is not a static equilibrium, however, and during this period the values of [A] and [C] fluctuate a great deal about their average values. Taking averages for the last 500 iterations of the run we find that for this specific run $[A]_{avg} = 154.6 \pm 5.0$ and $[C]_{avg} = 345.4 \pm 5.0$. The actual instantaneous values for [A] range from a low of 140 to a high of 166 during this 500-iteration period, and the values for [C] vary from 334 to 360. Since [A] = [B] and [C] = [D], the observed average values yield $K_{eq} = [C][D]/[A][B] = (345.6)^2/(154.6)^2 = 4.99$, a number fortuitously close to the deterministic value of $K_{eq} = 5.0$. However, an analysis of the instantaneous values of K_{eq} obtained during the final 500 iterations shows that these instantaneous values fluctuate over a considerable range—from a minimum of 4.05 to a maximum of 6.61. The fluctuations demonstrate the stochastic behavior associated with relatively small, finite samples of interacting ingredients. Using these instantaneous values for K_{eq} one finds $K_{eq} = 5.02 \pm 0.47$. Again, this value is fortuitously close to 5.0, and in

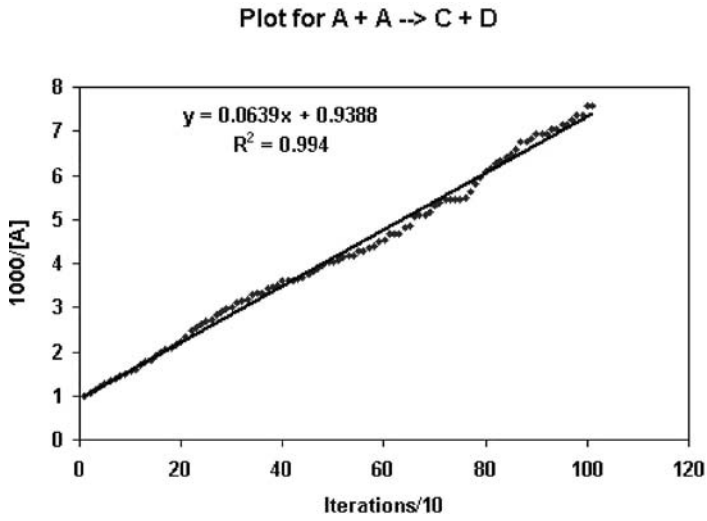


Figure 8.4. Plot of Eq. 8.5 for the self-reaction A + A → C + D of Example 8.2a.

most cases such excellent agreement is not found for a short period such as that examined here. However, if longer periods are studied, say the last 5000 itn. of a 6000 itn. run, very good agreement with the deterministic expectation is usually found.

Study 8.3a. Repeating the example for a longer time

Repeat Example 8.3, but now using a run of 6000 itn. Estimate K_{eq} from the data for the last 5000 itn.

Study 8.3b. Starting from the product side

Example 8.3 above showed that equilibrium was achieved when we started with reactants A and B, but what happens when we approach this from the opposite direction, i.e., starting from the product side? We can test this by starting with $[A]_0 = [B]_0 = 0$ cells, and $[C]_0 = [D]_0 = 500$ cells. Run the simulation just as in Example 8.3, but with these changed initial values. Is equilibrium achieved from the product direction? If so, what is the equilibrium constant K_{eq}? (Include the uncertainty in K_{eq}.)

Study 8.3c. Changing the starting concentrations

In Study 8.1c we examined how the reactant concentrations affected the forward reaction rate, but we have not yet examined how such a change influences the equilibrium condition. Change the initial concentrations to $[A]_0 = 700$ cells

and $[B]_0 = 700$ cells and compare your results with those obtained in Example 8.3. Is the equilibrium constant affected?

Study 8.3d. Unequal reactant concentrations

Repeat Example 8-3 but now with unequal initial concentrations of A and B. Use $[A]_0 = 400$ cells and $[B]_0 = 600$ cells. Perform the same tests as before and comment on the results. In particular, determine K_{eq}.

8.4. Effects of ingredient interactions on the rates

It is now interesting to see if changing the breaking probability $P_B(A,B)$ or the joining parameter J(A,B) affects the reaction rate. For this we first set up the situation as in Example 8.2 above, with a 100 × 100 grid and 500 A and 500 B cells, but now set $P_B(AB) = 0.5$ (instead of $P_B = 1.0$). This means that the A and B cells in contact with each other have some tendency to stay in contact.

Parameter setup for Example 8.4. Effect of $P_B(AB)$ on the rate

Grid: 100 × 100 = 10,000 cells (torus)
Parameters: $P_R(AB) = 0.05$, $P_B(AB) = 0.5$, $J(AB) = 1.0$,
A cells (blue) = 500
B cells (green) = 500
C cells (red) = 0
D cells (brown) = 0
Number of runs = 1
Simulation time = 1000 iterations
Report the concentrations of A and C cells and plot [A] and [C] versus iterations, n. Is the initial rate (slope of a plot of [C] vs. time) changed?

A typical result for this experiment is shown in Figure 8.5. The result of increasing the propensity of A and B to stick together when they meet is to increase the reaction rate.

Study 8.4a. Effect of the breaking probability $P_B(AB)$ on equilibrium

A further test of the equilibrium concept, introduced in Example 8.3, involves introducing the influence of the breaking probability into the equilibrium process. In this case repeat the process of Example 8.2, but with the change of $P_B(AB)$ to 0.5. Do you expect an effect? What do you observe? What is K_{eq} for this case?

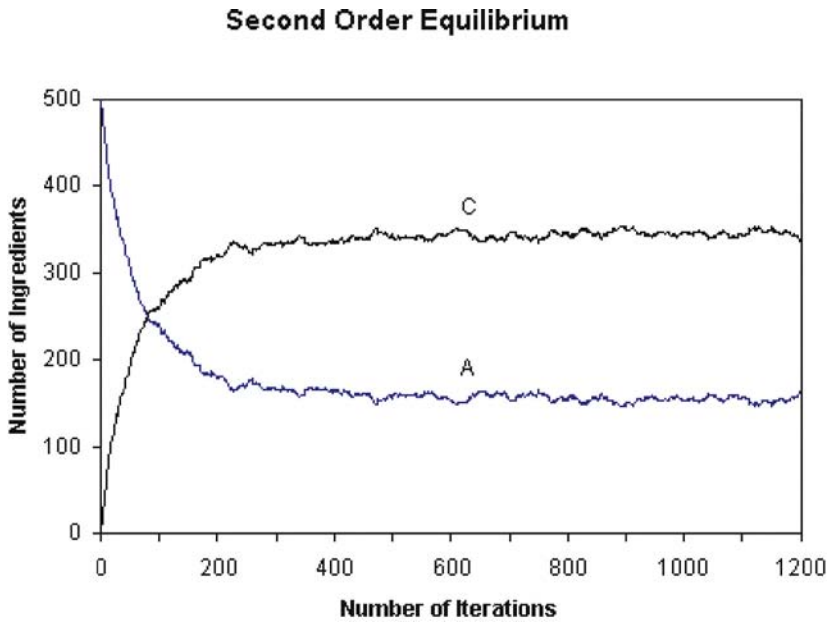


Figure 8.5. Concentration changes of ingredients A and C for the equilibrium $A + B \leftrightarrows C + D$ of Example 8.3.

Study 8.4b. Effect of the joining probability J(AB) on equilibrium

In Example 8.4 we saw that increasing the stickiness of A and B (by decreasing $P_B(AB)$ increased the reaction rate of the reaction $A + B \rightarrow C + D$. Now it is time to see what the influence of the joining probability J is on the reactions. Repeat the simulation of Example 8-1, but with a change of J(AB) from 1.0 to 2.0. This change increases the tendencies of A and B to move toward each other when they are separated by a vacant cell. Plot your results as in Exercise 8.2, and describe any changes observed. Next, repeat the exercise with $J(AB) = 0.5$. Again, plot your results and describe any changes in the results.

Study 8.4c. The influence of an inert species

Many gas-phase reactions are carried out with the addition of a "filler" species, an inert gas that increases the pressure, collides with the reactants, and otherwise influences the course of the reaction. It is interesting to observe whether such an added inert species has an effect on the kinetic results. To the conditions of Example 8.2 add 500 "E" cells, which do not interact with the

other species other than by occupying space on the grid. That is, the E cells have both $P_B(EX) = 1.0$ and $J(EX) = 1.0$, for X = A, B, C, D, and E, and $P_R(EX) = 0$. Compare the concentration vs. time curves obtained in this way with those from Example 8.2. What effect did adding E have on the results? What happens with 1000 E cells?

References

1. C. Miller, *Nature* **2003**, *425*, 564.
2. J. Moore and P. G. Seybold, unpublished results.

Chapter 9

ADDITIONAL APPLICATIONS IN CHEMICAL KINETICS

Complex systems can be identified by what they do (display organization without a central organizing authority—emergence) and also by how they may or may not be analyzed (as decomposing the system and analyzing sub-parts do not necessarily give a clue as to the behavior of the whole).

—J. M. Ottino [1]

In the previous two chapters we examined a number of examples of how cellular automata models can be used to simulate basic first-order and second-order phenomena. In this chapter, we shall examine some examples that are slightly more complicated and/or more specialized. First, we will look at the example of enzyme kinetics based on the classic Michaelis–Menten model. Next, we shall look at the equilibrium between a vapor and its liquid, watching the vapor condense under the joint influences of ingredient attractions and gravity. Then, we will consider a particular chemical reaction mechanism, the Lindemann mechanism, employing a simplified first-order approach. Finally, we shall model the kinetics associated with transitions involving atomic and molecular excited states, focusing on the specific example of the excited states of atomic oxygen responsible for the spectacular *Aurora borealis* displays.

9.1. Enzyme kinetics

The functioning of enzymes produces phenomena driving the processes which impart life to an organic system. The principal source of information about an enzyme-catalyzed reaction has been from analyses of the changes produced in concentrations of substrates and products. These observations have led to the construction of models invoking intermediate complexes of ingredients with the enzyme. One example is the Michaelis–Menten model, postulating an

intermediate enzyme–substrate complex, followed by changes in the substrate leading to a product.

$$\text{Enzyme (A)} + \text{substrate (B)} \leftrightarrow \text{enzyme–substrate complex (AB)} \leftrightarrow$$
$$\text{enzyme–product complex (AC)} \leftrightarrow \text{enzyme (A)} + \text{product (C)}$$

Enzyme reactions, like all chemical events, are dynamic. Information coming to us from experiments is not dynamic even though the intervals of time separating observations may be quite small. In addition, much information is denied to us because of technological limitations in the detection of chemical changes. Our models would be improved if we could observe and record all concentrations at very small intervals of time. One approach to this information lies in the creation of a model in which we know all of the concentrations at any time and know something of the structural attributes of each ingredient. A class of models based on computer simulations, such as molecular dynamics, Monte Carlo simulations, and cellular automata, offer such a possibility.

Application 9.1. Modeling an enzyme reaction

The goal in this application is to explore the possibility of using cellular automata to model some of the phenomena of enzyme reactions and to determine if there is a potential here for acquiring new information about these processes. These studies are conducted by varying the rules relating the enzyme, substrate, product, and water to themselves and to the other ingredients. The dynamics are allowed to proceed and the initial velocities and the progress of the product concentration are observed. This information is compared to a Michaelis–Menten model. Some general inferences about the influence of substrate–water and substrate–enzyme relationships on rates of reaction are made, illustrating the potential value of cellular automata in these studies. Kier and colleagues [2–4] have used cellular automata to model enzyme reactions.

Example 9.1. An enzyme-catalyzed reaction

The cellular automata dynamics are run on the surface of a torus to eliminate a boundary condition. The grid of cells is $110 \times 110 = 12{,}100$ cells. When additional ingredients are introduced, they are assumed to displace the water cell count on a one for one basis; thus the 31% cavity compliment is always maintained. The ingredients and their designations are enzyme A, substrate B, product C, and water D. The rules govern the movement, joining, and breaking of cell ingredients with each other and with neighboring cell ingredients. The rules take the form of probabilities. Each cell type has a set of parameters governing its relationship to itself and to all other cell ingredients.

The enzyme, A, is allowed to join with only one molecule of B, C, or D but not another A. Thereafter, further joining with any other ingredient is not

possible. The A cells are constrained to remain at some designated minimum distance from any other A cell. The parameter, P_R, is employed that describes the probability of an AB pair of cells changing to an AC pair of cells. The objective in these studies is to determine if Michaelis–Menten kinetics are observed from our cellular automata model of an enzyme reaction. Then fifty of the 12,100 cells are designated as enzymes and a variable number of cells are designated as substrates. The remaining cells of the grid are designated as water or cavities, the latter comprising 31% of the grid space. The P_R value was arbitrarily chosen to be 0.1 and $P_B(DD) = 0.375$ was chosen to simulate water at body temperature. From the Michaelis–Menten equation, there is a high degree of linearity between the reciprocals of the concentration and the initial velocity.

Parameter setup for Example 9.1. Rate of enzyme-catalyzed reaction

Grid 110 × 110 cells on a torus
Water, D, 7350 cells
Enzyme, A, 50 cells
Substrate, B, 500 cells (4%)
Product, C, 0 cells
$P_M(A) = 1.0$, $P_M(B) = 1.0$, $P_M(C) = 1.0$, $P_B(DD) = 0.375$, $J(DD) = 1.1$, $P_B(DB) = 0.75$, $J(DB) = 0.3$, $P_B(AB) = 0.75$, $J(AB) = 0.3$, $P_B(AC) = 0.75$, $J(AC) = 0.3$, $P_R(AB \rightarrow AC) = 0.1$, $P_R(AC \rightarrow AB) = 0$
Record the number of C cells at 80 iterations
Run for 100 iterations, 100 times to determine an average initial velocity, V_0.

The number of C cells recorded at 80 iterations divided by 80 is taken to be the initial velocity of the reaction, V_0. The initial concentration $[B_0]$ is taken to be the number of B cells at the start.

$$V_0 = [C]/80$$

From the Michaelis–Menten model, there is a relationship between $1/V_0$ and the initial substrate concentration, expressed as the reciprocal, $1/[B_0]$. To develop this relationship we shall repeat Example 9.1 using varying concentrations of B cells. Be sure to subtract the number of B_0 cells in each study from the total number of water, D, cells in the setup.

Studies 9.1a–d. Variation of substrate concentration

A systematic variation of the rules governing the interactions of B and A can be made to reveal the influence of substrate concentration, B_0. Run

the example again using B_0 concentrations of 1000, 1500, 2000, and 2500 cells. Be sure to subtract the same number of cells from the water D used in each study. Using the $[C]_{80}$ concentration as a measure of the extent of the reaction at a common time, in this case 80 iterations, it is possible to derive some understanding of the influence of interingredient relationships. The Lineweaver–Burk relationship, $1/[V_0]$ versus $1/[B_0]$, is calculated from this data. Plot this data using $1/V$ for the y-axis and $1/[B_0]$ for the x-axis. Calculate the relationship

$$1/V_0 = a \times 1/[B_0] + b$$

where a is the slope of the line and b is the intercept of the line on the plot. The y-axis intercept, b, is taken to be the reciprocal of the maximum velocity, $1/V_0$, of the reaction for that particular enzyme. The x-axis intercept of the line is taken to be the negative reciprocal of the K_m value. K_m is interpreted to be the equilibrium constant for the enzyme–substrate system.

$$A + B \leftrightarrow AB$$

Finally, a term reflecting the efficiency or competence of the enzyme, k_{cat}, is derived

$$k_{cat} = V_{max}/[A_0]$$

Studies 9.1e–f. Variation of the enzyme conversion probability

The influence of the P_R probability on this enzymatic process can be evaluated using different values of this parameter. In Example 9.1, the $P_R(A + B \rightarrow A + C)$ was chosen to be 0.1. Repeat that example and the four substrate concentration studies in Studies 9.1a–d using the same parameters but changing the P_R values to 0.4. Calculate the V_0, V_{max}, K_m, and k_{cat} values for this set of input variables. Repeat the process again using a P_R value of 0.7. Compare the Lineweaver–Burk plots and all the V_0, V_{max}, K_m, and k_{cat} values for these three P_R values. In essence, this is a comparison of three different enzymes acting on the same substrate. Rank them according to their catalytic efficiency in carrying out the reaction. These examples and studies clearly illustrate the power of cellular automata models in this area of chemistry. The reader is invited to consider the extensions of these procedures such as introducing an "inhibitor" molecule to retard the process. A series of reactions could be designed such as

$$B + A_1 \rightarrow BA_1 \rightarrow CA_1 \rightarrow C + A_1$$
$$C + A_2 \rightarrow CA_2 \rightarrow DA_2 \rightarrow D + A_2$$
$$D + A_3 \rightarrow \rightarrow \rightarrow \text{ and so on} \ldots$$

This has been done illustrating a feed-forward process [3]. Another application of these multistep reactions is the study of metabolic networks. Kier and colleagues have reported on such an example [4].

9.2. Liquid–vapor equilibrium

Application 9.2. Modeling liquid–vapor behavior

In Chapter 2, we examined the effect of introducing a gravity effect on the movements of the CA ingredients (Example 2.4). Now, we are ready to see what happens if we combine the gravity effect with attractions between the moving ingredients. First, however, we shall see what happens if we omit the gravity term and just allow attractions between the ingredients. This should give the equivalent of a "mist," as happens in moist air containing many small droplets of water molecules that are light enough not to settle quickly to the ground. After our initial examination, we shall add a gravity term and see what happens.

Example 9.2. Liquid–vapor equilibrium

First, we examine the case where the ingredients attract one another in the absence of gravity. The setup is shown as follows:

Parameter setup for Example 9.2. Liquid–vapor equilibrium

1500 ingredients placed on a 100 × 100 cylinder grid Parameters: $P_m = 1.0$, $P_B(AA) = 0.25$, and $J(AA) = 2.0$ Run length = 500 iterations Note the general character of the appearance of the system. Does it resemble a mist?

Study 9.2a.

Now, introduce an absolute gravity term $G_A(A) = 0.15$ and allow the simulation to run for 15,000 iterations. Note the appearance of the system and print out a picture of your screen at the end of the run.

Study 9.2b.

Increase the absolute gravity parameter to $G_A(A) = 0.30$, and repeat Study 9.2a.

9.3. The Lindemann mechanism [5]

A "mechanism" in chemical kinetics refers to a set of elementary reactions (i.e., reactions occurring on the molecular scale) that together lead to the observed overall macroscopic behavior. One of the more interesting of these mechanisms is the so-called "Lindemann mechanism," named after the British physicist F. A. Lindemann. This mechanism was devised to explain the curious phenomenon that certain gas phase chemical reactions appeared to be unimolecular, i.e., taking place without collisions, at high pressure, whereas the same reactions obeyed normal second-order collisional kinetics at low pressure. The Lindemann mechanism explained the observed results as arising from the generation of an "activated" intermediate form A* from collisions of the reacting species A with an inert ingredient M. The activated species A* could then either (1) be deactivated back to its parent species A by further collisions with M or (2) proceed to form a product P.

The reaction mechanism is described by the following set of elementary reactions:

$$\begin{aligned} &\mathrm{A} + \mathrm{M} \xrightarrow{k_1} \mathrm{A}^* + \mathrm{M} \\ &\mathrm{A}^* + \mathrm{M} \xrightarrow{k_2} \mathrm{A} + \mathrm{M} \\ &\mathrm{A}^* \xrightarrow{k_3} \mathrm{P} \end{aligned} \tag{9.1}$$

In this mechanism, conversion of the activated species A* to the product P (the last reaction) competes with deactivating collisions of A* with the species M (the second reaction).

The usual chemical kinetics approach to solving this problem is to set up the time-dependent changes in the reacting species in terms of a set of coupled differential rate equations [5,6].

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}[\mathrm{A}] &= k_1[\mathrm{A}^*][\mathrm{M}] - k_2[\mathrm{M}][\mathrm{A}] \\ \frac{\mathrm{d}}{\mathrm{d}t}[\mathrm{A}^*] &= -k_1[\mathrm{A}^*][\mathrm{M}] + k_2[\mathrm{M}][\mathrm{A}] - k_3[\mathrm{A}^*] \\ \frac{\mathrm{d}}{\mathrm{d}t}[\mathrm{P}] &= k_3[\mathrm{A}^*] \end{aligned} \tag{9.2}$$

Usually the steady-state approximation $\mathrm{d}[\mathrm{A}^*]/\mathrm{d}t \approx 0$ is applied at this point to help in solving the equations. This approximation postulates that the concentration of the intermediate A* remains relatively constant over a suitable portion of the reaction time. The result is the following expression for the reaction

rate v,

$$v = \frac{\mathrm{d}}{\mathrm{d}t}[\mathrm{P}] = \frac{k_1 k_3[\mathrm{A}][\mathrm{M}]}{k_3 + k_2[\mathrm{M}]} \tag{9.3}$$

Note here that at high pressures of M, $k_2[\mathrm{M}] \gg k_3$ and Eq. (9.3) reduces to the first-order rate expression $v \approx (k_1 k_3/k_2)[\mathrm{A}] = k'[\mathrm{A}]$, whereas at low pressures $k_2[\mathrm{M}] \ll k_3$ and the expression becomes $v \approx k_1[\mathrm{A}][\mathrm{M}]$, the normal second-order form. (Approximations such as these are commonly used in many areas of science and mathematics.)

Application 9.3. The Lindemann mechanism [5]

The cellular automata approach to this problem would generally demand a second-order simulation, and this indeed can be done [5]. However, it is possible to simplify the problem and write it in terms of first-order kinetics. To do this, one notes that both the activation and the deactivation of A* depend directly on the concentration [M] of the species M

$$\mathrm{Rate}_1 = k_1[\mathrm{A}][\mathrm{M}]$$
$$\mathrm{Rate}_2 = k_2[\mathrm{A}^*][\mathrm{M}]$$

Therefore, one can introduce the pseudo-first-order rate constants $k'_1 = k_1[\mathrm{M}]$ and $k'_2 = k_2[\mathrm{M}]$ for these processes, so that the abovementioned rate expressions become

$$\mathrm{Rate}_1 = k_1'[\mathrm{A}]$$
$$\mathrm{Rate}_2 = k_2'[\mathrm{A}^*]$$

This redefinition establishes the effective activation/deactivation equilibrium constant $K = k'_1/k'_2 = k_1/k_2$. (Note that in the cellular automata models, the rate constants k_i are expressed as transition probabilities per iteration P_i.) Using the above redefinition, the mechanism of Eq. (9.1) becomes a set of first-order reactions

$$\begin{aligned} \mathrm{A} &\xrightarrow{k'_1} \mathrm{A}^* \\ \mathrm{A}^* &\xrightarrow{k'_2} \mathrm{A} \\ \mathrm{A}^* &\xrightarrow{k_3} \mathrm{P} \end{aligned} \tag{9.4}$$

Thus, the competition between deactivation of the intermediate A* and product formation is given in terms of the ratio $\alpha = k'_2/k_3$. When the second-order rate constants k_1, k_2, and k_3 are set for the system, the ratio α is directly proportional to the pressure [M], since $\alpha = (k_2/k_3)[\mathrm{M}]$. Thus, the effect of varying [M], the variable in the Lindemann mechanism that defines the pressure, can

be simulated by varying the value of α while keeping k_3 and the ratio $K = k_1/k_2$ constant. Accordingly, in this cellular automaton model by simultaneously reducing the transition probabilities k'_1 and k'_2 (while keeping their ratio $K = k'_2/k'_1$ constant), one simulates a reduction in [M], and conversely increasing these probabilities simulates an increase in [M].

Example 9.3. The Lindemann mechanism

We shall set $K = 0.5$ and vary k'_1 and k'_2 from high values (high pressure) to low values (low pressure) while keeping their ratio steady at 0.5. We shall hold k_3, the rate for conversion of A* to P, steady at $k_3 = 0.01$ per iteration. The aim in this example is to measure the initial rate of product formation in terms of P formed per iteration. This can be found by measuring the initial linear part of the slope of a plot of [P] versus iterations. (Caution: there is usually a short "induction period" before the linear region is established. Do not include this period in your analysis.) We will use a grid of 10,000 ingredients. Let A = A (the starting ingredient), B = A* (the activated intermediate), and C = P (the product). Start with $k'_1 = P_T(\text{A} \rightarrow \text{B}) = 0.495$ iteration^{-1} and $k'_2 = P_T(\text{B} \rightarrow \text{A}) = 0.99$ iteration^{-1} and decrease these values in steps until $k'_1 = 0.000005$ iteration^{-1} and $k'_2 = 0.00001$ iteration^{-1}.

Parameter setup for Example 9.3. The Lindemann mechanism

Grid $100 \times 100 = 10{,}000$ cell torus grid
10,000 starting A ingredients
Starting transition probabilities:

$$P = \begin{pmatrix} 0.0 & 0.495 & 0.0 \\ 0.99 & 0.0 & 0.01 \\ 0.0 & 0.0 & 0.0 \end{pmatrix}$$

Run for enough iterations to obtain a linear early region for a plot of P versus time.

Determine the initial slope of this linear region and record it. Then, repeat lowering $P_T(\text{A} \rightarrow \text{B})$ to 0.25 iteration^{-1} and $P_T(\text{B} \rightarrow \text{A})$ to 0.50 iteration^{-1}, but keeping $P_T(\text{B} \rightarrow \text{C}) = 0.01$. Again, record the initial linear rate of formation of product P. Repeat again, but this time lowering $P_T(\text{A} \rightarrow \text{B})$ to 0.05 iteration^{-1} and $P_T(\text{B} \rightarrow \text{A})$ to 0.10 iteration^{-1}. Continue to reduce $P_T(\text{A} \rightarrow \text{B})$ and $P_T(\text{B} \rightarrow \text{A})$ in further steps, holding their ratio steady at 0.5 and keeping $P_T(\text{B} \rightarrow \text{C}) = 0.01$ iteration^{-1}. Finally, take $P_T(\text{A} \rightarrow \text{B}) = 0.000005$ iteration^{-1} and $P_T(\text{B} \rightarrow \text{A}) = 0.00001$ iteration^{-1} and record the initial production rate of P.

When you have all the initial rates R_i, plot $\log_{10}(R_i)$ versus $\log_{10}\alpha$, where $\alpha = k'_2/k_3 = P_T(B \rightarrow A)/0.01$. Note the form of this plot, which should pass from an initial linear region at low α to a flat final region at high α. Note that α is proportional to the pressure [M].

Study 9.3. The Lindemann mechanism variation in *K* parameter

Repeat Example 9.3 using $K = 0.2$ instead of $K = 0.5$. Again plot this result.

9.4. Excited-state photophysics [6]

Quantum theory, developed in the early years of the 20th century, forced scientists to accept the surprising fact that atoms and molecules cannot exist at arbitrary energies, but rather can exist only in certain discrete energy states, these states being characteristic of the particular atoms or molecules in question. Spectroscopy involves the study of the transitions between these discrete quantum energy states. The energy states themselves can be associated with the electronic, vibrational, and rotational motions that the molecules undergo. Molecular spectra observed in the visible and ultraviolet regions of the electromagnetic spectrum involve mostly transitions between different electronic energy states of the atoms and molecules. A second notion resulting from quantum theory was that the individual electrons, which together determine the electronic energy states, have an inherent property called *spin*, and, furthermore, that the electronic spins can only take on two forms, which for want of better names were termed "up" and "down." As a consequence, the electronic states of atoms and molecules exist in a variety of different overall spin versions, called "singlet," "doublet," "triplet," etc., states, depending on how the individual electron spins are aligned with respect to one another. Normally, transitions between electronic states with different overall spins are inhibited, i.e., they have slower rates than the transitions between electronic states of the same spin.

Usually the lowest or "ground" states of organic molecules are so-called "singlet" states in which all the spins of the electrons present are all paired up, "up" spins coupled with "down" spins. When subjected to a magnetic field, singlet states do not split in energy. In contrast, "triplet" electronic states, in which two of the electron spins have become uncoupled (both are up, or both down), split into three slightly different energy levels under the influence of an imposed magnetic field. The resulting excited-state situation is most easily visualized by a *Jablonski diagram*, as shown in Figure 9.1. The usual condition is that the molecular ground state is a singlet state (S_0), and there exists a

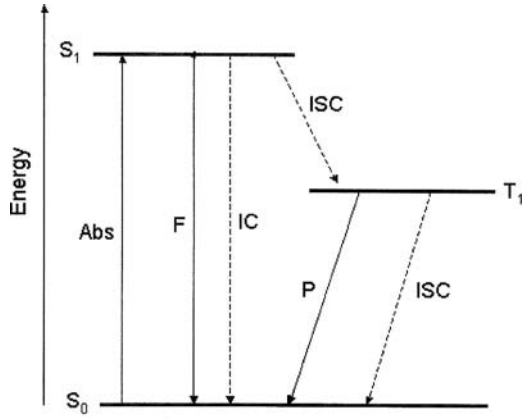


Figure 9.1. A Jablonski diagram. S_0 and S_1 are singlet states, T_1 is a triplet state. Abs, absorption; F, fluorescence; P, phosphorescence; IC, internal conversion; and ISC, intersystem crossing. Radiative transitions are represented by full lines and nonradiative transitions by dashed lines

host of excited states having either singlet (S_n) or triplet (T_n) character. The observed visible/ultraviolet absorption spectrum of the compound consists of transitions from the ground state S_0 to the excited singlet states S_1, S_2, S_3, etc. Absorptions to the excited triplet states are strongly inhibited ("forbidden") by the spin restriction mentioned above and do not normally appear in the spectrum.

Photophysics is the study of the transitions between the ground state and the excited states. If one is interested primarily in the luminescence of the molecules, i.e., the radiative transitions from the excited states to the ground state, it is usually sufficient to study only the lowest excited singlet and triplet states, S_1 and T_1, respectively. This is because of Kasha's rule, which states that the emitting state of a given spin multiplicity (singlet or triplet, in the present case) is the lowest excited state of that multiplicity [7]. This rule follows from the fact that excited states of the same spin are usually close in energy and strongly coupled to one another, so that after excitation to a higher excited singlet state, say S_2 or S_3, collisions with surrounding molecules quickly remove the excess energy and cause rapid nonradiative (without radiation) decay to the lowest excited singlet state S_1. Once at S_1, however, the molecules face a rather large energy gap to the ground state. Fluorescence, the spin-allowed radiative transition from S_1 to the ground state, is usually a fast process, taking place within several nanoseconds (10^{-9} s) of excitation. S_1 can also decay to S_0 without radiation via a process called internal conversion (IC) that competes with fluorescence. In addition, S_1 is normally close in energy to T_1, so that despite the spin restriction nonradiative transitions from S_1 to T_1 can also take place, a process called intersystem crossing (ISC). Thus, three processes can

occur from S_1: fluorescence $S_1 \rightarrow S_0$, ISC $S_1 \rightarrow T_1$, and nonradiative IC $S_1 \rightarrow S_0$.

If the molecule reaches the triplet state T_1, two further decay transitions can occur: radiative decay $T_1 \rightarrow S_0$ called phosphorescence and nonradiative ISC decay $T_1 \rightarrow S_0$. Because of the spin restriction involved and the rather large T_1-S_0 energy gap, both processes are rather slow and can have characteristic times ranging from microseconds (10^{-6} s) to seconds. In solution at room temperature, the long-lived (by molecular standards) triplet states of organic compounds are usually quenched by molecular oxygen dissolved in the solution and phosphorescence is not normally seen. (Molecular oxygen, O_2, has a triplet ground state and can quench other triplet states.) However, in low-temperature (e.g., liquid nitrogen temperature, 77 K) rigid glasses and under some other special conditions, it can be possible to observe phosphorescence from organic compounds. Moreover, many gaseous atoms and solid inorganic compounds phosphoresce strongly at room temperature.

Let us now examine how a luminescent organic substance might behave following elevation to an excited state.

Application 9.4. Excited-state decay following light absorption

When a compound absorbs light it is carried to a higher energy state. From there it will decay to lower energy states, possibly emitting a photon in the process, and eventually it will return to its ground state. We shall follow a typical case wherein a sample of identical compounds is excited by a light pulse to S_1 and allowed to cascade down through several possible pathways. For simplicity, we will ignore the IC $S_1 \rightarrow S_0$ from the excited singlet state to the ground state, since in many fluorescent dyes this decay route is not important. We cannot say just what an individual ingredient will do, but we will set the probabilities such that on an average 30% of the compounds fluoresce and 70% cross over to the triplet state T_1. This should result in an expected fluorescence quantum yieldof $\varphi_f \approx 0.30$ for a large sample and a corresponding triplet quantum yield of $\varphi_T \approx 0.70$, these numbers reflecting the fractions of ingredients taking each path. Once in the triplet state T_1, the compounds can either decay by emitting radiation as phosphorescence or decay nonradiatively to the ground state. Because of the large energy gap and the spin-forbidden natures of the $T_1 \rightarrow S_0$ decay transitions, their probabilities per unit time are much lower than those for the singlet state. The phosphorescence quantum yield φ_p is the product of the fraction of ingredients crossing over to the triplet state (here the triplet quantum yield $\varphi_T = 0.7$) and the fraction of ingredients reaching the triplet that then phosphoresce. The latter fraction is given by the ratio of the T_1 radiative decay probability per unit time to the total decay probability per unit time for T_1. In the present case, we will set the radiative probability

at $P_{\text{rad}}(T_1 \rightarrow S_0) = 0.00004$ and the nonradiative probability at $P_{\text{non}}(T_1 \rightarrow S_0) = 0.00006$, so that $\varphi_p = (0.7)(0.00004/0.00010) = 0.28$. These values, $\varphi_f = 0.7$ and $\varphi_p = 0.28$, correspond to the deterministic limiting values for these properties. (Note that the triplet state decay rates have been set higher than those normally encountered for purposes of illustration.)

We will first follow the decay paths taken during several individual runs, just to see how they can vary. Then we will examine the behavior of larger samples to find actual values for φ_f and φ_p for the samples. Since the cellular automata models are stochastic, the results for φ_f and φ_p for small samples will likely differ significantly from the deterministic values cited above. The differences between the observed and the deterministic values will normally decrease as the sample size is increased. We will also examine the observed "lifetimes" τ_f and τ_p of the decays of the S_1 and T_1 states (Chapter 7) and compare the values found with the corresponding deterministic values.

Example 9.4. Pulse excitation

The somewhat simplified CA program accompanying this text does not record the specific state-to-state transitions taking place during each iteration, so that we cannot use it directly to follow the paths taken by each individual ingredient. In order to determine these paths and their respective quantum yields, we would like some simple means for following these paths. In particular, we now have *two* different decay paths between state B (representing T_1) and state C (representing S_0): a radiative path with $P_{\text{rad}}(T_1 \rightarrow S_0) = 0.00004$ and a nonradiative path with $P_{\text{non}}(T_1 \rightarrow S_0) = 0.00006$, but our first-order transformation matrix only permits one path between any two states. We shall now introduce a simple "trick" to solve both of these problems. The "trick" is to divide the ground (i.e., C) state into three equivalent parts, one for each of the paths. To do this, we add states C′ and C″, as shown in Figure 9.2, to the state C. Thus the fluorescent $S_1 \rightarrow S_0$ path will go directly from state A to state C, the phosphorescent decay path from S_1 to T_1 to S_0 will go from A → B → C′, and the $S_1 \rightarrow T_1 \rightarrow S_0$ nonradiative path will go A → B → C″. Keeping track of the populations of C, C′, and C″ gives us the counts for these three paths. The total ground state population at any time will be the sum of the populations of C, C′, and C″. In Example 9.4, we shall examine some single decays so that we can get an idea of how the system works. Then in the studies, we will take the investigation further.

Parameter setup for Example 9.4. Pulse excitation

Grid size $1 \times 1 = 1$ cell
Starting A cells (blue) = 1 (representing the S_1 state)

Starting B cells (green) = 0 (representing the T_1 state)
Starting C cells (red) = 0 (S_0 for the fluorescence decay)
Starting C′ cells (brown) = 0 (S_0 for the phosphorescence decay)
Starting C″ cells (yellow) = 0 (S_0 for the nonradiative decay from state T_1)
The transition probability matrix is:

$$P = \begin{pmatrix} 0.0 & 0.70 & 0.30 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.000004 & 0.000006 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{pmatrix}$$

(Note that the numbers in the above matrix are the transition probabilities for each path)
Number of runs = 10
Simulation length for each run = 5000 iterations
For each of the 10 separate runs note the decay path taken.

Study 9.4a. Behavior of a 100-cell sample

We now consider a sample of 100 ingredients on a 10 × 10 cell grid. All of the starting ingredients will be blue, so they are starting in their S_1 states. With this sample size, although it is relatively small, we will begin to see the emergence of a pattern in the decays, although the pattern is not very clearly defined because of the small size of the sample. Maintain the other conditions, the transition probabilities, of the simulation the same as in Example 9.1, but now use just a single run of 5000 iterations. Determine φ_f and φ_p for this sample and compare these values with the deterministic values. (Note again that the

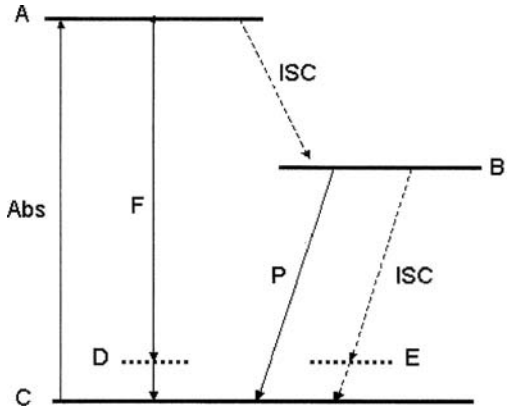


Figure 9.2. Cellular automaton setup for photophysics showing the final states C, C′, and C″

total numbers of ingredients taking the fluorescent route A → C, the radiative triplet decay route A → B → C′, and the nonradiative decay path A → B → C″ can be found simply by noting the final populations of the C, C′, and C″ states, respectively.)

Study 9.4b. Extension to a larger sample

We shall now examine the behavior of a fairly large sample of 10,000 cells using the same conditions as mentioned above. Again, use a single 5000 iteration run. (Check to see how many of the 10,000 starting A ingredients have ended up in the C states. Lengthen the run if too many have not completed their decays after 5000 iterations.) Determine φ_f, φ_p, τ_f, and τ_p for this sample and compare these values with the deterministic values. For the lifetimes τ_f and τ_p, plot ln(A or B) versus time for the first 70% of each decay period and determine the decay rates k from the slopes. The lifetimes are the inverses of the slopes, $\tau = 1/k$.

Application 9.5. Continuous excitation

In the above illustrations, we looked at the behavior of cells after a burst of excitation, where all the ingredients start in the excited S_1(A) state. In many, if not most, luminescence experiments the excitation is *continuous*, i.e., the sample is constantly illuminated so that the fluorescence is continuously observed. Simulation of this experimental mode requires the introduction of an excitation probability from the ground state (C, C′ or C″ in this case) to the excited singlet state (A), i.e., a value for the absorption probability $P_{abs}(S_0 \rightarrow S_1)$. In general, this value will depend on three factors: (1) the incoming light intensity, (2) the spectral nature of the incoming light, and (3) the ability of the luminescent compound to absorb the incoming light.

Thus, this value will depend on the experimental conditions, including the light source, the sample studied, and the overall optical setup. For the present illustration, we will set $P_{abs}(S_0 \rightarrow S_1) = 0.005$ per iteration.

Example 9.5. Continuous excitation

In this example, Example 9.2, we shall start with all the cells in the ground state S_0. Again, we will use the three parts of the ground state, C, C′, and C″, to keep track of the paths taken during this continuous process.

Parameter setup for Example 9.5. Continuous excitation

Grid size 100 × 100 = 10,000 cells
Starting A cells (blue) = 0 (representing the S_1 state)

Starting B cells (green) = 0 (representing the T_1 state)
Starting C cells (red) = 10,000 (S_0 for the fluorescent decay path)
Starting C′ cells (brown) = 0 (S_0 for the phosphorescence decay path)
Starting C″ cells (yellow) = 0 (S_0 for the nonradiative decay from state T_1)
The transition probability matrix is:

$$P = \begin{pmatrix} 0.0 & 0.70 & 0.30 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0004 & 0.0006 \\ 0.005 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.005 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.005 & 0.0 & 0.0 & 0.0 & 0.0 \end{pmatrix}$$

(In the above matrix, the numbers are the transition probabilities for each path)
Number of runs = 1
Simulation length = 10,000 iterations
Plot the concentrations (numbers) of the species A, B, C, C′, and C″ as a function of time
After how many iterations (roughly) is a steady-state condition reached? What are the steady-state concentrations of the different states (S_0, S_1, and T_1)? (Determine the latter over the final 5000 iterations.) Determine the quantum yields φ_f and φ_P from your results

A typical result is shown in Figure 9.3. Note that the concentration of S_0 initially falls as the ingredients are excited to S_1, but eventually reaches a relatively steady value after about 500 iterations as ingredients return to this state through various routes. During this same period, the triplet state (T_1) population first increases and then also reaches a relatively steady value. Because of its very short lifetime, the population of the excited singlet state S_1 never becomes very large, and is not much evident in Figure 9.3. Recall that to determine the total S_0 population, one must sum the values of the C, C′, and C″ populations, giving C_{tot}. Taking the interval between 1000 and 2000 iterations as a reasonable sample for the steady-state region, it was found for the sample run illustrated in Figure 9.3 that $[S_0]_{avg} = 7376 \pm 48$. Likewise, $[S_1]_{avg} = 37 \pm 6$ and $[T_1]_{avg} = 2588 \pm 46$. Breaking down the components of the S_0 state population for the 1000–2000 iteration interval gave $[C]_{avg} = 2219 \pm 33$, $[C']_{avg} = 2077 \pm 29$, and $[C'']_{avg} = 3079 \pm 38$. Using our "trick" described earlier, these numbers can be used to estimate the fluorescence quantum yield φ_f, the triplet yield φ_T, and the phosphorescence yield φ_p. Here $\varphi_f = [C]_{avg}/[C_{tot}]_{avg} = 2219/7376 = 0.301$, a value very close to the deterministic value of 0.3. Likewise, $\varphi_T = ([C']_{avg} + [C'']_{avg})/[C_{tot}]_{avg} = 5156/7376 = 0.699$, and $\varphi_p = [C']_{avg}/[C_{tot}]_{avg} = 2077/7376 = 0.282$.

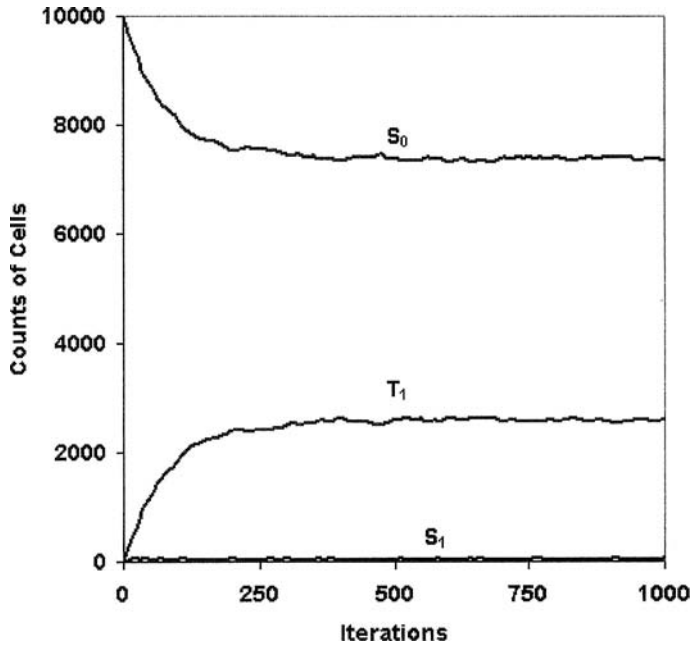


Figure 9.3. Illustration of the approach to steady-state conditions for the populations of states S_0, S_1, and T_1 under the continuous excitation conditions of Example 9.5

Study 9.5a. Repeating the result

Repeat the above study, now using a smaller $50 \times 50 = 2500$ cell sample. After the sample has been allowed to come to a steady state, determine the steady-state concentrations of S_0, S_1, and T_1, along with their standard deviations. Also determine the quantum yields φ_f and φ_P.

Study 9.5b. Increasing the light intensity

Repeat Example 9.2 (with 10,000 cells) after increasing the absorption rate to 0.05 iteration^{-1}. Note especially the steady-state concentrations of S_0, S_1, and T_1.

Study 9.5c. Excited-state dynamics of oxygen atoms [8,9]

Light emissions from excited oxygen atoms play a significant role in the *A. borealis*, or northern light displays, seen in the skies of the northern polar regions [10]. Similar light displays (*Aurora australis*) are also seen above the southern polar regions and in the atmospheres of Mars and Venus. Several atmospheric processes lead to the formation of excited oxygen atoms in their so-called ^{1}S (singlet S) and ^{1}D (singlet D) states, which lie 4.19 and 1.97 eV,

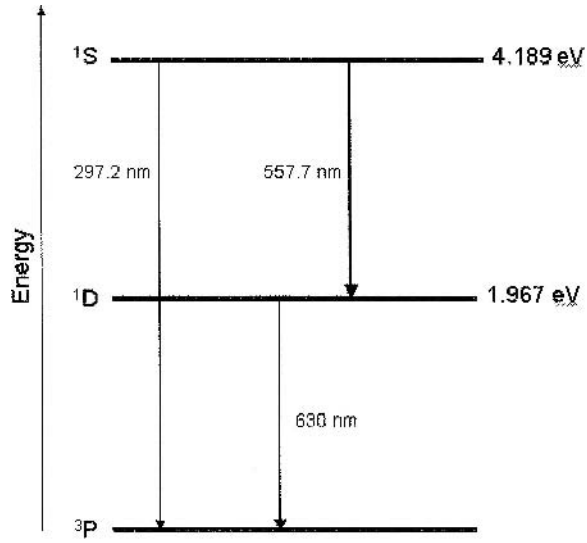


Figure 9.4. Excited states and transitions of oxygen atoms. The numbers on the lines refer to the wavelengths, in nanometers, of the transitions

respectively, above the atomic ground ^{3}P (triplet P) state of the oxygen atom. The energetic arrangement of these states is illustrated in Figure 9.4.

Since both the excited (^{1}S and ^{1}D) states are singlet states and the ground (^{3}P) state is a triplet state, transitions between the excited states and the ground state are spin-forbidden and have low probabilities (and hence long lifetimes). The $^1\text{S} \rightarrow {}^3\text{P}$ emission appears weakly at 297.2 nm in the ultraviolet region, and the $^1\text{D} \rightarrow {}^3\text{P}$ emission appears weakly at 630 nm in the red region. A stronger, spin-allowed emission, $^1\text{S} \rightarrow {}^1\text{D}$, appears in the green region at 557.7 nm and is found prominently in many auroral displays. Okabe [11] has compiled rate constants for a great number of spectroscopic transitions, including those for the oxygen atomic states. He lists the following rate constants for the above transitions:

$$P(^1\text{S} \rightarrow {}^3\text{P}) = 0.067 \text{ s}^{-1}$$
$$P(^1\text{D} \rightarrow {}^3\text{P}) = 0.0051 \text{ s}^{-1}$$
$$P(^1\text{S} \rightarrow {}^1\text{D}) = 1.34 \text{ s}^{-1}$$

In this study, set up the parameters to analyze the dynamics of the oxygen atoms in the atmosphere based on the above data. Use a $50 \times 50 = 2500$ cell grid and start with all the cells in the excited ^{1}S state. Let A = the ^{1}S state, B = the ^{1}D state, and C = the ground ^{3}P state. Start with all of the ingredients in the excited ^{1}S state. To convert from Okabe's rate constants, given in units of s^{-1}, to probabilities per iteration, divide the above values by 10 so that 10 iterations

correspond to 1 s. (Note that our transition probabilities must always be ≤ 1.0.) First, try a run of 2000 iterations. Plot the time course of the A, B, and C populations for this system. Determine the decay times, $\tau(^1\mathrm{S})$ and $\tau(^1\mathrm{D})$, for the two excited states.

References

1. J. M. Ottino, Engineering complex systems. *Nature*. 2004, 427, 399.
2. L. B. Kier, C.-K. Cheng, and B. Testa, A cellular automata model of enzyme kinetics. *J. Mol. Graph*. 1996, 14, 227–232.
3. L. B. Kier and C.-K. Cheng, A cellular automata model of an anticipatory system. *J. Mol. Graph*. *Model*. 2000, 18, 29–33.
4. L. B. Kier, D. Bonchev, and G. Buck, Cellular automata models of enzyme networks. *Chem. Biodivers*. 2005, 2, 233–243.
5. C. A. Hollingsworth, P. G. Seybold, L. B. Kier, and C.-K. Cheng, First-order stochastic cellular automata simulations of the Lindemann mechanism. *Int. J. Chem. Kinet*. 2004, 36, 230–237.
6. R. A. Alberty and R. J. Silbey, *Physical Chemistry*, 2nd ed. Wiley, New York, 1997, 648–651.
7. P. Atkins. *Physical Chemistry*, 6th ed. Freeman, New York, 1998, 782–785.
8. P. G. Seybold, L. B. Kier, and C.-K. Cheng, Stochastic cellular automata models of molecular excited-state dynamics. *J. Phys. Chem*. 1988, 102, 886–891.
9. P. G. Seybold, L. B. Kier, and C.-K. Cheng, Aurora borealis: stochastic cellular automata simulations of the excited-state dynamics of oxygen atoms. *Int. J. Quantum Chem*. 1999, 75, 751–756.
10. N. Davis, *The Aurora Watcher's Handbook*. University of Alaska Press, Fairbanks, 1992.
11. H. Okabe, *Photochemistry of Small Molecules*. Wiley, New York, 1978, p. 370.

Chapter 10

USE OF THE *CASim* PROGRAM

Only a new kind of science could begin to cross the great gulf between knowledge of what one thing does—one water molecule, one cell of heart tissue, one neuron—and what millions of them do.

—James Gleick[1]

This chapter prepares the reader to experience the exciting adventure of running *in silico* experiments. The **Examples** and **Studies** in Chapters 2 through 9 are coordinated with files in the *CASim* program, found on the compact disk accompanying this book. In the program the procedures for introducing or changing parameters, running studies, and collecting data have been simplified to provide what we hope is a user-friendly and fairly general means for performing cellular automata simulations. It is essential that the reader master the details in the present chapter very early in his or her work. Once this mastery is accomplished, the Examples and Studies can be run with minimum distraction and with maximum understanding of what cellular automata models can do and show. So, read, practice, learn, and enjoy.

The cellular automata simulation program *CASim* allows the user to select a variety of parameters and ingredient types, and to establish appropriate behavioral rules, initial configurations, and report formats. Allowing this flexibility, however, adds complexity to the system, and requires some attention to detail on the part of the user. For this book, we decided to structure the simulation program around the Examples provided, so that the reader does not need to learn the full procedures for setting up a number of different types of simulations. This format does, however, restrict the reader to simulations related to the Examples provided. In most cases, depending on how the Example is designed, the reader is permitted to change the parameter values, the size of the grid, and the numbers of the different ingredients. In some cases, however, it has been necessary to keep the grid size or ingredient numbers constant, and

in these cases the system will disable some functions (see below for specific details). In most cases the ingredient *types*, initial configuration set-up, and the information reported are pre-selected and cannot be changed.

In many cases readers will wish to change the parameters of an Example in order to explore the consequences. In order to provide for such changes, *CASim* uses so-called **Setting files** (with ".set" as the extension). Each Setting file must be associated with a specific Example. The Setting file therefore allows variations from the specific conditions of the Example to be examined. There are two kinds of Setting files: **core** and **non-core**. The core Setting files are those initially present on the *CASim* CD and use the Example names as their file names. For example, the core Setting file associated with Example2-1 is called Example2-1.set. The non-core Setting files are additional Setting files created by the user (see below). You can delete non-core Setting files, but you should not delete core Setting files, although you are free to change their parameters (see below).

10.1. System requirements

This software is written in Java programming language. To run this Java application, the J2SE (v 1.4.2 or higher version) Java runtime environment must be installed. There is a free download web site http://java.sun.com/j2se/1.4.2/download.html. The *CASim* software is designed to run on any computer that supports Java 1.4.2 or later Java runtime environment. (This includes almost all Windows, Macintosh, and linux systems.)

10.2. Installation of *CASim* on your computer

Installation is accomplished by simply copying the *CASim* directory from the accompanying CD onto the hard drive of your personal computer.

10.3. Organization of the CAProgram Directory

The *CASim* directory contains three subdirectories: CAExample, CAOutput and CAProgram. In the CAProgram directory, there are three files—CASim.class, CASim.jar, and CASim.BAT.

In the CAExample directory there are eight subdirectories, one for each chapter of this book (except Chapter 1 and the present Chapter). Each chapter subdirectory contains a Hidden subdirectory that contains certain files needed to set up the examples in that chapter properly. **Do not change anything in these Hidden subdirectories**, since changes in these files may prevent the program from working properly. In addition, each chapter subdirectory contains core

Setting files, one for each example in the chapter. Again, do not delete core Setting files.

The CAOutput directory is initially empty. It is the directory into which the output of each simulation will be placed. Please see the **Data Collection** section below for details.

10.4. Data Collection

When a simulation is completed, the data gathered during the simulation are collected into a report. This report, as a text file, will be found in the CAOutput directory with a file name indicating the Example, the Setting file, and the time of the simulation. For instance, the file name for a report of a simulation which ran Example3-1 with Setting file abc.set on May 10 at 13th hour and 45 minutes would be

Ex3-1_abc_05101345.txt.

In this form, "05101345" repesents mm/dd/hh/mm. Expressing the file in this way reduces the chance of accidentally overwriting a previous report.

You may wish to rename the output file so that it will not be overwritten when you run the same Example again. For this, it is recommended that you choose a name that you can easily relate to the appropriate topic and Example. You may want to remove this file from the CAOutput subdirectory, to keep the directory from becoming too crowded.

10.5. Graphic User Interface window and Graphic Simulation window

The system employs two windows. One is the **Graphic User Interface window** which provides facilities for the user to load, edit, and/or to view various parameter settings. Furthermore, the user can issue commands to save the settings and to run the simulation. The other window is **the Graphic Simulation window** which presents the dynamics of the ingredients movement in the grid. Both windows can be closed by clicking the "cross" at the right top corner of the window. These two windows are very much related, therefore the following two rules must be observed.

1. **The Graphic Simulation window** should be closed before the **Graphic User Interface window**.
2. When **the Graphic Simulation window** is still open, the **Graphic User Interface window** should not be used.

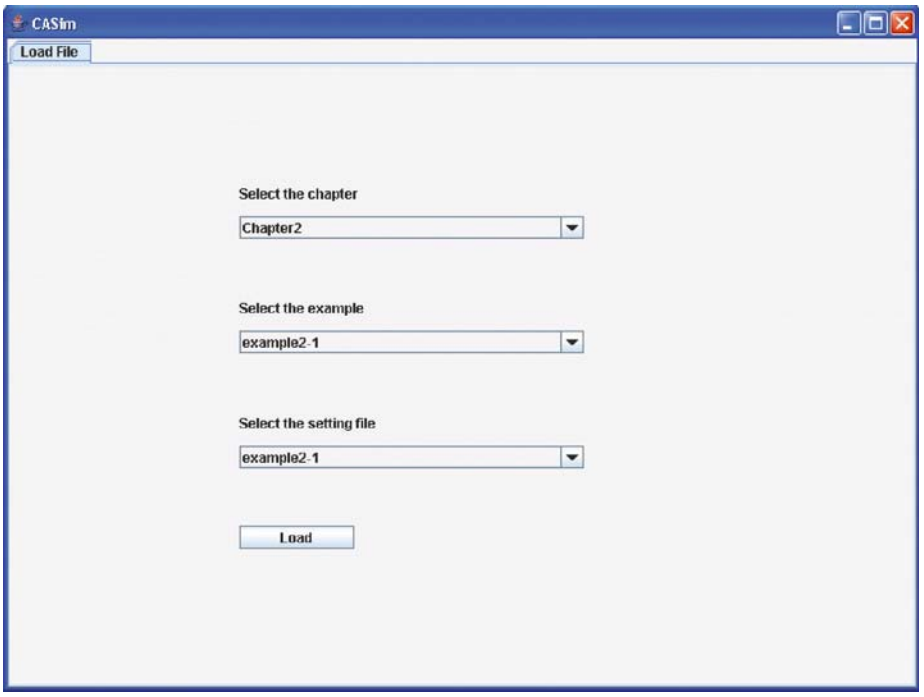


Figure 10.1.

10.6. Steps in setting up a simulation

The steps needed to set up a simulation, run it, and recover data are described here. More detailed instructions for each of the Steps 2–4 will be described later.

Step 1. Pick an Example in the text that represents the kind of simulation you wish to study.

Step 2. For MS Windows users, start the *CASim* software by double clicking ***CASim.BAT*** in the CAProgram directory. If you wish, you can bring ***CASim.BAT*** to the desktop so that you can start from there. For other platforms, first set CAProgram as the default directory, then you may start the software by entering "java CASim" at the command line.

Step 3. After Step 2, you will be greeted with a window with one tab (see **Figure 10.1).** Use the list box beneath "Select the chapter" to choose the desired chapter. The contents of the other two list boxes will change according to the selection of the chapter. Use the list box beneath "Select the example" to choose the desired Example. The content of the lowest list box will change to display the Setting files in the selected chapter's directory that are associated with the selected example. Use the list box beneath "Select the Setting file" to choose the desired Setting file. Click

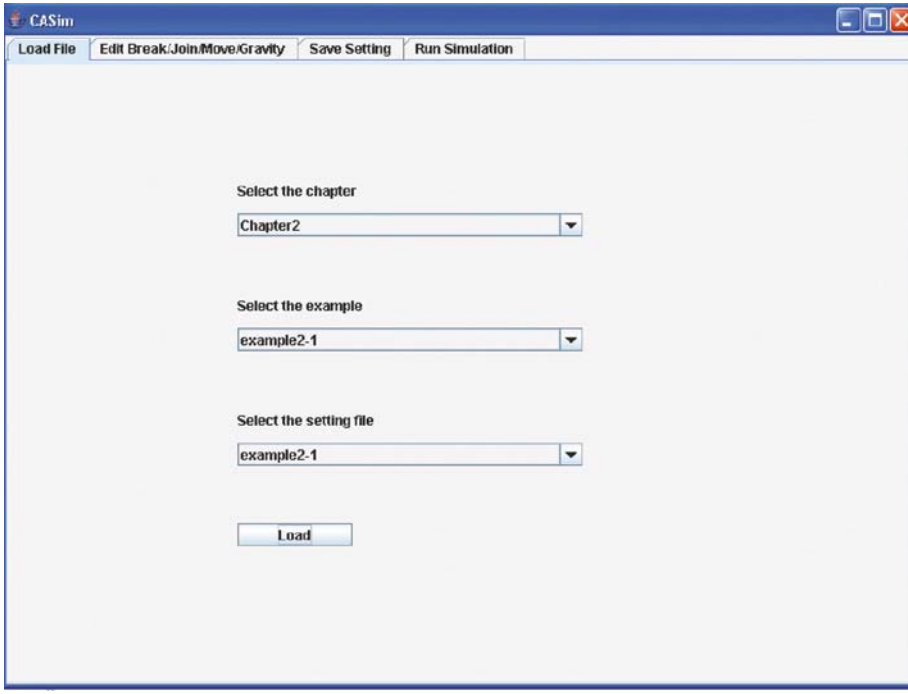


Figure 10.2.

the Load button; and you will see the same window but with more tabs (see **Figure 10.2**). Depending on which example is loaded, there will be different tabs. We will choose the tab "Run Simulation" in the next step, and postpone the discussion of other tabs until we reach the section on " Editing and Saving".

Step 4. Choose the tab "Run Simulation"; the window will show the corresponding page (see **Figure 10.3**). This page allows you to set the number of iterations required for this simulation, the iteration at which the program should start to gather data, the number of runs in this simulation, how often the data should be collected (e.g., once in 10 iterations), and a factor allowing you to slow down the simulation. There are five descriptors on the rightmost column on this page, corresponding to each of these parameters. Entering values into the edit boxes will set the corresponding parameters. Once all the parameter are set, you can click the Run button to start the simulation. The **Graphic Simulation window** will appear. The initial configuration of the grid is displayed, and at this moment, the simulation is in the suspend mode. A click on the ***Suspend/Resume*** button will start the dynamic process of the simulation. In fact, you can suspend or resume the activity by clicking the ***Suspend/Resume*** button. You may

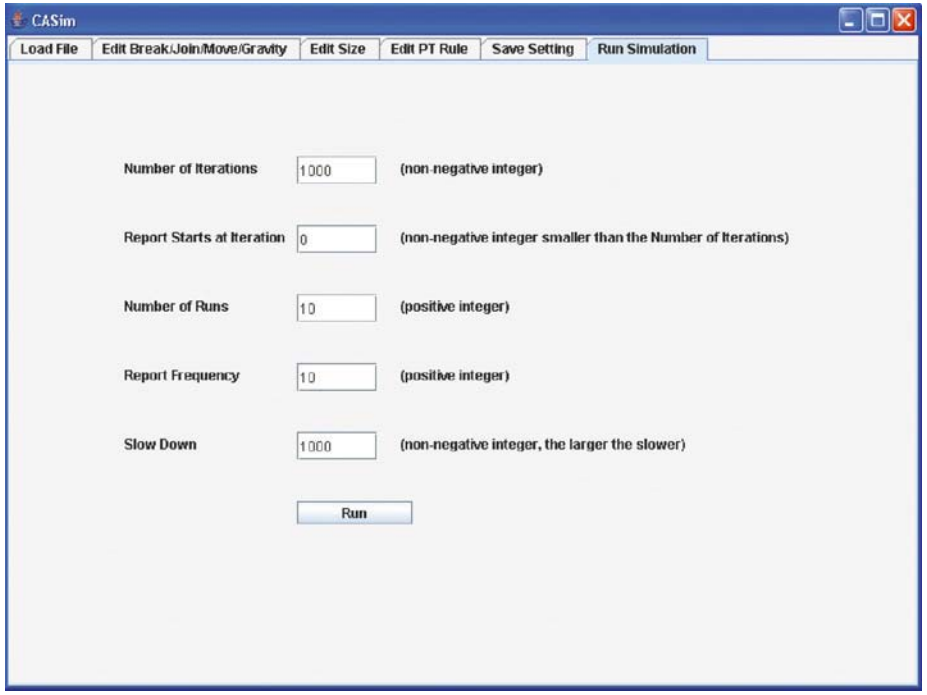


Figure 10.3.

close the **Graphic Simulation window** before or after the simulation is over. Once that is done, you can work with the **Graphic User Interface window** to start a new run, load a new example setting, or simply close the window.

Step 5. After the simulation, you can find the report file in the CAOutput directory. At this point you may wish to rename the output file so that you can relate it to the specific settings employed. You may also wish to remove this file from the CAOutput subdirectory so that this directory does not become too crowded. If you wish to perform a statistical analysis of the simulation data, you can import the data into an appropriate statistical analysis program, such as Microsoft's EXCEL®

10.7. Editing and Saving

Beside the "Load File" and "Run Simulation" tabs, there are five other tabs. These tabs are used to change specific settings and for saving the settings to a new or an existing Setting file. We will discuss the pages corresponding to each of these tabs below.

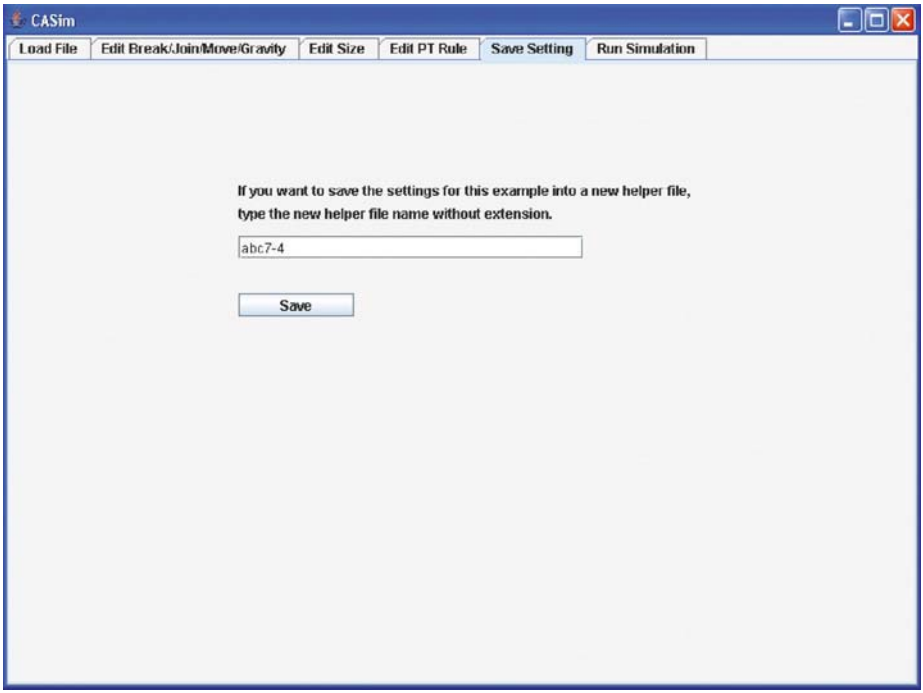


Figure 10.4.

10.8. Saving Setting tab

This page will allow you save the updated setting for future use (see **Figure 10.4**). If you do not want to save the settings to the file indicated in the edit box, key in the file name into which you want to save these settings. Once, you click the "Save" button, the present settings will be saved to the Setting file identified in the edit box. This Setting file will be associated with the most recently loaded Example.

10.9. Edit Size tab

This page will allow you to view and edit the number of ingredients and size of the grid (see **Figure 10.5**). The list box beneath "Ingredient Type" will display all the ingredient types involved in the selected example. The edit box to the right (beneath "ingredient number") will show the numbers of the ingredients of each type. Changing the number in the edit box will change the number of ingredients of the selected type. Changing the number in the edit boxes beneath "Number of Rows" and Number of Columns" will change the numbers of rows and columns of the grid . Note that the sum of the ingredients

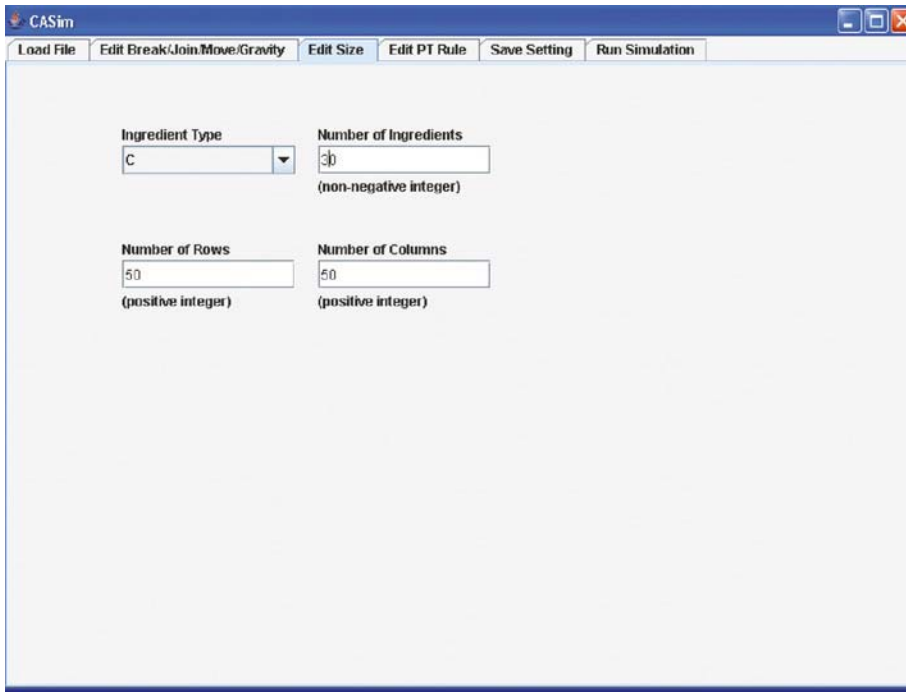


Figure 10.5.

must not be greater than the product of the number of rows and the number of columns (the grid size). An error message will appear later in the sequence if this rule is not observed.

10.10. Edit P_T Rule tab

This page will allow you to view and change the transition probabilities (see **Figure 10.6**). The present version of *CASim* supports the use of up to five ingredient types, namely A, B, C, D, and E. For each type of ingredient you can enter the desired P_T values ($0 \geq P_T \leq 1$) in the corresponding edit boxes to change their transition probabilities. For instance, if $P_T(A \rightarrow C)$ is 0.9, then enter 0.9 in the edit box at the intersection of the "From A" row and the "To C" column. *CASim* hides any edit boxes that should not be edited. For instance, the diagonal terms of the matrix should always be zero, so those edit boxes are hidden, also when the chosen example only uses A,B and C types, the edit boxes for the D and E rows and columns are hidden. **Note that the sum of entries in each row must not exceed 1**.

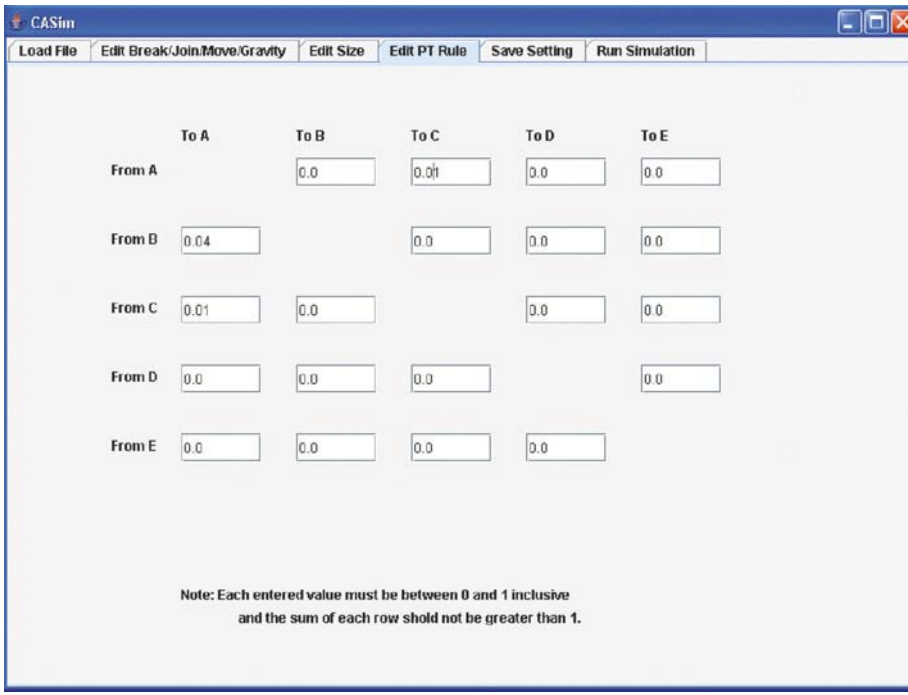


Figure 10.6.

10.11. Edit P_R Rule tab

This page will allow you to view and change the reaction probabilities (see Figure 10.7). The present version of *CASim* supports five reaction probability (P_R) possibilities, namely $A + A \rightarrow C + D$, $A + B \rightarrow C + D$, $C + D \rightarrow A + B$, $A + B \rightarrow A + C$, and $A + C \rightarrow A + B$. You can enter the desired P_R values ($0 \geq P_R \leq 1$) in the corresponding edit boxes. For instance, if you would like to set P_R for $A + B \rightarrow C + D$ to 0.5, enter 0.5 in the edit box to the right of "Probability for $A + B \rightarrow C + D$".

10.12. Edit Break/Join/Move/Gravity tab

This page will allow you to view and change the *breaking* and *free movement* probabilities and the *joining*, *absolute gravity*, and *relative gravity* factors (see **Figure 10.8**). In the leftmost column of this page, there are four list boxes with descriptive texts above each of them. To the right of those list boxes, there are edit boxes for the corresponding parameters . When you select an item in a list box, the edit box (or boxes) to the right will display the present value(s) of the parameter(s) related to the selected item. When you enter any new value into

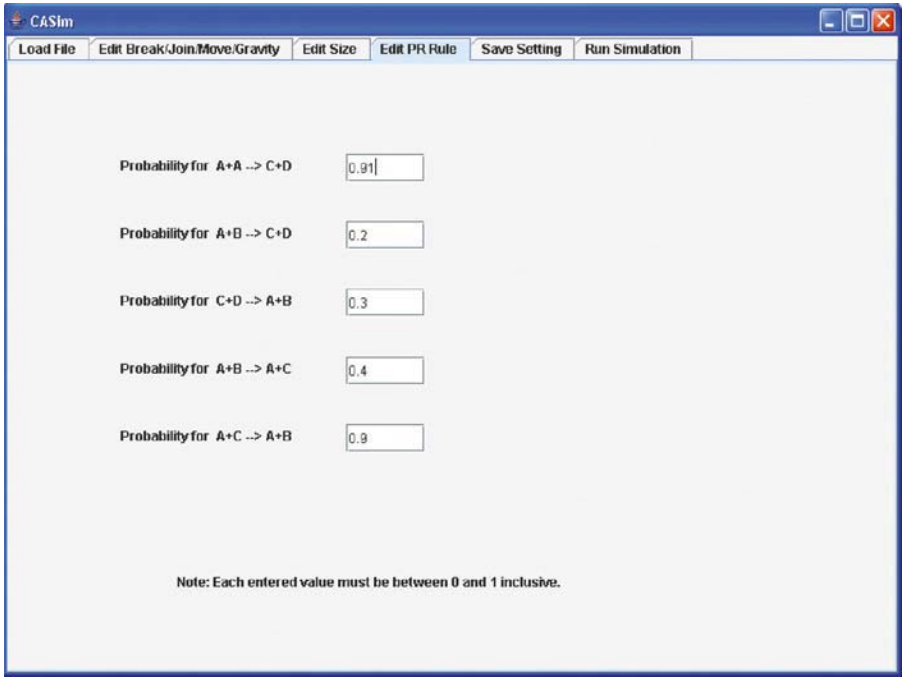


Figure 10.7.

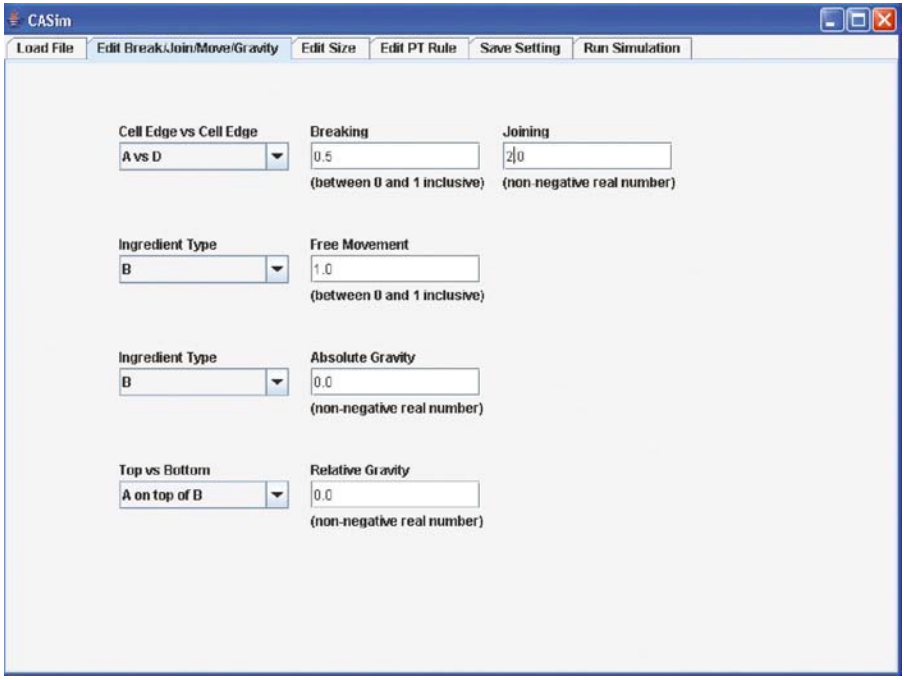


Figure 10.8.

an edit box, the value of the corresponding parameter will be changed to the entered value. For example, if Example4-5 is the chosen example and "W vs W" is selected in the topmost list box, the top edit box beneath "Breaking" might display 0.3. This indicates that the breaking probability P_B for an ingredient W breaking away from an adjacent ingredient W is 0.3. This can be changed as desired. Likewise, if the selected item in the second list box from the top is "W", then the free movement probability for all the W-type ingredients is set to 0.8. This same general procedure holds for the other parameters.

10.13. Error Messages

When the user clicks the "Save" or "Run" buttons, the system will invoke its consistency check routine, so that the system will always create Setting files with "admissible" values, and run a simulation with "admissible" settings. It will catch those inputted values which are an incorrect type, too big, etc. Whenever it finds a violation, an error message box will appear to inform the user where the problem is. For instance a message like "Error 06_301:NUMBER_OF_INGREDIENT value must be a non-negative integer" indicates that a negative value is entered into the NUMBER_OF_INGREDIENT edit box, which is on the page controlled by the "Edit Size" tab.

10.14. Identifying the *CASim* program

You may have occasion in a report or publication, to refer to the ***CASim*** program. The program should be identified as follows:

C.-K. Cheng, L. B. Kier, and P. G. Seybold, *CASim* Version 1.0, Center for the Study of Biological Complexity, Virginia Commonwealth University, Richmond, VA 23284, USA, 2005.

Reference

1. J. Gleick, *Chaos: Making a New Science*, Penguin: New York, 1988.

Index